Pushing Performance

People | Power | Partnership

HARTING EUROMAP 63 Gateway for MICA
User Manual

1. Edition 2018

Contents

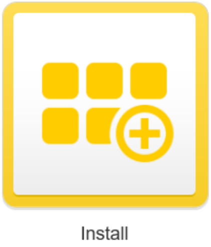
1 Installation

This installation guide refers to the MICA Base Version 2 and higher.

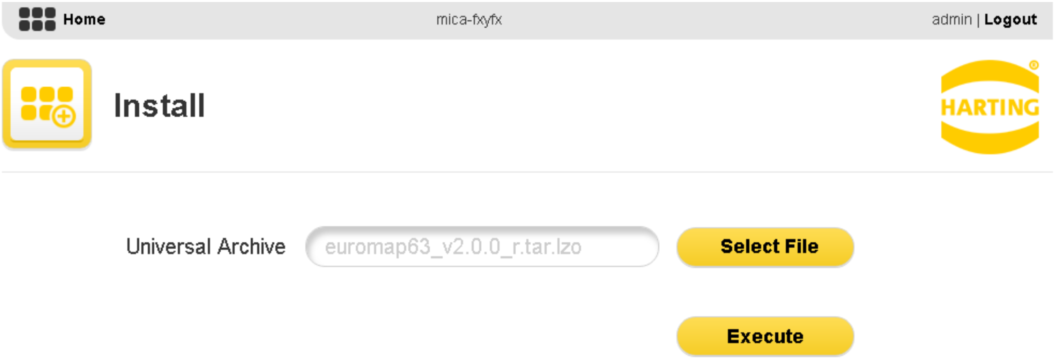
You may skip this chapter, if the “EUROMAP 63 Gateway” is already installed.

1.1 Installation of the EUROMAP 63 Gateway

Log in to the MICA with admin rights and click the “Install” icon.

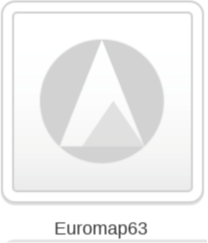


On the next view, select the installation archive of the “EUROMAP 63 Gateway”.
Next, click the “Execute” button. The container will be installed.



Universal Archive: euromap63_v2.0.0_r.tar.lzo

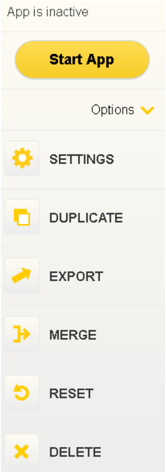
After the installation, the “EUROMAP 63 Gateway” Container icon appears.



As soon as the icon is visible you can start to configure the container (see chapter 1.1.1.1).

1.1.1 Basic Container functionality

Right clicking the “EUROMAP 63 Gateway” Container tile opens the container’s context menu. Press “Options” to expand the following functions:



Current state of the container.

Starts / Stops the container.

Expands / Hides the basic container functions.

Provides the container information and lets you configure the IPv4 / IPv6 settings, an Additional Network Interface and the Single Sign On Mode.

Duplicates the container on your MICA.

Exports the container to your PC. All configurations you set will be kept.

Overwrites the reset point of the Container with its current configuration.

Resets the configuration of the “EUROMAP 63 Gateway” to the last reset point (factory default if no merge was executed before).

Deletes the Container.

1.1.1.1 Network Settings

In the container’s context menu, click “SETTINGS” to configure the container’s IPv4 setting and confirm by clicking the “Activate Settings” buttons.
When done, go back to “Home” and click “Start App” in the containers context menu to start the container.
The “EUROMAP 63 Gateway” is now ready to be connected to a machine.

2 EUROMAP 63

2.1 About EUROMAP 63

EUROMAP 63 is a basic communication protocol which is necessary to establish a communication between a central computer and an injection molding machine via a file based data exchange interface. Here we use a MICA as the central computer to communicate with the machine.

2.2 The Communication Principle

The file based data exchange interface itself is accomplished using ASCII files. An underlying network file system is responsible for providing access to shared network file servers.

Communication sessions are initiated by placing a “Session Request File” in a particular machine’s “Session Directory”. The location of the “Session Directory” as well as the maximum number of communication sessions that can be activated at a time are specified in the “Machine Initialization File”.

Upon detection of a “Session Request File” from the MICA, the machine will open the file and begin to process the commands. As the commands are processed, response information is written to a “Session Response File”.

Once all “Session Request Commands” have been processed by the machine, the request file is deleted by the machine.

Upon detection of the deleted request file, the MICA can review the response file for command execution confirmation. When complete, the MICA deletes the response file.

Files termed “Job Files” are created to identify the information to be communicated with a particular machine. The machine will process this job file to both receive information as well as identify information to be written by the machine to a “Response File”.

For example, a job file for sending a complete set of machine setup parameters to a machine might have the following appearance (keywords in capital letters):

Command	Description
JOB test RESPONSE = test.log;	Specification of the data set.
UPLOAD \\sv1\vol1\data\test;	Use data from data set “test”.
START TIME >= 10:05:00;	Start upload at 10:05.

