

Web Console for Managing WAY4 Applications

Table of Contents

INTRODUCTION	3
CHAPTER 1. WEB CONSOLE DESCRIPTION	4
CHAPTER 2. STRUCTURE OF THE WEB CONSOLE'S MAIN WINDOW	5
Applications	6
Services	7
Servers	8
Actions	9
Information about a WAY4 application and service	11
Managing WAY4 applications and services	12
CHAPTER 3. WEB CONSOLE MAIN CONFIGURATION PARAMETERS	14
CHAPTER 4. LOGIN AND PASSWORD AUTHENTICATION OF CLIENTS IN THE WEB CONSOLE	17
CHAPTER 5. AUDIT LOG	19
CHAPTER 6. WEB CONSOLE IN CLUSTER MODE	20
Configuring web console to run in cluster mode	20
Starting a cluster	21
Managing cluster nodes	21
Actions on nodes	22
CHAPTER 7. MONITORING AND MANAGING NETSERVER	23
NetServer settings	23
Web console settings for interaction with NetSever	24

Introduction

This document contains information that is required to install and configure the web console ("console" system application supplied with the WAY4 Application Server distribution) and an introduction to user interface features.


When working with this document, it is recommended to use the following resources from the OpenWay documentation series:


- "WAY4™ Application Server Requirements".
- "Administering WAY4™ Application Server".

The following notation is used in this document:

- Screen form field labels are shown in *italics*.
- Key combinations are shown in angular brackets, for example, <Ctrl>+<F3>.
- Screen form button and tab labels are shown in square brackets, as in [OK].
- The names of commands and links on web console tabs are shown in curly brackets, as in {Create}.
- Variables that differ for each local instance, for example, directory and file names, as well as file paths, are shown in angular brackets; for example, <AppServer_HOME>.
- Mandatory fields in web client forms are highlighted in yellow.

Warnings and information messages are marked as follows:

 Warnings about potentially hazardous situations or actions.

 Messages with information about important features, additional options, or the best use of certain system functions.


Chapter 1. Web console description

The web console for managing WAY4 Applications is a WAY4 Application Server component. This component is used for the following:


- Monitoring the state of WAY4 applications and WAY4 Application Server components.
- Managing WAY4 applications and Transaction Switch services (starting, restarting, stopping).
- Viewing and editing configuration files for WAY4 applications, WAY4 Application Server and the web server component.

The web console is a system web application named "console".

The web console is automatically installed in the "<AppServer_HOME>/applications" directory during installation of WAY4 Application Server (see "Installing WAY4 Application Server" of the document "Administering WAY4 Application Server"). The web console is upgraded during upgrade of WAY4 Application Server.

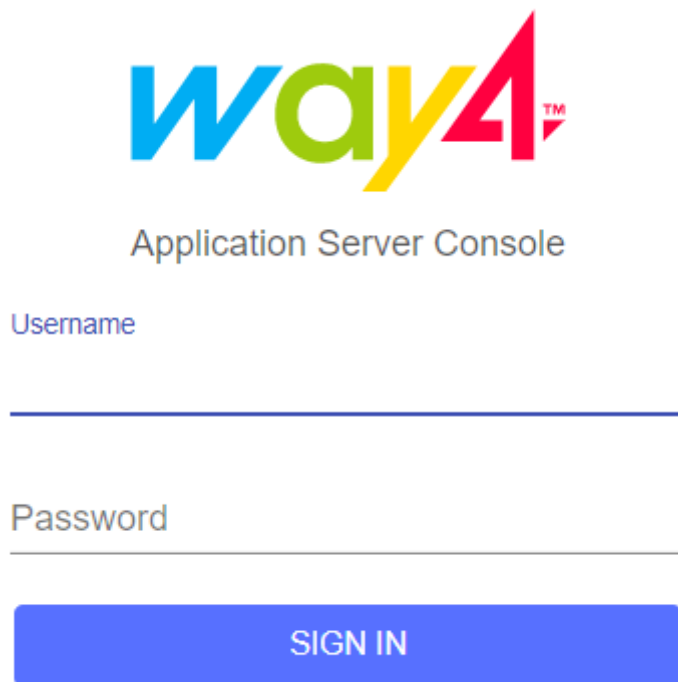
 If the console's web page was open in a browser during WAY4 Application Server upgrade, the web page must be refreshed (Ctrl+F5).

Management web console parameters can be changed in the appropriate configuration files (see "Configuring WAY4 Applications for Execution on WAY4 Application Server" of the document "Administering WAY4 Application Server") and in the file "<AppServer_HOME>/applications/console/conf/application.conf" (see "Web console main configuration parameters").

 After the web console has been installed, set up secure access via SSL/HTTPS for it. Setup can be performed during installation of WAY4 Application Server (see "Installing WAY4 Application Server under Unix" of the document "Administering WAY4 Application Server") or after installation using the "importcert" and "usrmgmt" utilities (adding users) (see "Utilities for Secure Access Setup" of the document "Administering WAY4 Application Server"). Password requirements pursuant to PCI DSS are described in the section "Registering Users and Setting Passwords" of the document "WAY4™ PCI DSS Implementation Guide".

Chapter 2. Structure of the web console's main window

After starting the console, enter the resource's URL "https://appserver_host_name:console_https_port" in the browser; where "appserver_host_name", is the domain name according to which WAY4 Application Server is accessed and "console_https_port" is the HTTPS port of the console specified in the file "<AppServer_HOME>/applications/console/conf/application.conf" (see "Web console main configuration parameters"). Next, the user is redirected to the login page:



way4™

Application Server Console

Username

Password

SIGN IN

Logging in to the console

If authentication is successful, the "Application Server Console" page will be displayed (see the figure).

