orkhestra_admin: Orkhestra: Central Administration Version 1

CML00121-00

Code Magus Limited (England reg. no. 4024745)
Number 6, 69 Woodstock Road
Oxford, OX2 6EY, United Kingdom
www.codemagus.com

November 7, 2023

Contents

1 Introduction

The program *orkadmin* is a command line tool that interacts with the specified instance of *Orkhestra* Central in order to perform functions such as prime an *Orkhestra* Central, define an initial administrator, administer users and assign permissions, add/delete configurations, add/delete dashboards, launch details, log and view events.

For further information on *Orkhestra* and state machines please refer to orkhestra: Configuration and User Reference Version 1[1]

2 Synopsis

```
jan@bruinkop:~/software/orkcntrl$ orkadmin --help
Code Magus Limited Orkhestra Central V3: build 2023-10-20-13.39.54
[orkadmin] $Id: orkadmin_help.tex,v 1.7 2023/10/26 11:15:01 janvlok Exp $
Copyright (c) 2009-2021 by Code Magus Limited. All rights reserved.
 [Contact: stephen@codemagus.com].
Usage: orkadmin [OPTION...]
-h, --central-host=<host or ipaddr>     Okhestra central database server host
-p, --central-port=<port>               Okhestra central database server port
-a, --central-admin-user=<user>         Admin user
-A, --central-admin-password=<password> Admin user password
--delete-user                           Delete a user, require --user-email
--login-test                            Login test, uses --central-admin-user
                                        and --central-admin-password
--list-user                             List user(s), optional --user-email
--reset-user                            Reset user password - require --user-email
--set-permissions                       Set user permissions, requires --user-email
                                        and --permissions
--get-user_conf_perm                    Get user configuration permission, requires
                                        --user-email and --configuration-name
--set-user_conf_perm                    Set user configuration permission, requires
                                        --user-email,
                                        --configuration-name and --permissions
--delete-user_conf_perm                 Delete user configuration permission,
                                        requires --user-email and
                                        --configuration-name
--register-user                         Register a user, req --user-email and
                                        --first_name and --last-name
--user-email=<email>                    User email (ALL := NULL for --list-user))
--first-name=<name>                     User first name
--last-name=<surname>                   User last name
--permissions=<single digit>            User permissions
--list-configuration                    List configuration(s) optional
                                        --configuration-name
--list-start-up-config                  List the ORKSTRPR configuration for a
                                        configuration  name
--list-dashboard-index                  List the configuration index for a dashboard
```

```
--add-configuration                     Add an Orkhestra configuration
--start-up-config                       Create/Update the ORKSTRPR configuration
--parse-start-up-config                 Request an upload and parse of the loaded
                                        ORKSTRPR configuration
--add-dashboard                         Add an Orkhestra dashboard configuration
--add-dashboard-filter                  Add an Orkhestra dashboard filter and
                                        save it
--delete-configuration                  Delete an Orkhestra configuration
--delete-dashboard                      Delete an Orkhestra dashboard configuration
--upload-dashboard                      Upload a orkhestra dashboard source
--download-dashboard                    Download a orkhestra dashboard source
--configuration-name=<name>             Configuration name (ALL := NULL for
                                        --list-configuration)
--configuration-title=<title>           Configuration title
--start-process-host=<host or ipaddr>   Configuration start process name
--start-process-group=<name>            Configuration start process group
--start-process-name=<name>             Configuration start process name
--start-process-config=<file name>      Configuration start process configuration
--dashboard-name=<name>                 Configuration dashboard name
--dashboard-filter-name=<name>          Dashboard filter name
--dashboard-filter-cset=<file name>     Dashboard filter control set file name
--dashboard-title=<title>               Configuration dashboard title
--dashboard-source=<filename>           Dashboard source
--active                                List all active(running) orkhestras
--log-event                             Log an event to the server
--event-originator={orkadmin|<name>}    Event originator
--event-topic=<topic>                   Event topic
--event-message=<message>               Event message
--list-events                           List events, filter --event-age,
                                        --event-originator-pattern
--event-age={24|<hours>}                Event list age filter
--event-originator-pattern={NULL|<sqlite like pattern>}
                                        Event list originator filter
--event-format-is-json                  Default list format is CSV
--export                                Export a configuration
--export-prefix=<prefix>                Prefix for exported script names
--export-to-conf=<new name>             Exported configuration name
--export-from-module=<NFT module>       Exported from NFT (CVS) module
--export-to-module=<NFT module>         Exported to NFT (CVS) module
--export-from-start-group=<name>        Exported from start process group name
--export-to-start-group=<name>          Exported to start process group name
-v, --verbose                           Verbose

Help options:
-?, --help                              Show this help message
    --usage                             Display brief usage message
```

3 *Orkhestra* Central Address

The following two options specify the connection details to the *Orkhestra* central server:

- `-h, --central-host`
  Host name or IP ADDRESS for the *Orkhestra* central server.
- `-p, --central-port`
  Port number for the *Orkhestra* central server.

If the environment variable `ORKHESTRA_CENTRAL_LOG` is set to `<HOST>:<PORT>` then it will be used if the host name and port are not given on the command line.

For example:

```
export ORKHESTRA_CENTRAL_LOG=network.codemagus.com:60002
```

4 *Orkhestra* Central Credentials

The following two options specify the credentials for accessing the *Orkhestra* Central server:

- `-a, --central-admin-user`
  User ID (Email) with administration rights.
- `-A, --central-admin-password`
  User password.