ℹ For further information refer to:

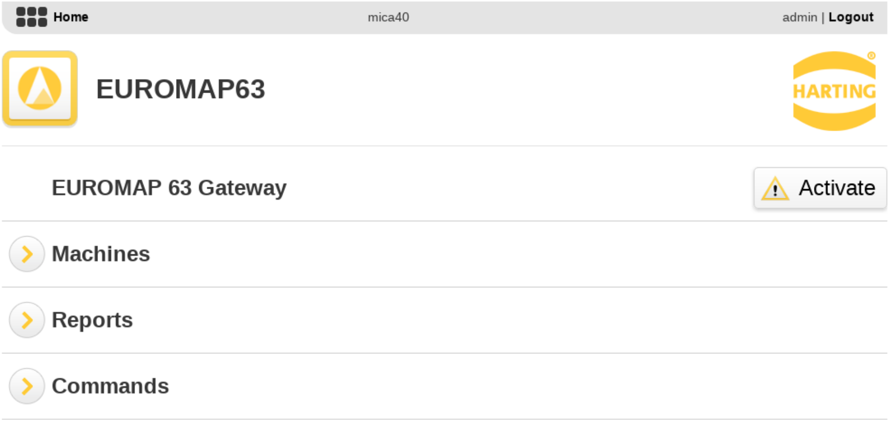
http://www.euromap.org

3 The EUROMAP 63 Gateway user interface

With a click on the “EUROMAP 63 Gateway” icon, the user interface of the gateway appears.

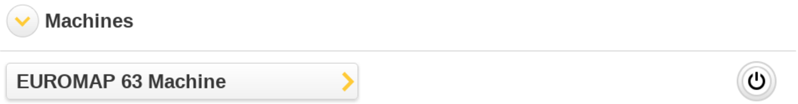
The exclamation mark below the HARTING logo indicates that the “EUROMAP 63 Gateway” Container is not yet licensed. Therefore it will run for a test period of 6h only, but can be restarted manually.

To obtain a license key, click on “Activate” and follow the steps in the popup box.

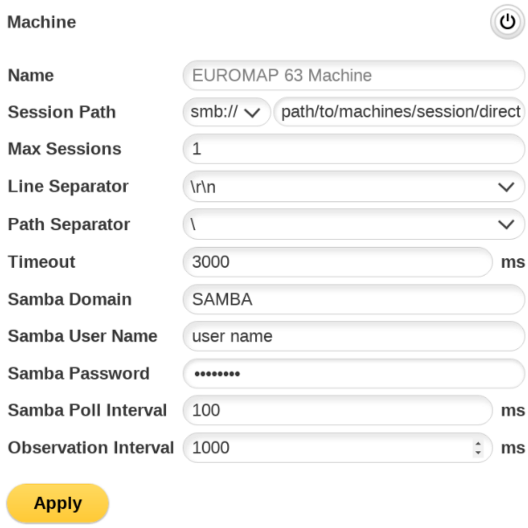


3.1 Machines

The section “Machines” is for connecting your machine to your MICA.



With a click on the applied “EUROMAP 63 Machine”, a configuration window appears.



Name	EUROMAP 63 Machine	Name of the machine.
Session Path	smb:// path/to/machines/session/direct	**Path to the machines “Session Directory”.**
Max Sessions	1	Maximum number of sessions.
Line Separator	\r\n	Line separator used in the machines response files.
Path Separator	\	Path separator used in the machine.
Timeout	3000 ms	Timeout in milliseconds.
Samba Domain	SAMBA	**Domain of the machines “Session Directory”.**
Samba User Name	user name	**User Name for the machines “Session Directory”.**
Samba Password	••••••••	**Password for the machines “Session Directory”.**
Samba Poll Interval	100 ms	Poll Interval for the “Session Directory” in milliseconds.
Observation Interval	1000 ms	Observation Interval in milliseconds.

The “Session Path” has to end with a “/”.

The “EUROMAP 63 Gateway” will create the following directory structure in the machines “Session Directory”:

- application
  - download
  - event
  - get_id
  - get_info
  - report
  - upload
- tmp

The result of a specific command will be saved temporally in the corresponding subdirectory of the application directory.

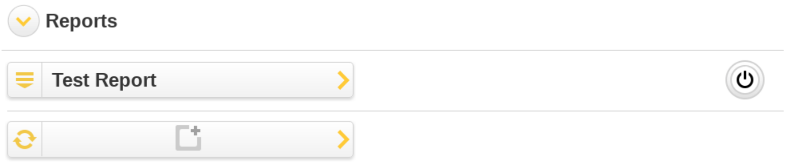
You can use the “Machine Info” command (see section 3.3.2) to request the maximum number of sessions your machine is able to handle.