1. Header – area at the top of the screen that includes the console's name and an open tab, an icon for refreshing data, user name and other information or warning signs that appear when an event occurs.
2. Tabs for access to different WAY4 Application Server components:
 - "Applications" – list of WAY4 applications that are installed, statistics about their operation and buttons for managing applications.
 - "Services" – list of available services for installed applications, filter for the list of services, buttons for managing services.
 - "Servers" – server or servers on which WAY4 Application Server instances are installed.

By default, the console is run in standalone mode; that is, on one server. This means the console makes it possible to manage WAY4 applications and services that are installed on the same WAY4 Application Server instance as the console application.

To allow management of resources on a local and remote WAY4 Application Server instances, the console must be started in cluster mode.

❗ Cluster mode is a licensed option. To obtain access to the feature, contact OpenWay.

- "Actions" – log of actions performed in the console.

3. Area with data and control elements for the selected tab.

<input type="checkbox"/>	Name	Server	Status	Actions	Version	Restarts	HTTP Port	HTTPS Port	Start Time
<input type="checkbox"/>	bgw	beta-lin1	STOPPED		2.9.13	0			
<input type="checkbox"/>	console	beta-lin1	STARTED		2.0.857	0			19-09-19 13:09
<input type="checkbox"/>	console_beta	beta-lin1	STARTED		2.0-SNAPSHOT	0			19-10-09 11:30
<input type="checkbox"/>	logagent	beta-lin1	STARTED		2.0.1045	0			19-10-04 10:40
<input type="checkbox"/>	logstash	beta-lin1	STARTED		7.3.1-674	0			19-09-20 13:43
<input type="checkbox"/>	m2_web_console	beta-lin1	STOPPED		1.0.2100	0	9094	9095	
<input type="checkbox"/>	monitoring	beta-lin1	STARTED		1.0.1536	0			19-09-20 12:32
<input type="checkbox"/>	monitoring_ui	beta-lin1	STARTED		1.0.1536	1	9090		19-09-29 23:17
<input type="checkbox"/>	sm-test-1	beta-lin1	STOPPED		0.1	0	20100		
<input type="checkbox"/>	test	beta-lin1	STOPPED			0	20000		
<input type="checkbox"/>	trsw2	beta-lin1	STARTED		1.0.139	0			19-10-03 15:04

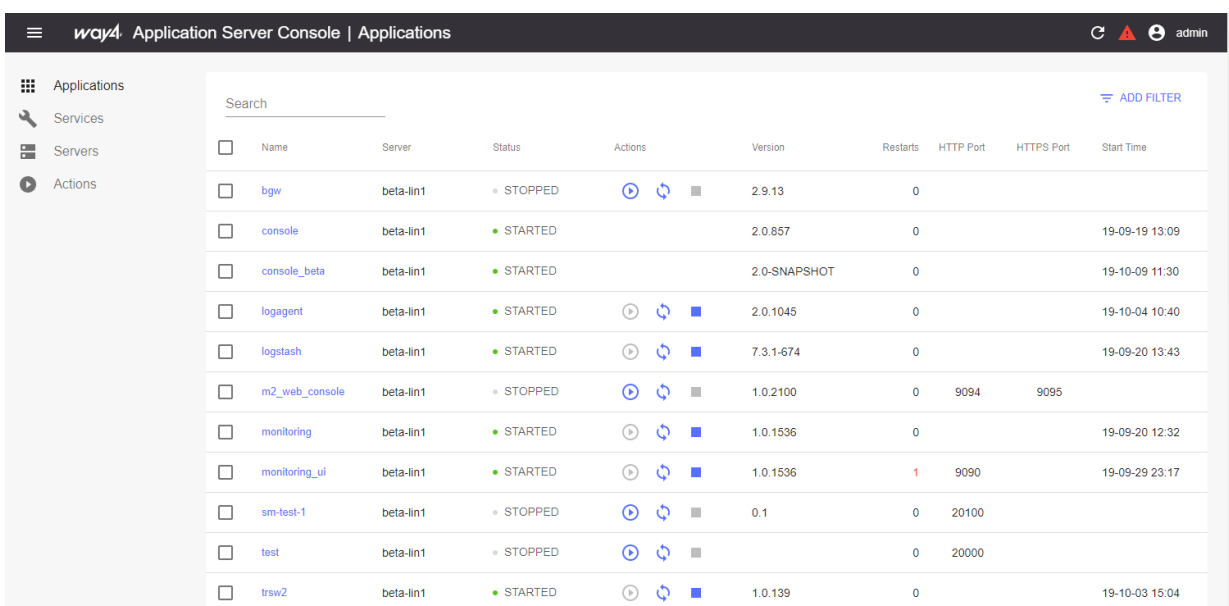
Main page of the web console for managing WAY4 applications

Applications

The "Applications" tab shows a list of WAY4 applications that are installed, statistics about their operation and buttons for managing applications (see the figure):

- *Name* – application name
- *Server* – name of the server on which the application is installed
- *Status* – application status
 - "STARTED" – the application is running
 - "STOPPED" – the application has been stopped
 - "STARTING" – the application is being started
 - "STOPPING" – the application is being stopped
 - "RESTARTING" – the application is being restarted
 - "CRASHED" – the application is not running due to an error in its behaviour (for example, the application needs a connection with the database, but the database is unavailable)

- "CORRUPTED" – a directory with the name of the application has been created, but files required to run the application are missing.
- "MALFUNCTION" – one or several of the application's services has the "CRASHED" or "MALFUNCTION" status.
- *Actions* – buttons for managing applications (see "Managing WAY4 Applications").
- *Version* – application version
- *Restarts* – number of application restarts
- *HTTP Port* – port used by the application
- *HTTPS Port* – HTTPS port of the server
- *Start Time* – for an application that is running ("STARTED" is specified in the STATUS field), the date and time it was started is specified