5 User Maintenance

All the actions is this section require credentials, see section 4 on page 5.

5.1 List

List user(s) details.

Command line options:

- `--list-user`
  Request user(s) details to be displayed.
- `--user-email=<email>`
  Optional user ID to list, if not specified all users will be listed.

Example:

List user `Some2.Body2@someplace.com`:

```
orkadmin
   --central-host=${CENTRAL_HOST} --central-port=${CENTRAL_PORT}
   --central-admin-user=${ADMIN_ID} --central-admin-password=${ADMIN_PASSWD}
   --list-user
   --user-email=Some2.Body2@someplace.com
   |jq
```

Response:

```
{
  "DefaultPermissions": 1,
  "Users": [
    {
      "eMail": "Some2.Body2@someplace.com",
      "FirstName": "Some2",
      "LastName": "Body2",
      "Secret": "g37JFClFNaD1ucttQlnfM4whi44=",
      "Permissions": 1,
      "Registered": "2021-02-04 10:39:57",
      "LastLogon": "2023-10-23 15:40:52",
      "Configurations": [
        {
          "Name": "NCRATM",
          "Permissions": 2
        },
        {
          "Name": "TermAppISONFT",
          "Permissions": 0
        }
      ]
    }
  ]
}
```

Note `Some2.Body2@someplace.com` can view and monitor all configurations except for the following overrides:

- `NCRATM`
  Has full access to this configuration.
- `TermAppISONFT`
  Can not monitor this configuration.

See section 6 on page 10 for the documentation on permissions.

5.2 Register

Register a new user.

Command line options:

- `register-user`
  Request user registration.
- `--user-email=<email>`
  User ID to to register.
- `--first-name=<name>`
  Users first name..
- `--last-name=<surname>`
  Users last name.

Example:

Register a user with ID `admin@codemagus.com`:

```
orkadmin
   --central-host=${CENTRAL_HOST} --central-port=${CENTRAL_PORT}
   --central-admin-user=${ADMIN_ID} --central-admin-password=${ADMIN_PASSWD}
   --register-user
   --user-email=john@doe.com
   --first-name=john
   --last-name=doe
```

Response:

```
{
  "Result": "Success",
  "User": "john@doe.com",
  "Password": "A35F98C0B31211EBA1A23F1EEBFC4F69"
}
```

5.3 Delete

Delete a user.

Command line options:

- `--delete-user`
  Request user deletion.
- `--user-email=<email>`
  User ID to to delete.

Example:

Delete user `john@doe.com`:

```
orkadmin
   --central-host=${CENTRAL_HOST} --central-port=${CENTRAL_PORT}
   --central-admin-user=${ADMIN_ID} --central-admin-password=${ADMIN_PASSWD}
   --delete-user
   --user-email=john@doe.com
   |jq
```

Response:

```
{
  "Result": "Success",
}
```

5.4 Reset

Reset a user password.

Command line options:

- `--reset-user`
  Request user reset.
- `--user-email=<email>`
  User ID to reset the password for.

Example:

Reset the password for user `john@doe.com`:

```
orkadmin
   --central-host=${CENTRAL_HOST} --central-port=${CENTRAL_PORT}
   --central-admin-user=${ADMIN_ID} --central-admin-password=${ADMIN_PASSWD}
   --reset-user
   --user-email=john@doe.com
   |jq
```

Response:

```
{
  "Result": "Success",
  "Response": "A generated password has been emailed to john@doe.com"
}
```

6 Permissions

All the actions is this section require credentials, see section 4 on page 5.

User access to *Orkhestra* Central via the WEB is controlled by user permissions. This is a single digit value:

- `0`
  Only allow the user to see the list of configurations and the dashboards list for a configuration.
- `1`
  The user can view every thing, but can't submit any commands to *Orkhestra*. or take a leading role.
- `2 to 9`
  The user has full access. In addition to monitoring the user can control this configuration with the *Orkhestra* commands available.

When a user logs in, the user permissions are set to the defined value for the user. If this value is zero, it will be set to the default user permissions as defined by the *Orkhestra* central database server (section B.2 on page 39).

When a user selects a configuration, the user's configuration permissions table is referenced to check the user has an explicit permission for the configuration selected. If the row exists in the explicit configurations permissions table, then the corresponding value of the permission is used, otherwise the value defined for the user is used. This will mean that flicking from configuration to configuration, the permissions value could change between the various explicit configuration permission values or between an explicit configuration permission value and the defined user permission value.

6.1 User defined permissions

Change user defined permissions. Note when a user is registered this value is set to zero.

Command line options:

- `--set-permissions`
  Request change of user permissions.
- `--user-email=<email>`
  User ID .
- `--permissions=<single digit>`
  User permissions in the range 0 to 9.

Example:

Change permissions for user `john@doe.com` to 2.:

```
$ orkadmin
   --central-host=${CENTRAL_HOST} --central-port=${CENTRAL_PORT}
   --central-admin-user=${ADMIN_ID} --central-admin-password=${ADMIN_PASSWD}
   ----set-permissions
   --user-email=john@doe.com
   --permissions=2
```

Response:

```
{"Result": "Success",}
```

6.2 Set configuration permissions

Set user permissions for a configuration.