Increase the “Timeout” when you got the following error message:

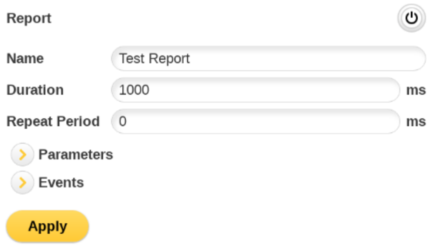
“Command processing timed out after <timeout>ms”

3.2 Reports

The section “Reports” is for publishing machine data on a “Subscriber” endpoint.



You can observe a set of variables (“Parameters”) as well as alarms or alarm changes (“Events”).


Report

Name: Test Report

Duration: 1000 ms

Repeat Period: 0 ms

Unique **Name** of the report specification.

Duration of the report in **ms**, i.e. how long the report should run.

Repeat Period of the report in **ms**, i.e. time interval after which the report should start again. The time interval starts after the previous report cycle ends.

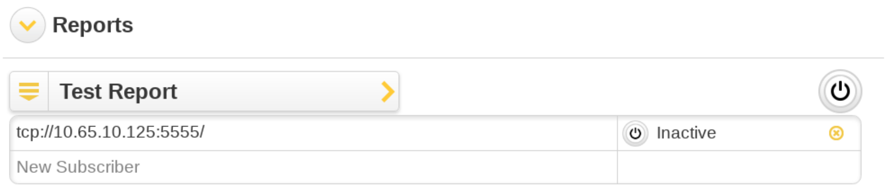
List of machine variables (**Parameters**) that should be reported.

List of **Events** that should be reported.

Add new parameters by clicking in the “New Parameter” field and type in the variable you want to observe. To get a list of all available variables you can use the “Variable Definition” command (see section 3.3.3).
Delete an existing variable by clicking on the ⊗.
Add particular output information to your report by clicking on the corresponding “Output” buttons.

Add alarms or alarm changes to your report by clicking the corresponding “Events” buttons. **Please note:** Individual reports will be published for the parameters and for each event.

For adding a new “Subscriber” endpoint, you have to click on the small icon ≡ on the left side of the report.

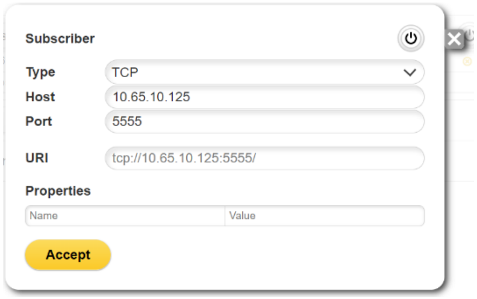


Reports

Test Report

tcp://10.65.10.125:5555/	⏻ Inactive
New Subscriber	

A menu will appear where you can add a new “Subscriber” endpoint to publish the gathered information from the machine. The supported types of endpoints are HTTP, HTTPS, MQTT, MQTTS, TCP and UDP. You can activate the endpoints with a click on ⏻ or delete an endpoint with the delete button.



Subscriber

Type: TCP

Host: 10.65.10.125

Port: 5555

URI: tcp://10.65.10.125:5555/

Properties

Name	Value

The result of an EUROMAP 63 `REPORT` job will not be deleted and will remain in the report directory (<sessionPath>/application/report) even when the report of the gateway will be deactivated.

If the gateway loose its connection to the machine unexpectedly while an EUROMAP 63 Report or Event Job is running on the machine, the job has to be aborted manually (see section 5.1).

3.2.1 Example Report

If you have configured your “Machine” and your “Reports”, you will get a report after the configured “Duration”.

A “`Report`” is a JSON structure containing the following fields:

Report
`applicationId : String;`
`reportName : String;`
`date : String;`
`totalMilliseconds : Integer;`
`devices : Device[]`

The fields of the “`Reports`” are specified in the following table:

Field	Type	Description
`applicationId`	`String`	ID of the application. In this case it is always “EUROMAP 63 Gateway”.
`reportName`	`String`	Name of the report.
`date`	`String`	Time stamp of the report.
`totalMilliseconds`	`Integer`	Duration of the report.
`devices`	`Device[]`	All monitored machines containing all captured fields.

Example report:

```
{
   "applicationId":"Euromap63Gateway",
   "reportName":"Test Report",
   "date":"2018-05-15T14:33:16.561",
   "totalMilliseconds":1000,
   "initiation":"REPEAT_PERIOD",
   "initiator":null,
   "termination":"DURATION",
   "terminator":null,
   "devices":[
      {
         "name":"EUROMAP 63 Machine",
         "fields":[
            {
               "name":"report:specId=4",
               "value":"@4410,SetTimMach,ActStsMach
                       75,20180516011440,\"0U001\"
                       ","date":"2018-05-15T14:33:15.633"
            }
         ]
      }
   ]
}
```

3.3 Commands

In the section “Commands” you can issue most EUROMAP 63 commands.

3.3.1 General

The button Request is for requesting information from the machine.

The button Transfer is for transferring information to the machine.