Name	Server	Status	Actions	Version	Restarts	HTTP Port	HTTPS Port	Start Time
bgw	beta-lin1	STOPPED	[Start] [Stop] [Restart]	2.9.13	0			
console	beta-lin1	STARTED	[Start] [Stop] [Restart]	2.0.857	0			19-09-19 13:09
console_beta	beta-lin1	STARTED	[Start] [Stop] [Restart]	2.0-SNAPSHOT	0			19-10-09 11:30
logagent	beta-lin1	STARTED	[Start] [Stop] [Restart]	2.0.1045	0			19-10-04 10:40
logstash	beta-lin1	STARTED	[Start] [Stop] [Restart]	7.3.1-674	0			19-09-20 13:43
m2_web_console	beta-lin1	STOPPED	[Start] [Stop] [Restart]	1.0.2100	0	9094	9095	
monitoring	beta-lin1	STARTED	[Start] [Stop] [Restart]	1.0.1536	0			19-09-20 12:32
monitoring_ui	beta-lin1	STARTED	[Start] [Stop] [Restart]	1.0.1536	1	9090		19-09-29 23:17
sm-test-1	beta-lin1	STOPPED	[Start] [Stop] [Restart]	0.1	0	20100		
test	beta-lin1	STOPPED	[Start] [Stop] [Restart]		0	20000		
trsw2	beta-lin1	STARTED	[Start] [Stop] [Restart]	1.0.139	0			19-10-03 15:04

"Applications" tab

Services

The "Services" tab shows the list of available services for installed applications, filter for the list of services, buttons for managing services see the figure):

- *Service* – service name.
- *Application* – name of the application providing the service
- *Server* – name of the server on which the application that provides the service is running
- *Status* – service status
 - "STARTED" – the service is running
 - "STOPPED" – the service has been stopped
 - "STARTING" – the service is being started
 - "STOPPING" – the service is being stopped
 - "RESTARTING" – the service is being restarted

- "CRASHED" – the service is not running due to an error
- "MALFUNCTION" – the service is running with errors
- *Status Codes* – service status code:
 - "DISCONN01" – the service is not connected with an external system
 - "UNAVAIL01" – the service is not available for internal calls
 - "CONF01" – the service cannot be started due to syntax errors in its configuration
- *Actions* – buttons for managing services (see "Actions").
- *Srv* – number of successful messages/transactions.
- *Srv Error* – number of unsuccessful messages/transactions.

	Service	Application	Server	Status	Status Codes	Actions	Srv	Srv Error
<input type="checkbox"/>	service01	trsw2	beta-lin1	STOPPED		[play] [refresh] [stop] [menu]	-1 328 179 181	5
<input type="checkbox"/>	service02	trsw2	beta-lin1	STOPPED		[play] [refresh] [stop] [menu]	-1 328 179 181	5
<input type="checkbox"/>	AcqDB	trsw1	beta-sol1	STARTED		[play] [refresh] [stop] [menu]	0	0
<input type="checkbox"/>	HSMAdapter	trsw1	beta-sol1	STOPPED		[play] [refresh] [stop] [menu]	0	0
<input type="checkbox"/>	IssDB	trsw1	beta-sol1	CRASHED		[play] [refresh] [stop] [menu]	0	0
<input type="checkbox"/>	MC_CREDIT	trsw1	beta-sol1	MALFUNCTION	DISCONN01 UNAVAIL01	[play] [refresh] [stop] [menu]	0	0
<input type="checkbox"/>	Platform	trsw1	beta-sol1	STARTED		[play] [refresh] [stop] [menu]	0	0
<input type="checkbox"/>	WAY4DB	trsw1	beta-sol1	MALFUNCTION	DISCONN01	[play] [refresh] [stop] [menu]	0	14 357

Rows per page: 25 1-8 of 8

Copyright © OpenWay, 2019. All rights reserved. Application Server Console version 2.0-SNAPSHOT.

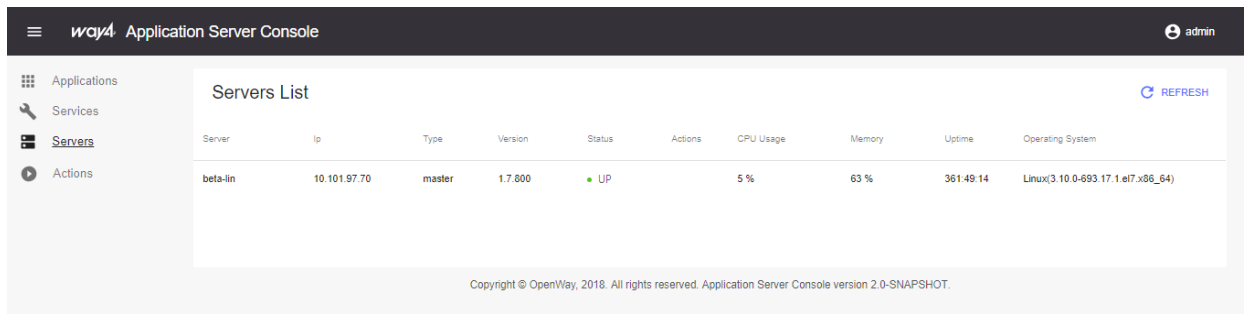
Services tab

Servers

The "Servers" tab shows information about the server or servers (for more information, see "Web console in cluster mode"), on which a WAY4 Application Server instance is installed (see the figure):

- *Server*– server name
- *Ip* – server's IP address
- *Type* – node type (master or slave)
- *Version* – WAY4 Application Server version
- *Status* – server status
- *Actions* – buttons for managing servers Only available in cluster mode (see "Web console in cluster mode");
- *CPU Usage* – processor load as a percentage
- *Memory* – percentage of physical memory used on the server

- *Uptime* – time for which WAY4 Application Server has been running without interruption
- *Operating System* – server's operating system