Command line options:

- `set-user_conf_perm`
  Request setting of a user configuration permissions.
- `--user-email=<email>`
  User ID .
- `--configuration-name=<name>`
  Name of the configuration.
- `--permissions=<single digit>`
  Permissions in the range 0 to 9.

Example:

Set permissions for user `Some2.Body2@someplace.com` configuration `NCRATM` to 2.:

```
$ orkadmin
   --central-admin-user=${ADMIN_USER} --central-admin-password=${ADMIN_PASSWD}
   --set-user_conf_perm
   --user-email=Some2.Body2@someplace.com
   --configuration-name=NCRATM
   --permissions=2
```

Response:

```
{
  "Result": "Success",
  "User": "Some2.Body2@someplace.com",
  "Message": "Updated user Some2.Body2@someplace.com, configuration NCRATM permissions to 2"
}
```

6.3 Delete configuration permissions

Delete user permissions for a configuration.

Command line options:

- `delete-user_conf_perm`
  Request to delete a user's configuration permissions.
- `--user-email=<email>`
  User ID .
- `--configuration-name=<name>`
  Name of the configuration.

Example:

Delete permissions for user `Some2.Body2@someplace.com` configuration `NCRATM` :

```
$ orkadmin
   --central-admin-user=${ADMIN_USER} --central-admin-password=${ADMIN_PASSWD}
   --delete-user_conf_perm
   --user-email=Some2.Body2@someplace.com
   --configuration-name=NCRATM
```

Response:

```
{"Result": "Success"}
```

7 Configurations

All the actions in this section, except for list (section 7.1 on page 14) require credentials, see section 4 on page 5.

7.1 List

7.1.1 Active

This command lists all active, running orkhestra instances registered with *Orkhestra* Central.

Command line options:

- `--active`
  List all active, running orkhestras.

Example:

```
orkadmin
   --central-host=${CENTRAL_HOST} --central-port=${CENTRAL_PORT}
   --active
   2>/dev/null
```

Response:

```
ITAAuth_xqacml3
 PID = 17795
 Machine MoneyNFT
 Group MoneyNFT
 echo 45306 > .orkcmdport
 echo itaauth@XQACML3:/home/app/itaauth/CodeMagus/MoneyNFT/testdata/scripts
       > .orkcmdhost
TermAppISONFT
 PID = 3746
 Machine trmapp
 Group trmapp
 echo 33951 > .orkcmdport
 echo cmlstress@codemagus:/opt/data/home/cmlstress/CodeMagus/TermAppISONFT/testdata/scripts
       > .orkcmdhost
```

7.1.2 Details

List configuration(s) details.

Command line options:

- `--list-configuration`
  Request configuration(s) details to be displayed.
- `--configuration-name=<name>`
  Optional configuration name to list, if not specified all configurations will be listed.

Example:

List configuration `NCRATM`:

```
orkadmin
   --central-host=${CENTRAL_HOST} --central-port=${CENTRAL_PORT}
   --list-configuration
   --configuration-name=NCRATM
   2>/dev/null
```

Response:

```
{
  "Result": "Success",
  "Name": "NCRATM",
  "Title": "NCR ATM NFT",
  "Index": 40,
  "Status": "Registered",
  "PidHost": null,
  "Machine": "ncratm",
  "Group": "ncratm",
  "Host": "cmlqa0.it.nednet.co.za",
  "StartProcessGroup": "NCRATMNFT",
  "StartProcessName": "orkhestra",
  "NextDashbordIndex": 4,
  "Dashboards": [
    {
      "Name": "dbcmd_ncratm",
      "Index": 0,
      "Type": "System",
      "Title": "Totals",
      "SourceTimeStamp": "2021-10-04 11:31:14"
    },
.
.
.
  ]
}
```

7.2 Add

Add *Orkhestra* configuration details.

Command line options:

- `--add-configuration`
  Request configuration details to be added.
- `--configuration-name=<name>`
  Name of the configuration.
- `--configuration-title=<title>`
  Orkhestra configuration title.

Example:

Add configuration `NCRATM`:

```
orkadmin \
   --central-host=${CENTRAL_HOST} --central-port=${CENTRAL_PORT}
   --central-admin-user=${ADMIN_USER} --central-admin-password=${ADMIN_PASSWORD}
   --add-configuration
   --configuration-name="NCRATM"
   --configuration-title="NCR ATM NFT"
```

Response:

```
{"Result":"Success"}
```

7.3 Delete

Delete an *Orkhestra* configuration.

Command line options:

- `--delete-configuration`
  Request configuration details to be deleted.
- `--configuration-name=<name>`
  Name of the configuration to be deleted.

Example:

Delete configuration `NCRATM`:

```
orkadmin \
   --central-host=${CENTRAL_HOST} --central-port=${CENTRAL_PORT}
   --central-admin-user=${ADMIN_USER} --central-admin-password=${ADMIN_PASSWORD}
   --delete-configuration
   --configuration-name="NCRATM"
```

Response:

```
Submitted
```

8 ORKSTRPR Configurations

ORKSTRPR refers to the CML attachments to processes daemon, see **orkhestra_attproc**: Orkhestra: Attachments to Processes Version 1[2].

Only action create in this section requires credentials, see section 4 on page 5.

8.1 List

List the ORKSTRPR startup configuration details.