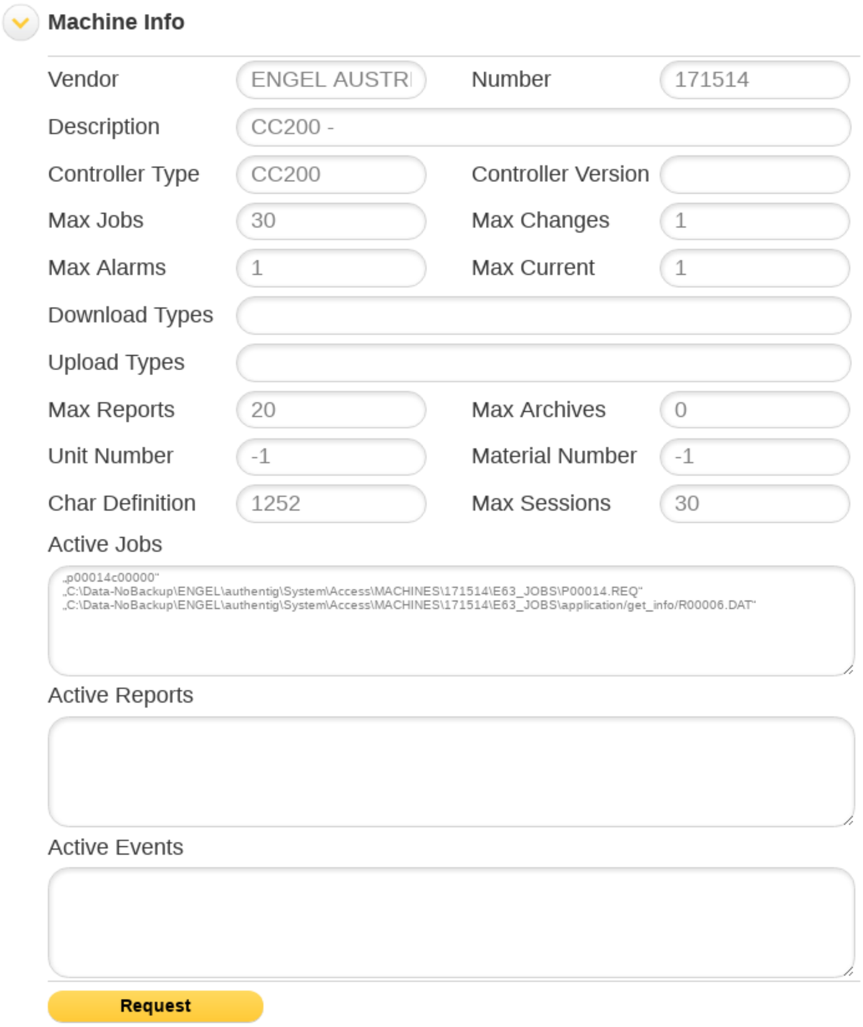
If you want to request or transfer information from/to the machine you have to click on the corresponding button. A loading icon will appear.

If the process was successful the loading icon disappears.

If the process was not successful the loading icon disappears and an error message appears at the bottom of your window. By clicking in your window the error message disappears.

3.3.2 Machine Info

In the subsection “Machine Info” you can get information about the hardware and the software from your machine.



Vendor	ENGEL AUSTRI	Number	171514
Description	CC200 -		
Controller Type	CC200	Controller Version	
Max Jobs	30	Max Changes	1
Max Alarms	1	Max Current	1
Download Types			
Upload Types			
Max Reports	20	Max Archives	0
Unit Number	-1	Material Number	-1
Char Definition	1252	Max Sessions	30

Active Jobs

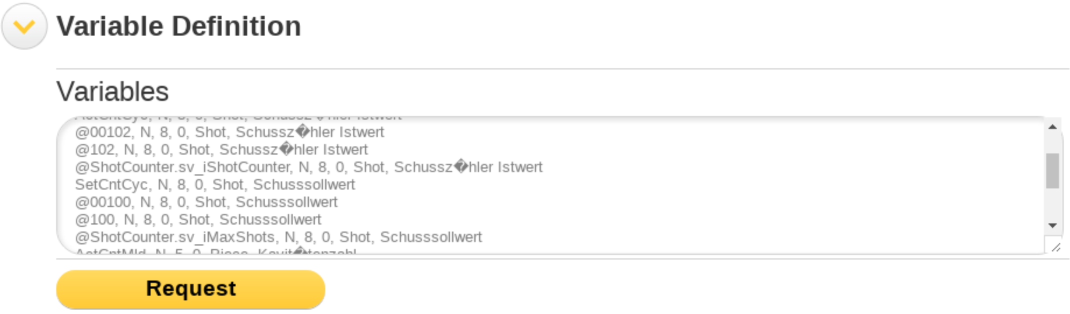
```
„p00014c00000“
„C:\Data-NoBackup\ENGEL\authentig\System\Access\MACHINES\171514\E63_JOBS\P00014.REQ“
„C:\Data-NoBackup\ENGEL\authentig\System\Access\MACHINES\171514\E63_JOBS\application/get_info/R00006.DAT“
```

Active Reports

Active Events

3.3.3 Variable Definition

In the subsection “Variable Definition” you can get all available variables from your machine.