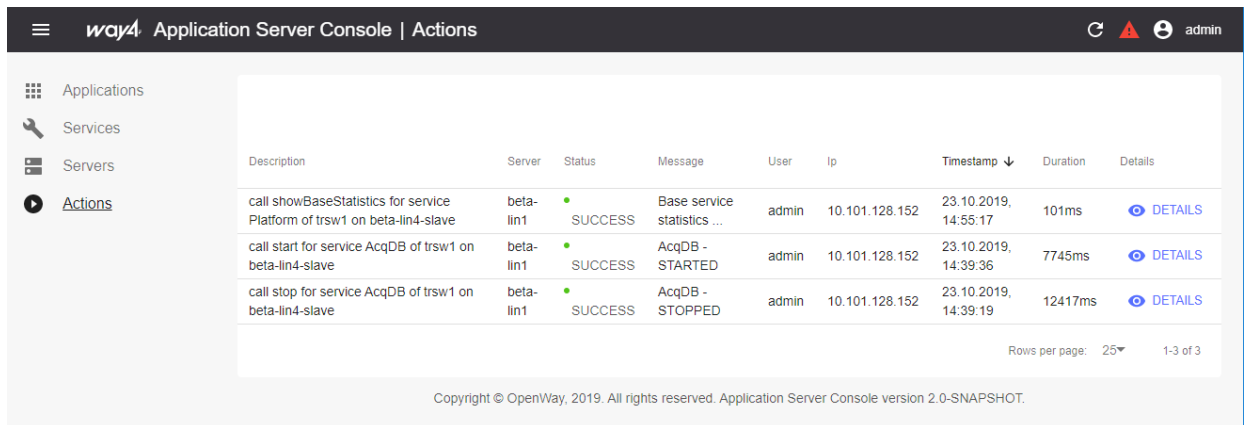
Server	Ip	Type	Version	Status	Actions	CPU Usage	Memory	Uptime	Operating System
beta-lin	10.101.97.70	master	1.7.800	UP		5 %	63 %	361:49:14	Linux(3.10.0-693.17.1.el7.x86_64)

Copyright © OpenWay, 2018. All rights reserved. Application Server Console version 2.0-SNAPSHOT.

Servers tab

Actions

The "Actions" tab shows the log of actions performed on WAY4 applications and services in the web console (see the figure).



Description	Server	Status	Message	User	Ip	Timestamp ↓	Duration	Details
call showBaseStatistics for service Platform of trsw1 on beta-lin4-slave	beta-lin1	SUCCESS	Base service statistics ...	admin	10.101.128.152	23.10.2019, 14:55:17	101ms	DETAILS
call start for service AcqDB of trsw1 on beta-lin4-slave	beta-lin1	SUCCESS	AcqDB - STARTED	admin	10.101.128.152	23.10.2019, 14:39:36	7745ms	DETAILS
call stop for service AcqDB of trsw1 on beta-lin4-slave	beta-lin1	SUCCESS	AcqDB - STOPPED	admin	10.101.128.152	23.10.2019, 14:39:19	12417ms	DETAILS

Rows per page: 25 1-3 of 3

Copyright © OpenWay, 2019. All rights reserved. Application Server Console version 2.0-SNAPSHOT.

"Actions" tab

Click the [DETAILS] button to display detailed information about an action. For example, when the AcqDB service is started, information shown in the figure will be written to the log.

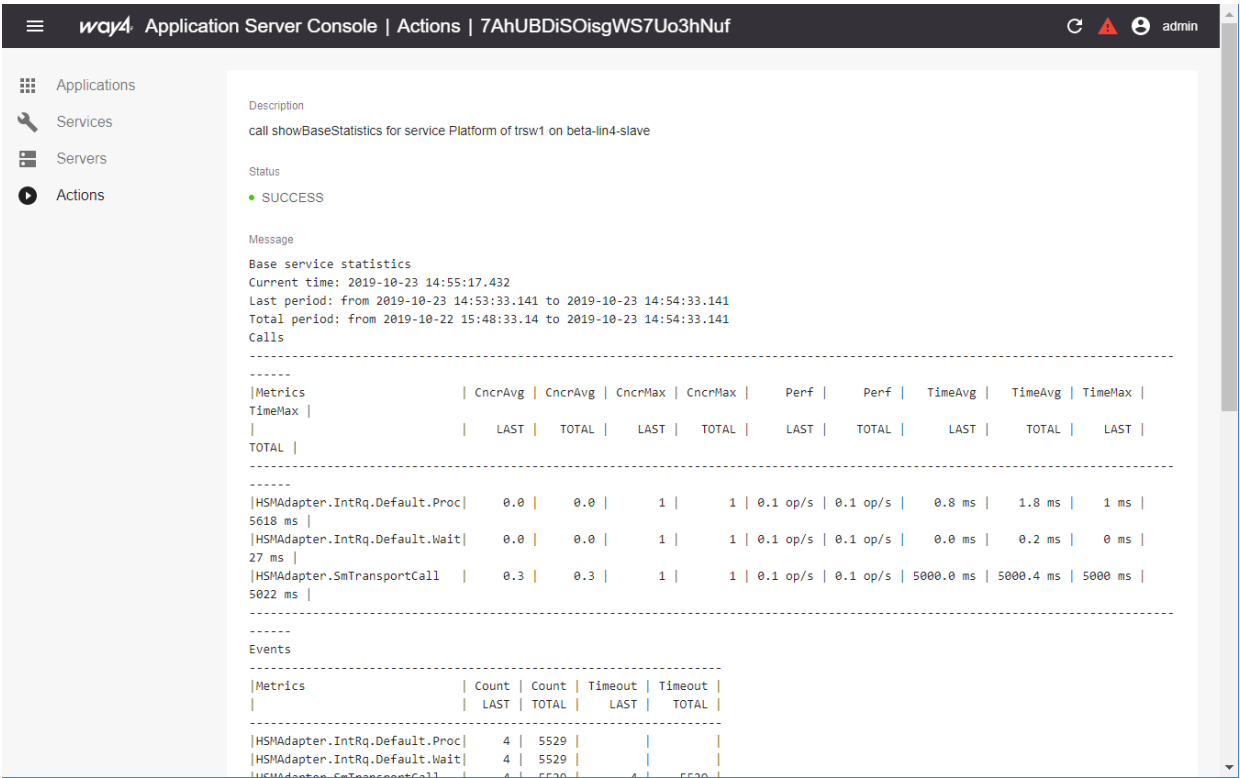
The screenshot displays the OpenWay Application Server Console interface. The top navigation bar includes a hamburger menu, the 'way4' logo, the title 'Application Server Console | Actions | USshkGu8WS9I', a refresh icon, a warning icon, and a user profile icon labeled 'admin'. The left sidebar contains a list of navigation items: 'Applications' (grid icon), 'Services' (wrench icon), 'Servers' (server rack icon), and 'Actions' (play button icon). The main content area shows the details of a specific action:

- Description:** call start for service AcqDB of trsw1 on beta-lin4-slave
- Status:** SUCCESS (indicated by a green dot)
- Message:** AcqDB - STARTED
- User:** admin
- Ip:** 10.101.128.152
- Timestamp:** 23.10.2019, 14:39:36
- Duration:** 7745ms
- Id:** req:USshkGu8WS9IeEa0UXAXi

At the bottom of the console, a copyright notice reads: 'Copyright © OpenWay, 2019. All rights reserved. Application Server Console version 2.0-SNAPSHOT.'

Detailed information about an action

When the additional command "showBaseStatistics" is executed, when the [DETAILS] button is clicked, basic statistics for the service will be shown.

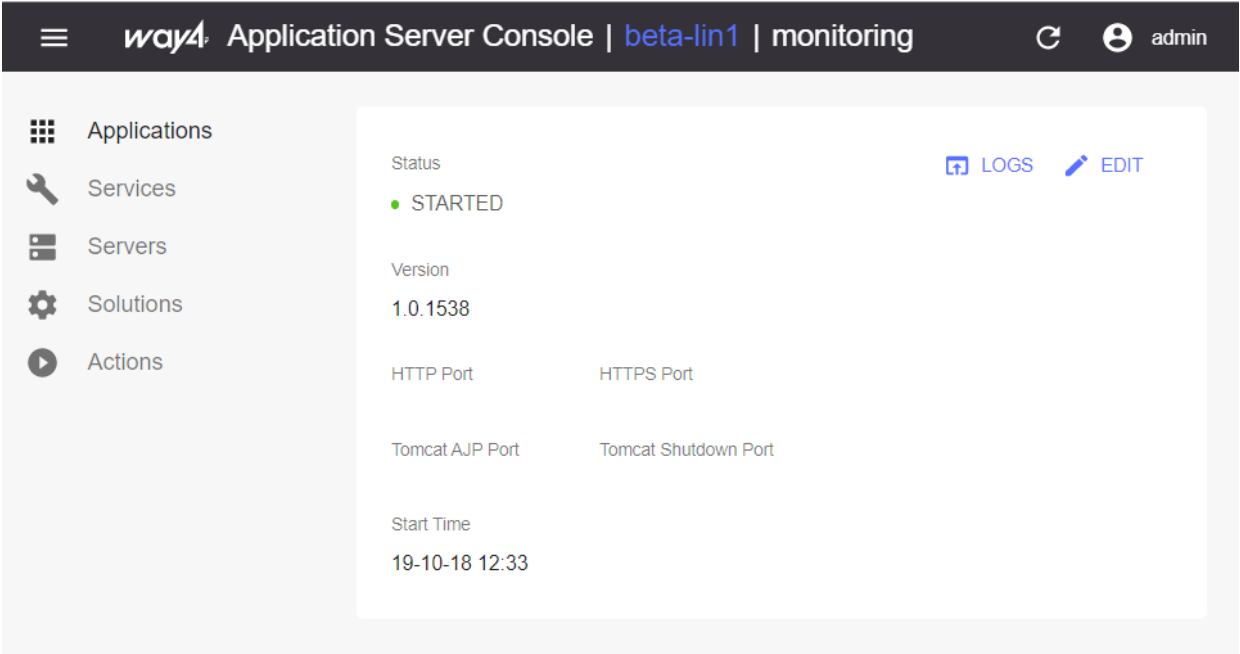


Detailed information when the additional command is executed

Information about a WAY4 application and service


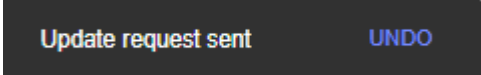
To view and edit configuration files for a WAY4 application or service, do as follows:

- On the web console's main page, click on the name of the WAY3 application or service.
- On the page that opens, click the [EDIT] button (see the figure);



Information about an application's operation





- On the page that opens, select the necessary configuration file and edit it.

To save changes to the configuration file, click . To cancel changes that were made to the configuration, click .


To view log files, click the [LOGS] button on the page with information about operation of the WAY4 application or service.

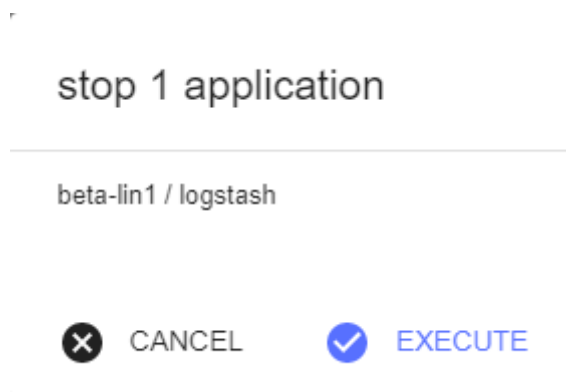
Managing WAY4 applications and services

To manage WAY4 applications and services, click the appropriate button to determine the action that must be performed on the selected WAY4 application. The following buttons can be used:

-  – start a WAY4 application or service
-  – stop a WAY4 application or service
-  – restart a WAY4 application or service
-  – open a context menu with a list of additional commands that are available for the service. This button will only be available for services that have additional commands.