Command line options:

- `--list-start-up-config`
  Request ORKSTRPR startup details to be displayed.
- `--configuration-name=<name>`
  Configuration name to list the startup details.

Example:

List configuration `NCRATM`:

```
orkadmin
   --central-host=${CENTRAL_HOST} --central-port=${CENTRAL_PORT}
   --list-start-up-config \
   --configuration-name=NCRATM
```

Response:

```
ORKSTRPR startup details for NCRATM:
   Host          localhost
   Process group NCRATMNFT
   Process name  orkhestra
Process configuration:
group NCRATMNFT
   {
.
.
.
   }
```

8.2 Create

Create the ORKSTRPR startup configuration details. If the details exist, they will be updated.

Command line options:

- `--start-up-config`
  Request ORKSTRPR configuration to be added.
- `--configuration-name=<name>`
  Name of the configuration. If it exists it will be updated.
- `--start-process-host=<host/ipaddr>`
  The host name or IPADDRESS this *Orkhestra* configuration runs on.
- `--start-process-group=<name>`
  Defines the ORKSTRPR group name.
- `--start-process-name=<name>`
  Defines the ORKSTRPR process name (to start *Orkhestra* with this configuration).
- `--start-process-config=<file name>`
  Optional, specifies a file that contains the ORKSTRPR startup variables.

Example:

Create/update configuration `NCRATM`:

```
orkadmin \
   --central-host=${CENTRAL_HOST} --central-port=${CENTRAL_PORT}
   --central-admin-user=${ADMIN_USER} --central-admin-password=${ADMIN_PASSWORD}
   --add-configuration
   --configuration-name="NCRATM"
   --start-process-host=localhost
   --start-process-group=NCRATMNFT
   --start-process-name=orkhestra
   --start-process-config=${SCRIPTS}/ncratm.conf
```

Response:

```
{"Result":"Success"}
```

8.3 Parse

Request *Orkhestra* Central to upload the ORKSTRPR startup configuration source to the startup daemon. The daemon will parse the source and respond with an appropriate message as to the outcome of the parsing.

Command line options:

- `--parse-start-up-config`
  Request ORKSTRPR configuration to be uploaded to the startup daemon for parsing.
- `--configuration-name=<name>`
  Name of the configuration.

Example:

Upload the startup configuration for `NCRATM`:

```
orkadmin \
   --central-host=${CENTRAL_HOST} --central-port=${CENTRAL_PORT} \
   --parse-start-up-config \
   --configuration-name="NCRATM" \
```

Example of a good response:

```
Parsing config for NCRATM: remote start process daemon(localhost:62200):
From Server jvubuntu: orkstrprd():
Parsed, startup group NCRATMNFT Description NCRATMNFT
```

Example of a parse error response:

```
Parsing config for NCRATM: remote start process daemon(localhost:62200):
orkstrpr_upload_config() Error: From Server jvubuntu: syntax error
   --> at line 30 column 6
```

9 Dashboards

All the actions is this section require credentials, see section 4 on page 5.

9.1 Add

Add a dashboard to a configuration.

Command line options:

- `--add-dashboard`
  Request add of a dashboard.
- `--configuration-name=<name>`
  Name of the configuration containing the dashboard.
- `--dashboard-name=<name>`
  Name of the dashboard.
- `--dashboard-title=<title>`
  Dashboard title.

Example:

Add dashboard `web_dbcmd_totals` for configuration `ACPNFT`.

```
orkadmin \
   --central-host=${CENTRAL_HOST} --central-port=${CENTRAL_PORT}
   --central-admin-user=${ADMIN_USER} --central-admin-password=${ADMIN_PASSWORD}
   --add-dashboard
   --configuration-name="ACPNFT"
   --dashboard-name="web_dbcmd_totals"
   --dashboard-title="Totals"
```

Response:

```
{"Result":"Success"}
```

9.2 Delete

Delete a dashboard.

Command line options:

- `--delete-dashboard`
  Request deletion of a dashboard.
- `--configuration-name=<name>`
  Name of the configuration containing the dashboard.

- `--dashboard-name=<name>`
  Name of the dashboard.

Example:

Delete dashboard `web_dbcmd_totals` from configuration `ACPNFT`.

```
orkadmin \
   --central-host=${CENTRAL_HOST} --central-port=${CENTRAL_PORT}
   --central-admin-user=${ADMIN_USER} --central-admin-password=${ADMIN_PASSWORD}
   --delete-dashboard
   --configuration-name="ACPNFT"
   --dashboard-name="web_dbcmd_totals"
```

Response:

```
Submitted
```

9.3 Filtering

Add a filter to a specified dashboard of the specific *Orkhestra* configuration.

Command line options:

- `--add-dashboard-filter`
  Add an Orkhestra dashboard filter and save it.
- `--configuration-name=<name>`
  Name of the configuration containing the dashboard.
- `--dashboard-name=<name>`
  Name of the dashboard.
- `--dashboard-filter-name=<name>`
  Name of the dashboard filter to be saved.
- `--dashboard-filter-cset=<file name>`
  File name containing the filter specification.