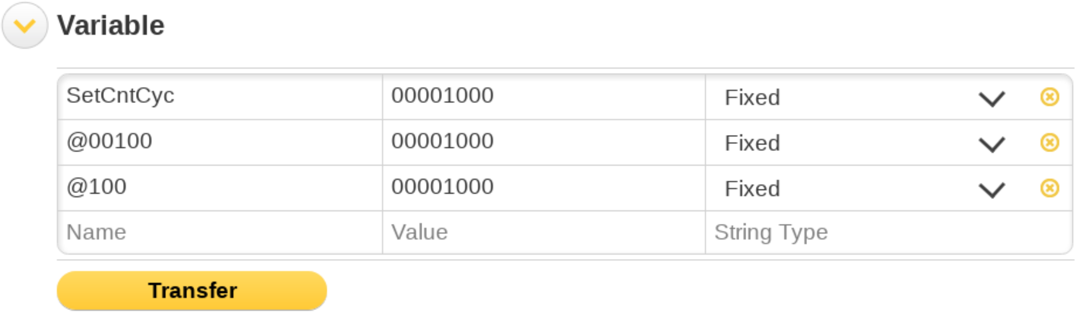
Variable Definition

Variables

```
ActCntCyc, N, 8, 0, Shot, Schussz�hler Istwert
@00102, N, 8, 0, Shot, Schussz�hler Istwert
@102, N, 8, 0, Shot, Schussz�hler Istwert
@ShotCounter.sv_iShotCounter, N, 8, 0, Shot, Schussz�hler Istwert
SetCntCyc, N, 8, 0, Shot, Schusssollwert
@00100, N, 8, 0, Shot, Schusssollwert
@100, N, 8, 0, Shot, Schusssollwert
@ShotCounter.sv_iMaxShots, N, 8, 0, Shot, Schusssollwert
ActCntMld, N, 5, 0, Piece, Kavit�tenzahl
```

3.3.4 Variable

In the subsection “Variable” you can set a variable to your machine.

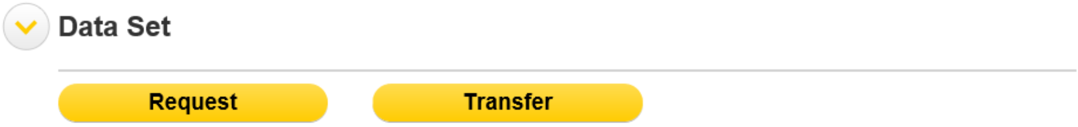


Variable

SetCntCyc	00001000	Fixed
@00100	00001000	Fixed
@100	00001000	Fixed
Name	Value	String Type

3.3.5 Data Set

In the subsection “Data Set” you can request and transfer from/to your machine the ***active*** setup data set as a zip file.



Data Set

⚠ You may get the following error message, when trying to transfer a data set:

```
Invalid DOWNLOAD command syntax. ( FSpec is not a valid
name, this char is not allowed (hex): 2f)
```

If so, please change the value for the ***Path Seperator*** in the ***Machines*** section to \.

4 The EUROMAP 63 Gateway REST interface

The “EUROMAP 63 Gateway” provides a REST interface which provides the same functionality as the user interface. The REST based http interface will look like shown in the following table while each resource begins with:

http://[Container-Host]/rest/EUROMAP/63

```
                                 <Interface>
                              EUROMAP 63 Gateway

Method     Resource                                     Parameter (JSON)
-------    ---------                                    ----------------
GET        ./devices                                    MachineConfiguration[]
GET/PUT    ./devices/{name}                             MachineConfiguration

GET        ./devices/{name}/fields/info                 MachineInfo
GET        ./devices/{name}/fields/variabledefinition   VariableDefinition
PUT        ./devices/{name}/fields/variables            Variable[]
GET/PUT    ./devices/{name}/fields/dataset              String //Octed String

GET        ./reports                                    ReportSpec[]
POST       ./reports                                    ReportSpec
GET/PUT    ./reports/{name}                             ReportSpec
DELETE     ./reports/{name}

GET        ./reports/{name}/subscribers                 Subscriber[]
POST       ./reports/{name}/subscribers                 Subscriber
GET/PUT    ./reports/{name}/subscribers/{subscriberID}  Subscriber
DELETE     ./reports/{name}/subscribers/{subscriberID}  Subscriber

                                <<Interface>>
                                   Callback
POST       http(s)://host:port/remainder-of-URL
           tcp://host:port | udp://host:port
           mqtt(s)://host:port/topic
```

4.1 MachineConfiguration

URL	Parameter	Description
`GET /devices`	`MachineConfiguration[]`	This resource can be used to request the list of defined machine configurations.
`GET/PUT /devices/{name}`	`MachineConfiguration`	This resource can be used to request or update a particular machine.