 Note that the list of commands depends on the service.

Clicking a button for managing WAY4 applications opens a window in which execution of the command must be confirmed by clicking the "EXECUTE" button. For example, when the button  is clicked for the "logstash" application, a window like the one shown in the figure opens.



"stop 1 application" window

Group commands can also be used to manage WAY4 applications and services:

- START – start a group of selected WAY4 applications or services
- RESTART – restart a group of selected WAY4 applications or services
- STOP – stop a group of selected WAY4 applications or services

To start, restart, or stop a group of WAY4 applications or services, select the check box to the left the name of the WAY4 application or service and select the required command in the upper right corner of the window (see the figure).

☰

way4 Application Server Console | Applications

🔄 ⚠️ 👤 admin

☰ Applications

🔧 Services

🖨 Servers

▶ Actions

2 items selected

▶ START

🔄 RESTART

■ STOP

<input type="checkbox"/>	Name	Server	Status	Actions	Version	Restarts	HTTP Port	HTTPS Port	Start Time
<input type="checkbox"/>	bgw	beta-lin1	⚪ STOPPED	▶ 🔄 ■	2.9.13	0			
<input type="checkbox"/>	console	beta-lin1	🟢 STARTED		2.0.657	0			19-09-19 13:09
<input type="checkbox"/>	console_beta	beta-lin1	🟢 STARTED		2.0-SNAPSHOT	0			19-10-09 11:30
<input type="checkbox"/>	logagent	beta-lin1	🟢 STARTED	▶ 🔄 ■	2.0.1045	0			19-10-04 10:40
<input type="checkbox"/>	logstash	beta-lin1	🟢 STARTED	▶ 🔄 ■	7.3.1-674	0			19-09-20 13:43
<input type="checkbox"/>	m2_web_console	beta-lin1	⚪ STOPPED	▶ 🔄 ■	1.0.2100	0	9094	9095	
<input checked="" type="checkbox"/>	monitoring	beta-lin1	🟢 STARTED	▶ 🔄 ■	1.0.1536	0			19-09-20 12:32
<input checked="" type="checkbox"/>	monitoring_ui	beta-lin1	🟢 STARTED	▶ 🔄 ■	1.0.1536	1	9090		19-09-29 23:17
<input type="checkbox"/>	sm-test-1	beta-lin1	⚪ STOPPED	▶ 🔄 ■	0.1	0	20100		
<input type="checkbox"/>	test	beta-lin1	⚪ STOPPED	▶ 🔄 ■		0	20000		
<input type="checkbox"/>	trsw2	beta-lin1	🟢 STARTED	▶ 🔄 ■	1.0.139	0			19-10-03 15:04

Group execution of commands

When a group control command is selected, a confirmation window will open. Click "EXECUTE" in this window. For example, when the "STOP" command is selected for the "monitoring_ui" and "monitoring" applications, a window opens like the one shown in the figure.

stop 2 applications

beta-lin1 / monitoring_ui

beta-lin1 / monitoring

✕ CANCEL

✓ EXECUTE

"stop 2 applications" window


Chapter 3. Web console main configuration parameters

The web console's main configuration file is "<AppServer_HOME>/applications/console/conf/application.conf". Basic parameters required for the web console to operate are shown in the table.

To set up operation of the web console in cluster mode and to set up monitoring and management of NetServer, values must be specified for additional parameters (for more information, see "Web console in cluster mode" and "Monitoring and managing NetServer").

i After changes have been made to the web console's configuration file parameters, restart the web console. If the console's web page was open in the browser when changes were made to the configuration, the web page must be refreshed (Ctrl+F5).

Parameter group	Parameter name	Default value	Description
HTTPS	app.http.port	9085	HTTPS server port. If the port is already used, change the default value.
Node	app.node.ip	Automatically determined IP address <getHostAddress>	Node's IP address available from the network. The address must correspond to the IPv4 protocol. The default value "<getHostAddress>" means that the address is determined automatically when an application is started. If this is a feedback address (for example, "127.0.0.1"), the application is not started. In this case, the parameter's value must be set explicitly. The value "127.0.0.1" is available if the application will only be used from a local host.
			The node's allowed DNS name that is available on the network or the IP address, if the DNS name is unavailable. The default value "<getHostName>" means that the name is determined automatically when the application is started.
	app.node.dns-name	Automatically determined DNS name <getHostName>	If the automatically determined host name is converted to a feedback address (for example, "localhost"), the application is not started. In this case, the parameter's value

Parameter group	Parameter name	Default value	Description
			must be set explicitly. The value "localhost" is available if the application will only be used from a local host.
	app.node.name	\${app.node.dns-name}	Node's name that is shown in the web console interface.
Logger	app.logger.log-level	INFO	Level of detail for data output to the message log. The following logging levels are supported (in ascending order): TRACE, DEBUG, INFO, WARN, ERROR.
	app.logger.max-file-size	10MB	Maximum log file size. The following formats are supported for setting the parameter's value – 5000000, 5000KB, 5MB and 5GB.
	app.logger.total-size-cap	1GB	Maximum size of all log files. The following formats are supported for setting the parameter's value – 5000000, 5000KB, 5MB and 5GB (for one set of log files with the same prefix).
	app.logger.max-file-history	60	Maximum number of log files (for one set of log files with the same prefix).
Authentication	app.http.authentication.jwt.expiration-time	4 hours	JWT token duration (used for access to the REST API). The user interface updates the JWT token automatically in background mode. Note that this parameter's value must be at least 7 minutes.
	app.http.authentication.embedded.user-database.expiration-time	90 days	User password duration time in days. The following formats are supported for setting the parameter's value – 90d, 90 days, 90 day.
	app.http.authentication.password-expiration-warning	15 days	Number of days until the user password expires, during which the upper part of the web console window will contain a warning sign  with the message "Password expires in <N> day(s)". The following formats are supported for setting the parameter's value – 15d, 15 days, 15 day.