Example:

Save filter `Colours22` for dashboard `dbcmd_ncratm` from configuration `NCRATM`.

```
orkadmin \
   --central-host=${CENTRAL_HOST} --central-port=${CENTRAL_PORT}
   --central-admin-user=${ADMIN_USER} --central-admin-password=${ADMIN_PASSWORD}
   --add-dashboard-filter\
   --configuration-name=NCRATM
   --dashboard-name=dbcmd_ncratm
   --dashboard-filter-name=Colours22
   --dashboard-filter-cset=/home/jan/tmp/A_NCRATM_Colours20.cset
```

Response:

```
Saved Filter Colours22 for dashboard dbcmd_ncratm
```

10 Dashboard Source

All the actions in this section require credentials, see section 4 on page 5.

10.1 Upload

Upload the source for a dashboard.

Command line options:

- `--upload-dashboard`
  Request upload of a dashboard source.
- `--configuration-name=<name>`
  Name of the configuration containing the dashboard.
- `--dashboard-name=<name>`
  Name of the dashboard.
- `--dashboard-source=<file name>`
  File name containing the dashboard source.

Example:

Upload the source for dashboard `dbcmd_ncratm` for configuration `NCRATM` from `"${HOME}/CodeMagus/NCRATMNFT/testdata/orkhestra/dbcmd_ncratm"`.

```
orkadmin \
   --central-host=${CENTRAL_HOST} --central-port=${CENTRAL_PORT}
   --central-admin-user=${ADMIN_USER} --central-admin-password=${ADMIN_PASSWORD}
   --upload-dashboard
   --configuration-name="NCRATM"
   --dashboard-name="dbcmd_ncratm"
   --dashboard-source="${HOME}/CodeMagus/NCRATMNFT/testdata/orkhestra/dbcmd_ncratm"
```

Response:

```
Success
```

10.2 Download

Download the source of a dashboard to `stdout`.

Command line options:

- `--download-dashboard`
  Request download of a dashboard source.
- `--configuration-name=<name>`
  Name of the configuration containing the dashboard.

- `--dashboard-name=<name>`
  Name of the dashboard.

Example:

Download dashboard `dbcmd_ncratm` source for configuration `NCRATM` to `stdout`.

```
orkadmin
   --central-host=${CENTRAL_HOST} --central-port=${CENTRAL_PORT}
   --central-admin-user=${ADMIN_USER} --central-admin-password=${ADMIN_PASSWORD}
   --download-dashboard
   --configuration-name="NCRATM"
   --dashboard-name="dbcmd_ncratm"
   2>/dev/null
```

11 Events

Events are logged to *Orkhestra* central from various sources. This include *Orkhestra*, *Orkhestra* control programs, *orkadmin* and *Orkhestra* central.

11.1 Log an event

Command line options:

- `--log-event`
  Log an event to *Orkhestra* central..
- `--event-originator={orkadmin|<name>}`
  Name of the event originator. If not specified, *orkadmin* will be used.
- `--event-topic`
  The event topic.
- `--event-message=<message>`
  The event Message.

Example:

```
$ orkadmin --log-event
   --event-originator=me
   --event-topic=something
   --event-message="Bla Bla" 2>dev/null

Event logged
```

11.2 List events

Command line options:

- `--list-events`
  List events.
- `--event-age={24|<hours>}`
  Event age filter specified in hours. The default, if not specified, is 24 hours.
- `--event-originator-pattern={NULL|<sqlite like pattern>`
  Optional event originator filter.
- `--event-format-is-json`
  List response in json format versus comma delimited.

Example:

Comma delimited formatted response:

```
$ orkadmin --list-events --event-originator-pattern='%ja%' 2>/dev/null
TimeStamp,Originator,OriginatorPid,OriginatorHost,OriginatorIPADDR,Severity,Topic,Message
2023-10-24 14:30:08,jan,0,,127.0.0.1,0,abc,BlaBla
2023-10-24 14:30:25,jan,0,,127.0.0.1,0,abc,BlaBla
2023-10-24 14:43:00,jan,0,,127.0.0.1,0,abc,BlaBla
2023-10-24 14:45:06,jan,0,,127.0.0.1,0,abc,BlaBla
```

json formatted response:

```
$ orkadmin --list-events --event-originator-pattern='%ja%' --event-format-is-json
   2>/dev/null|jq
{
  "Result": "Success",
  "Events": [
    {
      "TimeStamp": "2023-10-24 14:30:08",
      "Originator": "jan",
      "OriginatorPid": 0,
      "OriginatorHost": "",
      "OriginatorIPADDR": "127.0.0.1",
      "Severity": 0,
      "Topic": "abc",
      "Message": "BlaBla"
    },
    ...
    {
      "TimeStamp": "2023-10-24 14:45:06",
      "Originator": "jan",
      "OriginatorPid": 0,
      "OriginatorHost": "",
      "OriginatorIPADDR": "127.0.0.1",
      "Severity": 0,
      "Topic": "abc",
      "Message": "BlaBla"
    }
  ]
}
```

12 Export

Export an *Orkhestra* Central configuration and create scripts to clone the configuration and the NFT module.

Command line options:

- `--export`
  Export a configuration.
- `--configuration-name=<name>`
  Name of the *Orkhestra* Central configuration to be exported.
- `--export-to-conf=<new name>`
  Name of the cloned *Orkhestra* Central configuration when importing the new configuration. This is also used as a base name for the configuration files and scripts along with `--export-prefix`.
- `--export-prefix=<prefix>>`
  Prefix to be used for exported script file names.
- `--export-from-module=<NFT module>`
  Name of the *Orkhestra* NFT (CVS) module.
- `--export-to-module=<NFT module>`
  Name of the cloned *Orkhestra* NFT (CVS) module.
- `--export-from-start-group=<name>`
  Name of the *Orkhestra* remote start process group.
- `--export-to-start-group=<name>`
  Name of the cloned *Orkhestra* remote start process group.