`MachineConfiguration` is a JSON structure containing the following fields:

```
MachineConfiguration

name : String;
enabled : Boolean;
sessionPath : String;
maxSessions : Integer;
lineSeparator : String; // \r\n (carriage return + line feed)
                        // \r (carriage return)
pathSeparator : String; // \
                        // /
connectTimeout : Integer;
smbDomain : String;
smbUsername : String;
smbPassword : String;
smbPollInterval : Integer;
cyclicPollInterval : Integer;
```

The fields of `MachineConfiguration` are specified in the following table:

Field	Type	Description
`name`	`String`	Unique name of this machine configuration.
`enabled`	`String`	Indicates whether this machine configuration is active or inactive.
`sessionPath`	`String`	Location of the directory to exchange files with the physical machine as a file URI.
`maxSessions`	`Integer`	Maximum number of communication sessions that can be activated at a time.
`lineSeparator`	`String`	Line separator which will be used in the exchanged files with the physical machine.
`pathSeparator`	`String`	Path separator which will be used in the exchanged files with the physical machine.
`connectTimeout`	`Integer`	Maximum timespan after which the machine should response in milliseconds.

smbDomain	String	(optional) Domain of a SMB file server.
smbUsername	String	(optional) User name of a SMB file server.
smbPassword	String	(optional) Password for the user name.
smbPollInterval	Integer	Poll interval for detecting file changes on the SMB file server.
cyclicPollInterval	Integer	Interval in which cyclic requests should be send to the machine in milliseconds.

Example:

```
{
   "name": "EUROMAP 63 Machine",
   "enabled": true,
   "sessionPath": "smb://10.65.54.100/imm_171514/",
   "maxSessions": 1,
   "lineSeparator": "\\r\\n",
   "pathSeparator": "\\",
   "connectTimeout": 3000,
   "smbDomain": null,
   "smbUsername": "admin",
   "smbPassword": "admin",
   "smbPollInterval": 100,
   "cyclicPollInterval": 1000
}
```

4.2 MachineInfo

URL	Parameter	Description
GET /devices/{name}/fields/info	MachineInfo	This resource can be used to request the information about the hardware and the software from the machine.

MachineInfo is a JSON structure containing the following fields:

MachineInfo
vendor : String;
number : String;
description : String;
controllerType : String;
controllerVersion : String;
softwareVersion : String;
maxJobs : Integer;
maxChangesEvents : Integer;
maxCurrentEvents : Integer;
maxAlarmEvents : Integer;

```
downloadTypes : String;
uploadTypes : String;
maxReports : Integer;
maxArchives : Integer;
injectionUnitNumber : Integer;
materialNumber : Integer;
characterDefinition : String;
maxSession : Integer;
activeJobs : Active[];
activeReports : Active[];
activeEvents : Active[];
```

The fields of `MachineInfo` are specified in the following table:

Field	Type	Description
`vendor`	`String`	Name of machine vendor.
`number`	`String`	Machine serial number.
`description`	`String`	Machine description.
`controllerType`	`String`	Controller Type.
`controllerVersion`	`String`	Controller version.
`softwareVersion`	`String`	Software version – version of EUROMAP-SPI definition.
`maxJobs`	`Integer`	Maximum number of jobs.
`maxChangesEvents`	`Integer`	Maximum number of active changes events.
`maxCurrentEvents`	`Integer`	Maximum number of active current alarms events.
`maxAlarmEvents`	`Integer`	Maximum number of active alarm events.
`downloadTypes`	`String`	List of manufactory specific keywords which specify a proprietary download.
`uploadTypes`	`String`	List of manufactory specific keywords which specify a proprietary upload.
`maxReports`	`Integer`	Maximum number of active reports.
`maxArchives`	`Integer`	Maximum number of archives.
`injectionUnitNumber`	`Integer`	Number of injection units.
`materialNumber`	`Integer`	Number of materials.
`characterDefinition`	`String`	Character definition, DOS-Codepage
`maxSession`	`Integer`	Maximum number of sessions supported by the machine.
`activeJobs`	`Active[]`	List of all active jobs.
`activeReports`	`Active[]`	List of all active reports.
`activeEvents`	`Active[]`	List of all active event logs.