Parameter group	Parameter name	Default value	Description
	app.http.authentication.auto-logout-timeout	15 minutes	Timeout for automatic log out from the web console interface if no user actions were performed during this time.

Chapter 4. Login and password authentication of clients in the web console

To restrict access privileges, various types of users can be created for the web console:

- **A user** can monitor the states of WAY4 applications and services, manage WAY4 applications and services, and can edit configurations. No restrictions.
- **An operator** can monitor the states of WAY4 applications and services, manage WAY4 applications and services, but cannot edit configurations (a configuration is shown as read-only and instead of the [Edit] button, there will be a [View] button for viewing the configuration).
- **A watcher** can monitor the states of WAY4 applications and services, but cannot edit configurations (a configuration is shown as read-only and instead of the [Edit] button, there will be a [View] button for viewing the configuration). Watchers cannot manage WAY4 applications and services.

To set up login and password authentication of clients in the web console, do as follows:

- Use the "usrmgmt" utility to add a new user (see "usrmgmt" Utility in the document "Administering WAY4™ Application Server"), where <role name> is the user role:
 - "user"
 - "watcher"
 - "operator"

Note that before executing the "usrmgmt" command, the certificate for the "console" application must have been imported using the "importcert" command (see "importcert" Utility in the document "Administering WAY4™ Application Server").

```
$AS_HOME/appserver/bin/usrmgmt app_name=console auth_method=FORM oper=add  
user_name=<user> password=<password> role_name=<role name>
```

Example of a correct "tomcat-users.xml" file:


```
<tomcat-users>  
<role rolename="user"/>  
<user lastUpdate="1956806027707" password="7fcf4ba391c48784edde599889d6e3f1e47a27  
db36ecc050cc92f259bfac38afad2c68a1ae804d77075e8fb722503f3eca2b2c1006ee6f6c7b7628c  
b45fffd1d" roles="user" username="owsuser"/>  
<role rolename="operator"/>  
<user lastUpdate="1571130843126" password="7fcf4ba391c48784edde599889d6e3f1e47a27  
db36ecc050cc92f259bfac38afad2c68a1ae804d77075e8fb722503f3eca2b2c1006ee6f6c7b7628c  
b45fffd1d" roles="operator" username="operator"/>  
<role rolename="watcher"/>  
<user lastUpdate="1571130858072" password="7fcf4ba391c48784edde599889d6e3f1e47a27  
db36ecc050cc92f259bfac38afad2c68a1ae804d77075e8fb722503f3eca2b2c1006ee6f6c7b7628c  
b45fffd1d" roles="watcher" username="watcher"/>  
</tomcat-users>
```


Chapter 5. Audit log


Web console audit log files (audit*.log) are written to the directory "<AS_HOME>/appserver/applications/console/logs/".

The following events are logged:

- User actions that change the system's state (for example, starting, stopping, restarting a WAY4 application or service)
- log into and log out of the system
- reading and importing log files

 If a log file remains open in the user interface for a long time, this may cause additional audit log files to be written. This happens because the user interface is periodically refreshed by background tasks.

Chapter 6. Web console in cluster mode

 Cluster mode is a licensed option. To obtain access to the feature, contact OpenWay.

A cluster consists of several instances of the web console application that are running on different servers (the application runs on WAY4 Application Server). Each instance of the web console is a cluster node.

Starting a node means starting a web console that is set up to use cluster mode (see the section "Configuring web console to run in cluster mode"). Stopping the web console application on a server means stopping the node. If the node is stopped, this node's applications will be unavailable in the web console.

To use cluster mode, set up and start the web console on each server in the cluster.

Configuring web console to run in cluster mode

By default, the web console is started in standalone mode, meaning the application makes it possible to manage resources (applications, services), that belong to the local WAY4 Application Server instance only.

To allow management of resources on local and remote application server instances, web console must be started in cluster mode.

To set up web console to run in cluster mode, set the following parameters:

Group	Name	Default value	Description
Cluster	app.cluster.port	2552	TCP port used for internetwork data exchange. If the default port is already used, change the parameter's value.
	app.cluster.host	\${app.node.ip}	IP address used for internetwork data exchange on the IPv4 protocol. For example: app.cluster.host=10.101.100.100
	app.cluster.node-type	standalone	The parameter is used to distinguish nodes as main and subordinate. This is necessary to reduce the number of links between nodes. Node type: <ul style="list-style-type: none">standalone – standalone mode.master – cluster mode, main node. Resources on a master node can be managed from the local node or from any other master node.slave – cluster mode, subordinate node. Resources on a slave node can be managed from the local slave node or from any master node.

Group	Name	Default value	Description
			For example: app.cluster.node-type=master
	app.cluster.seed-nodes	[] (empty list)	<p>The list of a cluster's master nodes for connecting when the application is started. Each element must be a string in the format: "\${app.cluster.host}:\${app.cluster.port}".</p> <p>This parameter's value must be the same on each (both master and slave) node in the cluster. Otherwise, when the cluster is started, it may split into several independent clusters.</p> <p>It is highly recommended to have an uneven number of nodes in this list to minimize the probability of a split into several clusters (cluster split-brain).</p> <p>Example: app.cluster.seed-nodes=["10.101.100.100:2552","10.101.100.101:2552","10.101.100.102:2552"]</p>

Starting a cluster

A cluster's nodes must be started in the following order:

- The first master node is started from the list of nodes in the "app.cluster.seed-nodes" configuration parameter.
- The remaining master nodes are started from the list of nodes in the "app.cluster.seed-nodes" configuration parameter.
- The cluster's remaining slave nodes are started (can be started in any order).
- Log in to the console application on any master node and go to the "Servers" tab. Nodes that were started successfully have the "UP" status (see the figure).