Example:

Export *Orkhestra* Central configuration `NCRATM` to be cloned as `NCRATM_clone1`.

```
orkadmin \
   --central-host=${CENTRAL_HOST} --central-port=${CENTRAL_PORT}
   --central-admin-user=${ADMIN_USER} --central-admin-password=${ADMIN_PASSWORD}
   --export
   --configuration-name=NCRATM --export-to-conf=NCRATM_clone1
   --export-from-module=NCRATMNFT --export-to-module=NCRATM_clone1NFT
   --export-from-start-group=NCRATMNFT --export-to-start-group=NCRATM_clone1NFT
   2>/dev/null
```

Response:

```
Configuration NCRATM:
  Title             "NCRATM bla"
  Start Process:
    Host            "localhost;bruinkop"
    Group           "NCRATMNFT"
    Name            "orkhestra"
    Configuration  Yes
  Dashboards:
    dbcmd_ncratm
      Title               "Totals"
      Source size         32135
    Colours20
      Parent              dbcmd_ncratm
      Control set size    676
    Colours21
      Parent              dbcmd_ncratm
      Control set size    676
Creating Clone load script:
   /home/jan/tmp/A_NCRATM_clone1_load.s
Creating Clone duplicate script for NFT module:
    /home/jan/tmp/A_NCRATM_clone1_copy_nft_module.sh
Scripts and data files created:
-rw-r--r-- 1 jan users   676 Feb  2 09:42 /home/jan/tmp/A_NCRATM_clone1_Colours20.cset
-rw-r--r-- 1 jan users   676 Feb  2 09:42 /home/jan/tmp/A_NCRATM_clone1_Colours21.cset
-rwxr-xr-x 1 jan users   263 Feb  2 09:42 /home/jan/tmp/A_NCRATM_clone1_copy_nft_module.sh
-rw-r--r-- 1 jan users 32135 Feb  2 09:42 /home/jan/tmp/A_NCRATM_clone1_dbcmd_ncratm.dboard
-rwxr-xr-x 1 jan users  1962 Feb  2 09:42 /home/jan/tmp/A_NCRATM_clone1_load.sh
-rw-r--r-- 1 jan users  1371 Feb  2 09:42 /home/jan/tmp/A_NCRATM_clone1_orkstrpr.conf
-rw-r--r-- 1 jan users   179 Feb  2 09:42 /home/jan/tmp/A_NCRATM_clone1.sed
```

A How To

A.1 Generic*DashBoard*

This example shows the generation of a full and flat dashboard based on the state machine definition. This ensures that a dashboard with the default name of `Generic` is always available and correct.

For demonstration purposes the `TermAppISONFT` *NFT* is used in the following example.

A.1.1 Exporting the *State Machine* and the generic *DashBoard*

Run the script `export_state_machine.sh` to invoke *Orkhestra* to export the *State Machine*. The script defines the exported file prefix and command file that loads the machine to be exported. Once done *Orkhestra* will terminate.

A.1.2 **`export_state_machine.sh`**

Executes *Orkhestra* with the following options:

- `--command='"export_state_machine.cmd"'`
  This file holds the *Orkhestra* commands to load the machine.
- `--export_machine=exported_`
  Instructs orkhestra to export the *State Machine* and store it in files with the given prefix. The files are:
  - machine `exported_machine_trmapp.csv`
  - generic dashboard `exported_dashboard_trmapp.cmd`
- `--background`
  Run *Orkhestra* in the back ground.

A.1.3 **`export_state_machine.cmd`**

Contains the *Orkhestra* commands to define the required *State Machine Control Program*, load the *State Machine* and `shutdown` *Orkhestra*.

A.1.4 Running the Export

```
[orkhestra@load0 scripts]$ ./export_state_machine.sh
```

```
.
.
.
define control_program trmapp (path("trmappcp") parameter("dummy "))
Response: Control program trmapp defined;
load machine "../orkhestra/trmapp.mch"
Response: Parsing Machine definition file ../orkhestra/trmapp.mch;
Response: Machine trmapp defined;
********************************************************
Exporting machine trmapp to exported_machine_trmapp.csv
Creating generic dashboard exported_dashboard_trmapp.cmd
********************************************************
shutdown
```

A.1.5 Post export actions

Fix the `group` name in the *DashBoard*; At the export stage, the *State Machine* has not been started and *Orkhestra* has no knowledge of the `group` name, and so uses the *State Machine* name appended with `'???'` as the `group` name. In this instance the `group` name should just be `trmapp`, so the `'???'` portion is just removed.