Example:

```
{
    "vendor": "ENGEL AUSTRIA",
    "number": "171514",
    "description": "CC200 - ",
    "controllerType": "CC200",
    "controllerVersion": "",
    "softwareVersion": "1.05",
    "maxJobs": 30,
    "maxChangesEvents": 1,
    "maxCurrentEvents": 1,
    "maxAlarmEvents": 1,
    "downloadTypes": "",
    "uploadTypes": "",
    "maxReports": 20,
    "maxArchives": 0,
    "injectionUnitNumber": -1,
    "materialNumber": "",
    "characterDefinition": "1252",
    "maxSessions": 30,
    "activeJobs": [
        {
            "name": "p00045c00000",
            "jobFspec":
"C:\\Engel\\authentig\\System\\Access\\MACHINES\\171514\\E63_JOBS\\P00045.REQ",
            "responseFspec":
"C:\\Engel\\authentig\\System\\Access\\MACHINES\\171514\\E63_JOBS\\application\\ev
ent\\R00013.DAT",
            "type": null
        },
        {
            "name": "p00046c00000",
            "jobFspec":
"C:\\Engel\\authentig\\System\\Access\\MACHINES\\171514\\E63_JOBS\\P00046.REQ",
            "responseFspec":
"C:\\Engel\\authentig\\System\\Access\\MACHINES\\171514\\E63_JOBS\\application\\re
port\\R00014.DAT",
            "type": null
        },
        {
            "name": "p00047c00000",
            "jobFspec":
"C:\\Engel\\authentig\\System\\Access\\MACHINES\\171514\\E63_JOBS\\P00047.REQ",
            "responseFspec":
"C:\\Engel\\authentig\\System\\Access\\MACHINES\\171514\\E63_JOBS\\application\\ge
t_info\\R00004.DAT",
            "type": null
        }
    ],
    "activeReports": [
        {
            "name": "p00046c00001",
            "jobFspec":
"C:\\Engel\\authentig\\System\\Access\\MACHINES\\171514\\E63_JOBS\\P00046.REQ",
            "responseFspec":
"C:\\Engel\\authentig\\System\\Access\\MACHINES\\171514\\E63_JOBS\\application\\re
port\\R00014.DAT",
            "type": null
        }
    ],
    "activeEvents": [
        {
            "name": "p00045c00001",
            "jobFspec":
"C:\\Engel\\authentig\\System\\Access\\MACHINES\\171514\\E63_JOBS\\P00045.REQ",
            "responseFspec":
"C:\\Engel\\authentig\\System\\Access\\MACHINES\\171514\\E63_JOBS\\application\\ev
ent\\R00013.DAT",
            "type": null
        }
    ]
}
```

4.3 VariableDefinition

URL	Parameter	Description
GET /devices/{name}/fields /variabledefinition	VariableDefinition	This resource can be used to get all available variables from a particular machine.

VariableDefinition is a JSON structure containing the following fields:

VariableDefinition
name : String;
type : String;
integerDigits : String;
fractionalDigits : String
writable : Boolean;
unit : String
description : String;

The fields of VariableDefinition are specified in the following table:

Field	Type	Description
name	String	The name of the variable.

type	String	Type of the parameter, where A – alphanumeric value N – numeric value B – boolean value (0=false)
integerDigits	String	Number of integer digits.
fractionalDigits	String	Number of fractional digits.
writable	String	Indicates whether an update is allowed or not allowed.
unit	String	Unit of the variable.
description	String	Description of the variable.

Example:

```
{
    "name": "ActTimFill[1]",
    "type": "N",
    "integerDigits": "3",
    "fractionalDigits": "1",
    "writable": false,
    "unit": "Seconds",
    "description": "Actual Fill Time for injection unit 1"
}
```

4.4 Variable

URL	Parameter	Description
PUT /devices/{name}/fields/variables	Variable[]	With this resource you can set a list of variables.

`Variable` is a JSON structure containing the following fields:

Variable
name : String;
value : String;
vstring : Boolean;

The fields of `Variable` are specified in the following table:

Field	Type	Description
name	String	Unique name of the variable.
value	String	Value of the variable.
vstring	Boolean	Indicates if value is a variable length string.

Example:

```
[
    {
        "name": "SetDescMold",
        "value": "Mold 1314",
        "vstring": true
    },
    {
        "name": "SetTimMach",
        "value": 10150019971201,
        "vstring": false
    },
    {
        "name": "@SetMyPara_1",
        "value": 23.6,
        "vstring": false
    }
]
```

4.5 ReportSpec

URL	Parameter	Description
`GET` `/reports`	`ReportSpec[]`	This resource can be used to request the list of defined report specifications.
`POST` `/reports`	`ReportSpec`	Using the POST method, this resource can be used to add a new report specification.

`GET`
`/reports/{reportID}`
		This resource can be used to request a particular report specification.

`PUT`
`/reports/{reportID}`
	`ReportSpec`	Using the PUT method, this resource can be used to update a particular report specification.

`ReportSpec` is a JSON structure containing the following fields:

ReportSpec

`name : String;`
`deviceName : String;`
`enable : Boolean;`

```
duration: Long;
repeatPeriod: Long;
parameters : String[];
paramOutput : String[];      // TIME, DATE, COUNT
events : Event.Type[];       // ALARMS, CURRENT_ALARMS, CHANGES
```

The fields of `ReportSpec` are specified in the following table:

Field	Type	Description
`name`	`String`	Unique name of the report specification.
`deviceName`	`String`	Name of the machine used in this report specification.
`enable`	`Boolean`	Indicates whether the report specification is active or inactive.
`duration`	`Long`	Duration of the report in ms, i.e. how long the report should run.
`repeatPeriod`	`Long`	Repeat period of the report in ms, i.e. time interval after which the report should start again. The time interval starts after the previous report cycle ends.
`parameters`	`String[]`	List of machine variables that should be part of the report.
`paramOutput`	`String[]`	List of pseudo parameters that should be part of the report.
`events`	`Event.Type[]`	List of events that should be part of the report.