Server	IP	Type	Version	Status	Actions	CPU Usage	Memory	Uptime	Operating System
beta-lin1	10.101.97.70	master	1.7.1362	UP		9 %	64 %	483:56:36	Linux(3.10.0-957.12.2.el7.x86_64)
beta-lin2	10.101.97.69	master	1.7.1362	UP		12 %	66 %	483:51:34	Linux(3.10.0-957.5.1.el7.x86_64)
beta-lin4-slave	10.101.97.67	master	1.7.1362	UP		10 %	84 %	483:38:33	Linux(3.10.0-693.17.1.el7.x86_64)
beta-sol1	10.101.83.245	master	1.7.1362	UP		6 %	72 %	20:12:41	SunOS(5.11)

Rows per page: 25 1-4 of 4

Copyright © OpenWay, 2019. All rights reserved. Application Server Console version 2.0-SNAPSHOT.

Managing cluster nodes

A cluster node can have the following statuses:

Status	Description
JOINING	Transition state when a node is joining a cluster.
UP	Normal state of operation.
LEAVING	Transition state when a node is correctly completing operation.
EXITING	Transition state when a node is correctly completing operation.
DOWN	The node was disconnected correctly or the connection with the node was lost and the node was disconnected manually (using the "DOWN" action). A node in this state cannot be reconnected to the cluster without restarting the node.
UNREACHABLE	Connection with the node was lost. The node automatically goes to the "UP" state after the connection has been restored.

When a cluster has nodes with the UNREACHABLE status, it goes into a "worsened" state. In this state, several actions on nodes are unavailable.

If a node becomes unavailable due to a temporary network failure, no user actions are required. The node will be restored after the connection has been restored.

If it is known that a network connection with an UNREACHABLE-node will not be restored in the near future, this node must be disconnected manually using the "DOWN" action.

Actions on nodes

Action	Status	Description
Down	UNREACHABLE	If a node's status has become UNREACHABLE, its status must be manually changed to DOWN. This is necessary when a node is unavailable and the network connection will not be restored in the near future. After the network connection has been restored, the node cannot automatically connect to the cluster. To attach this node to the cluster, the node must be restarted.
Forget	DOWN	Delete a node from the list of connected nodes.

Chapter 7. Monitoring and managing NetServer

❗ Monitoring and managing NetServer is a licensed option. To obtain access to the feature, contact OpenWay.

For monitoring and managing NetServer to be available in the web console, it is necessary to make settings in WAY4 Application Server and the NetServer application.

NetServer settings

In the script "<NS_HOME>/bin/way4d", replace the parameter:

```
EXEFILE="java -jar $NS_HOME/classes/Jdaemon2.jar"
```

with the following parameters:

```
AS_HOME=<path_to_appserver_installation>
CON=$AS_HOME/appcontainer/lib
NS_REST_PORT=16100
NS_REST_THREADS=6
NS_REST_APP_NAME=ns2
RMI="-javaagent:$CON/jolokia-jvm-agent.jar=port=$NS_REST_PORT,executor=fixed,
threadNr=$NS_REST_THREADS"
EXEFILE="java $RMI -Dapp.key=$NS_REST_APP_NAME -jar $NS_HOME/classes/Jdaemon2.
jar"
```

where:

- "AS_HOME" is the path to WAY4 Application Server. For example, "/home/way4/appserver".
- "\$NS_REST_PORT" is the REST port for managing NetServer. Any free port can be used.
- "\$NS_REST_THREADS" is the number of threads for managing NetServer channels. The number of threads must be equal to the maximum number of simultaneously reloaded channels plus one thread. An additional thread is required to refresh information in the web console about NetServer's status and its channels.
- "\$NS_REST_APP_NAME" is the NetServer application name that will be used in the web console.

Web console settings for interaction with NetSever

In the configuration file "<AppServer_HOME>/applications/console/conf/application.conf" add the following parameters and set their values:

Parameter group	Parameter name	Description
NetServer	app.applications.external.<app_name>.app-type	Type of external application – netserver.
	app.applications.external.<app_name>.rest-port	REST port for managing NetServer. The value of the "\$NS_REST_PORT" variable specified in the "way4d" script.
	app.applications.external.<app_name>.collection-user	NetServer user for collecting information about NetServer channels.

Where "<app_name>" is the value of the "\$NS_REST_APP_NAME" variable specified in the "way4d" script.

Information about users who can work with NetServer can be found in the file "<NS_HOME>/netserver/conf/daemon/userreg". To manage NetServer using the web console, it is necessary to generate user certificates (see the description of the command file "new_cert" in the section "Web Application Certificate Generation" of the document "Administering WAY4 Application Server"); the certificate name must match the NetServer user name from the "userreg" file. For example, if the user name "ns_user" is specified in the "userreg" file, to generate a user certificate, run the "new_cert" with the parameter "ns_user":

```
<AppServer_HOME>/certs/new_cert.sh ns_user
```

i Note that to manage NetServer, values must be set for all the parameters in the NetServer group. If it is not necessary to monitor and manage NetServer, delete (or comment out) all parameters from the NetServer group.

Sample web console configuration:

```
app.applications.external.ns2.app-type=netserver
app.applications.external.ns2.rest-port=16100
app.applications.external.ns2.collection-user=collection-user
```