A.1.6 `export_upload_gendb.sh`

Configure the generic *DashBoard* and uploads it:

```
ADMIN_USER=admin@codemagus.com
ADMIN_PASSWORD=FFCF098C0B5E11EE9C0A27C536683CB6

SOURCE=exported_dashboard_trmapp.cmd
CONF=TermAppISONFT
DBOARD=Generic

echo Add configuration ${CONF} dashboard ${DBOARD}
echo "*** Ignore the error if machine is defined ***"
orkadmin \
   --central-admin-user=${ADMIN_USER} --central-admin-password=${ADMIN_PASSWORD} \
   --add-dashboard \
   --configuration-name="${CONF}" \
   --dashboard-name="${DBOARD}" \
   --dashboard-title="${DBOARD}" \
   2>/dev/null

# NOTE the SM ’group’ name must be fixed.
# As the group name is the same as the machine name, just remove the "???"
#
sed --in-place=".bak" "s/???//g" ${SOURCE}

echo Uploading ${CONF} dashboard ${DBOARD} from ${SOURCE}
orkadmin \
   --central-admin-user=${ADMIN_USER} --central-admin-password=${ADMIN_PASSWORD} \
   --upload-dashboard \
   --configuration-name="${CONF}" \
   --dashboard-name="${DBOARD}" \
   --dashboard-source="${SOURCE}" \
   2>/dev/null
```

A.1.7 Execution

```
[orkhestra@load0 scripts]$ ./export_upload_gendb.sh
Add configuration TermAppISONFT dashboard Generic
*** Ignore the error if machine is defined ***
Error: Configuration TermAppISONFT, dashboard Generic is defined!
Uploading TermAppISONFT dashboard Generic from exported_dashboard_trmapp.cmd
Success
```

A.2 Standard filtered *DashBoards*

A.2.1 Overview

Standard filtered *DashBoards* require a generic dashboard to have been configured, see section A.1 on page 30 for more information.

A script *orkadmin_generic_filter_dashboards.sh* is provided to help generate a set of generic filter dashboards. A filter dashboard is different to the full and flat dashboard in that it may only show some information and possibly in a specific order. The location of this script is
/home/orkhestra/CodeMagus/OrkhestraCentral/testdata/dashboards.

This script generates a further script, named by the requester, to create three *filter* dashboards:

- `'OK'`
  Filter for good responses, sorted descending on `Resp.EWMA`
- `'Not OK'`
  Filter for bad/deny responses, sorted descending on `Resp.EWMA`
- `'Slow Response'`
  Filter for all responses, sorted descending on `Resp.EWMA`

In each of these scripts there is a placeholder for the field `Regex` that makes it obvious what needs to be changed to include the correct NFT responses for that dashboard.

A.2.2 Synopsis

```
.../dashboards$ ./orkadmin_generic_filter_dashboards.sh -h
Usage: orkadmin_generic_filter_dashboards.sh [OPTION...]
  -h            Show this help message
  -c <name>     OrkhestralCentral configuration name
  -s <name>     Create a filter set script with this name
```

A.2.3 Example

Following is an example of how to create the three filter dashboards for the `Generic` dashboard. This HowTo uses the `TermAppISONFT` configuration.

Creating the user script

```
jan@bruinkop:~dashboards$ ./orkadmin_generic_filter_dashboards.sh
   -c TermAppISONFT -s ~/tmp/TermAppISONFT_gen_filters.sh

Configuration TermAppISONFT
Creating /home/jan/tmp/TermAppISONFT_gen_filters.sh
jan@bruinkop:~dashboards
```

The script generated is shown in section A.2.4 on page 35.

A.2.4 TermAppISONFT_gen_filters.sh

```
#
# Generated by orkadmin_generic_filter_dashboards.sh
# Mon 24 Jul 15:10:23 BST 2023

cd /home/jan/ecosystem/CodeMagus/OrkhestraCentral/testdata/dashboards
. ./orkadmin_setparms.sh -u ADMIN
CONF=TermAppISONFT
PARENT=Generic
PARENT_IX=$(orkadmin --list-dashboard-index --configuration-name=${CONF} --dashboard-name=${PARENT} 2>/dev
if [ "${PARENT_IX}" = "x" ]; then
   echo "Error Configuration ${CONF} dashboard ${PARENT} NOT defined"
   exit 16
fi
export USER
export CONF
export PARENT_IX
./orkadmin_login_check.sh
RC=$?
if [ ${RC} != "0" ];then
   echo Aborted
   exit 16
fi

./orkadmin_show.sh >/dev/null
RC=$?
if [ ${RC} != "0" ];then
   echo Aborted
   exit 16
fi

#
# Slow Response
#
#  Note the filter ’Regex’ is generic and needs to be changed to select
#  the required metrics.
#
export FREQD=${REQD}
cat >${FREQD} <<EOF
{
 "Save": true,
 "Name": "Slow Response",
 "Control":{
   "Sort": [
    {
    "Dir": "Dsc",
    "Field": "Resp.EWMA"
    }],
   "Filters": [
    {
    "Type": "Match",
    "Field": "Metric.Name",
    "Regex": "*"
     }],
   "PrintColours": [
    {
    "Value": "green",
    "When": "Resp.Interval<1000"
    },{
     "Value": "yellow",
     "When": "(Resp.Interval>=1000) and (Resp.Interval<1500)"
    },{
     "Value": "orange",
     "When": "(Resp.Interval>=1500) and (Resp.Interval<2000)"
    },{
     "Value": "red",
     "When": "Resp.Interval>=2000"
    }]
}}
EOF

./orkadmin_set.sh
#
# OK
#
#  Note the filter ’Regex’ is generic and needs to be changed to select
#  the required metrics.
#
export FREQD=${REQD}
cat >${FREQD} <<EOF
{
 "Save": true,
 "Name": "OK",
 "Control":{
   "Sort": [
    {
    "Dir": "Dsc",
    "Field": "Resp.EWMA"
    }],
   "Filters": [
    {
    "Type": "Match",
    "Field": "Metric.Name",
    "Regex": "*"
     }],
   "PrintColours": [
    {
    "Value": "green",
    "When": "Resp.Interval<1000"
    },{
     "Value": "yellow",
     "When": "(Resp.Interval>=1000) and (Resp.Interval<1500)"
    },{
     "Value": "orange",
     "When": "(Resp.Interval>=1500) and (Resp.Interval<2000)"
    },{
     "Value": "red",
     "When": "Resp.Interval>=2000"
    }]
}}
EOF

./orkadmin_set.sh
#
# Not OK
#
#  Note the filter ’Regex’ is generic and needs to be changed to select
#  the required metrics.
#
export FREQD=${REQD}
cat >${FREQD} <<EOF
{
 "Save": true,
 "Name": "Not OK",
 "Control":{
   "Sort": [
    {
    "Dir": "Dsc",
    "Field": "Resp.EWMA"
    }],
   "Filters": [
    {
    "Type": "Match",
    "Field": "Metric.Name",
    "Regex": "*"
     }],
   "PrintColours": [
    {
    "Value": "green",
    "When": "Count.Interval=0"
    },{
     "Value": "red",
     "When": "Count.Interval>0"
    }]
}}
EOF

./orkadmin_set.sh
rm -f tmp/re*
```