⚠ **Please note:** If the gateway loose its connection to the machine unexpectedly while an EUROMAP 63 Report or Event job is running on the machine, the job has to be aborted manually. Please refer to the appendix in chapter 5.

Example:

```
{
    "name": "My report",
    "deviceName": "EUROMAP 63 Machine",
    "enabled": true,
    "duration": 1000,
    "repeatPeriod": 0,
    "parameters": ["SetDescMold", "@SetMyPara_1"],
    "paramOutput": ["TIME", "DATE"],
    "events": ["CURRENT_ALARMS", "CHANGES"]
}
```

4.6 Subscriber

URL	Parameter	Description

`GET`
`/reports/{reportID}/subscribers`
	`Map<String, Subscriber[]>`	This resource can be used to request the list of defined subscribers for a particular report specification.

`POST`
`/reports/{reportID}/subscribers`
	`Subscriber`	Using the POST method, this resource can be used to add a new subscriber to a particular report specification.

`GET`
`/reports/{reportID}/subscribers/{subscriberID}`
		This resource can be used to request a particular subscriber of a particular report.

`PUT`
`/reports/{reportID}/subscribers/{subscriberID}`
	`Subscriber`	Using the PUT method, this resource can be used to update a particular subscriber of a particular report.

`DELETE`
`/reports/{reportID}/subscribers/{subscriberID}`
		Using the DELETE method, this resource can be used to delete a particular subscriber of a particular report.

`Subscriber` is a JSON structure containing the following fields:

```
Subscriber

id : String;
uri : String;
enable : Boolean;
properties : Map<String, String>
```

The fields of `Subscriber` are specified in the following table:

Field	Type	Description
`id`	`String`	ID of the subscriber.
`uri`	`String`	URI of the subscriber.
`enable`	`Boolean`	Indicates whether the subscriber is activated or not.
`properties`	`Map<String, String>`	Properties of the subscriber.

Example:

```
{
   "uri" : "tcp://127.0.0.1:5555/",
   "enable" : false,
   "properties" : { }
}
```

5 Appendix

5.1 Aborting operations manually

If the connection between the MICA and the machine is interrupted unexpectedly, active operations on the machine have to be aborted manually by following the described steps. For more detailed information please refer to the official EUROMAP 63 specification at http://www.euromap.org.

1. Active operations
    a. If the MICA is connected again, use the “Machine Info” command to see all active operations. (see section 3.3.2)
    b. Command file – “GetInfo.JOB”
        i. Create a file “GetInfo.JOB” and copy it to the “Session Directory”.

```
JOB GetInfo RESPONSE “GetInfo.LOG”;
GETINFO “GetInfo.dat”;
```

GetInfo.JOB

    c. Session Request file – “SESSnnnn.REQ”
        i. Create a Session Request file “SESSnnnn.REQ” where “nnnn” is an open Session Number and copy it to the Session Directory.

Example:

```
0000001 EXECUTE “GetInfo.JOB”;
```

SESS0001.REQ

2. Command file – “Abort.JOB”

   Create a file “Abort.JOB” and put it in the “Session Directory”. (see section 5.2)

3. Session Request file – “SESSnnnn.REQ”

   Create a Session Request file “SESSnnnn.REQ” where “nnnn” is an open Session Number and copy it to the Session Directory.

   Example:

```
0000001 EXECUTE “Abort.JOB”;
```

SESS0001.REQ

4. Delete all unused files in
    a. the Session Directory
    b. the Report Directory “<sessionPath>/application/report”
    c. the Event Directory “<sessionPath>/application/event”

5.2 EUROMAP 63 Abort Request Command

Use the following ABORT Commands in the Command file “Abort.JOB”.

5.2.1 ABORT ALL

This command aborts all active operations (excluding the current job executing this command).

```
JOB Abort RESPONSE “Abort.LOG”;
ABORT ALL;
```

5.2.2 ABORT ALL [JOBS | REPORTS | EVENTS]

These commands abort all active operations of a specific type.

Example:

```
JOB Abort RESPONSE "Abort.LOG";
ABORT ALL REPORTS;
```

5.2.3 ABORT [JOB | REPORT | EVENT] {name}

These commands abort a specific operation.

Example:

```
JOB Abort RESPONSE "Abort.LOG";
ABORT EVENT {name};
```

The `{name}` data follows the pattern `pXXXXXcXXXXX`. It must be the name specified in the JOB/REPORT/EVENT command of the operation to abort. You will find the command in the “Machine Info” (see section 5.1, Active operations) and the corresponding Presentation Request file `Pxxxxx.REQ` in the Session Directory.