B *Orkhestra* Central SPA back end

The *Orkhestra* Central SPA server is *orksrvrr* and the database database server is *orksrvr*

B.1 *orksrvrr*

Orkhestra Central SPA server.

B.1.1 Synopsis

```
jan@bruinkop:~/software/orkcntrl$ orksrvrr --help
Code Magus Limited Orkhestra Central V3: build 2023-10-23-16.08.31
[orksrvrr] $Id: orkadmin.tex,v 1.18 2023/11/07 10:14:14 hayward Exp $
Copyright (c) 2009-2021 by Code Magus Limited. All rights reserved.
 [Contact: stephen@codemagus.com].
Usage: orksrvrr [OPTION...]
 -p, --port={8888|<port>}                http(s) rest API port
 -t, --ork-inactive-time=[90|<seconds>]  Reset orkhestra circuit if idle time
                                         exceeds this
 -v, --verbose                           Verbose

Help options:
 -?, --help                              Show this help message
      --usage                             Display brief usage message
```

Where:

- `-p, --port={8888|<port>}`
  WEB SPA connection port. The default port is `8888`.
- `-t,--ork-inactive-time=[90|<seconds>]`
  To prevent an *Orkhestra* connection circuit from going stale the circuit is reset after an inactive threshold. The default is 90 seconds.
- `-v, --verbose`
  When specified, *orksrvrr* operates in a verbose manner.

B.2 *orksrvr*

Orkhestra Central database server.

B.2.1 Synopsis

```
jan@bruinkop:~/software/orkcntrl$ orksrvr --help
Code Magus Limited Orkhestra Central V3: build 2023-10-20-13.43.53
[orksrvr] $Id: orkadmin.tex,v 1.18 2023/11/07 10:14:14 hayward Exp $
Copyright (c) 2009-2021 by Code Magus Limited. All rights reserved.
 [Contact: stephen@codemagus.com].
Usage: orksrvr [OPTION...]
 -d, --database=<DBname>                         SQLite database name
 -p, --port=[ORKHESTRA_CENTRAL_LOG|<port>]       Event log listen port
 -q, --query-port=[ORKHESTRA_CENTRAL_QUERY|<port>]  Query listen port
 -h, --register-script=<file name>               Script to email user credentials
 -P, --rdaemon-port=<port>                       remote start process daemon port
 -t, --ork-inactive-time=[90|<seconds>]          Reset orkhestra circuit if idle time exceeds this
 -s, --default-permissions=[1|<number>]          If user permissions set to zero
 -v, --verbose                                   Verbose

Help options:
 -?, --help                                      Show this help message
     --usage                                     Display brief usage message
```

Where:

- `-d, --database=<DBname>`
  Sqlite database name.

- `--port=[ORKHESTRA_CENTRAL_LOG|<port>]`
  Event log listen port. *orkadmin* uses this connection for connecting to the database server.

- `--query-port=[ORKHESTRA_CENTRAL_QUERY|<port>]`
  Query listen port. This for a telnet connection to do database enquiries.

- `--register-script=<file name>`
  Script for sending the user credentials on user registration.

- `--rdaemon-port=<port>`
  CML attachments to processes daemon connection port, see **orkhestra_attproc**: Orkhestra: Attachments to Processes Version 1[2].

- `-t,--ork-inactive-time=[90|<seconds>]`
  To prevent an *Orkhestra* connection circuit from going stale the circuit is reset after an inactive threshold. The default is 90 seconds.

- `--default-permissions=[1|<number>]`
  When a user logs in, the user permissions are set to the defined value for the user. If this value is zero, it will be set to this. The default value is 1.

- `-v, --verbose`
  When specified, *orksrvrr* operates in a verbose manner.

C Bibliography

References

[1] orkhestra: Configuration and User Reference Version 1. CML Document CML00041-01, Code Magus Limited, June 2011. PDF.

[2] **orkhestra_attproc**: Orkhestra: Attachments to Processes Version 1. CML Document CML00117-00, Code Magus Limited, November 2018. PDF.