Software interface

There are several software interfaces available to monitor the status of the RECS[®]|Box system. These are the Management WebGUI and a REST API providing XML based monitoring and management functionality. The Nagios NRPE interface was removed in RECS[®]|Box gen 4 systems.

Management WebGUI

The Management WebGUI is established on every RECS[®]|Box unit. Accessible by any known browser on the assigned IP address and the default port 80. The following views are dependent on the device and assembly.

In general these symbols have the following meaning on every page:

 \checkmark Everything is OK. Also indicated by a green line in a graph.

Warnung. Something is wrong, but the system is still fully functional. The system has to be checked so the problem doesn't get worse. Indicated by a yellow line in a graph.

Critical Error. The system must be checked immediately and maybe has to be shut down to prevent hardware damage. indicated by a red line in a graph.

Figure 1 shows the first call of the Management WebGUI. It is organized into three columns. The first is on the left-hand side and contains the following:

Overview: General overview of all managed RCUs, RPUs, installed nodes and health status Management: Selection of every managed RCU and RPU in the rack with a sensor view button for the Arneb

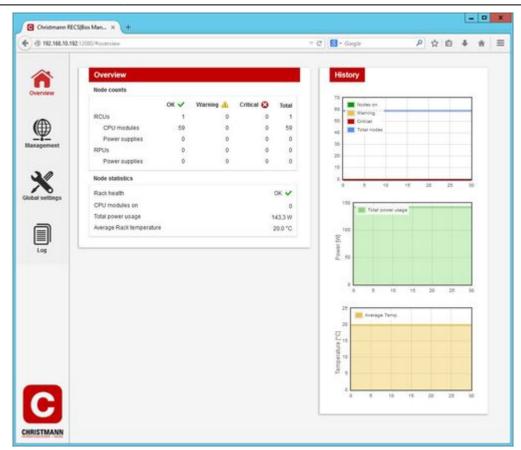
Global settings: IP filter and firmware update

Log: Logs from the management software about system health and java messages. The logs can be downloaded as a zipfile

The second colum contains the buttons and sliders to manipulate the system. While the third colum is mostly for history information like power usage and temperature graphs.

Overview

All units that are installed in the rack and that are managed by the software are summarized on this page. The total power usage is summed up over all managed units.





Management

An overview of the selected unit can be seen in this tab. The fans can be regulated by dragging the slider to the desired percentage. And multiple nodes can be selected. By klicking on a node the Node management page of the node is shown.

0 192,168,10	12010/Write	pet.jou-182.100.V01	£2		- C	- Google	Ŗ	☆ 自	+	Ĥ
	Cluster	RCU 192.16	8.10.192							
actient.	Ov	erview for RCU	192,168,10,192			History				
		-	1000	Bodes overview						
			designed to	Total		70	_		-	٦
A	1.15	Node 1-1 🗸	Node 18-4 🗸 🕈		59	80 Politer en	_	_		4
Ψ		Ande 5-2 🗸	Node 18-3 🗸	On	0	10 Diniar				
Inemage		Node 1-3 V	Node 15-2 V	OK V	59	45 Total todas				
Scores		Node 2-1 🗸	Node 17-4 🗸 -	Warning da	0	20 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1				1
		Node 2-3 🗸	Sede 17-3 🗸	Critical (3	0	20				1
10		Node 2-3 V Mode 2-4 V	Ande 17.2 V		· · · ·	*				1.
× 1		Node 3-1 V	Node 16-4 🗸	RCU status						۰.
•		Node 3-2 🗸	Node 16-3 🖌	HEO SIMUS			_	_	_	
secongs		Node 3-3 V	Node 16-2 V	Health status	OK V	6 8 9	e (#	20	M	50
		home 4-1	Node 15-4	Average node temperature	20.0 °C					
_	1 M	Node 4.2 V	Node 15-3	Highest node temperature	29.9 °C	1.00 Total p	and south			٦.
		hode #-3 V	Node 15-2							
		Node 5-1 V	Node 14-4	Total node power usage	0.00 W	2.75				1
og	1 1 1 1	Node 5-2 🗸	Node 14-3	Current KVM node	(hister)	2				
		Node 5-3 V	Node 14-2	Setsorview	Shaw	31.10				1.
		Node 5-1	Node 13-4	All and a second s		Lower M				
	1 H	Nore 6-2 🗸	Node 13-3	RCU settings		125				۰.
		Node 6-3 🗸	Node 15-2 🗸	Fan speed	37%					
		hiode 8-4 🗸	Node 13-1 🗸	r an apert	37.96	0.00			-	
	1.163	Aude 7.2 🗸	Node 12 😧	1		1. S.	18 18		15	30
		Node 7-3 🗸	× -	Manage multiple nodes		21				
		Note 8-1 V		Uut-selection		Avange	Terral.			
	1.163		Node 11 😧		mode	20	1	_		
		Nede 6-3 🗸	× -	Arried at Arried por	a. downt	7				
		Ande 5-4 🖌		04		5.0				
	1 18		Node 12 🚱			t t				
		Node 3-3 V Node 3-4 V	~	Philas private be		1.				
		-	A COLOR OF	Bet off.						
		1111 Z. N	Assemble	finish			1		-	
	-						10 10	28	28	90

Fig.

A quick menu to control a node can be opened by klicking on the gear next to an CXP node. In this menu the node can be switched on and off and the KVM can be switched to the node.

NUUE U-J		NUUC 13-2 V	
Node 6-4		Node 13-1	Fan speed:
Node 7-2	<u> </u>	On	-
Node 7-3	\geq	Press power butto	
Node 7-4	-	Press power butto	anage mul
Node 8-1		Hard off	
Node 8-3		Reset	Select all
Node 8-4	0	To KVM	
	-	Node 10 😧	

Fig.



Apalis nodes do not show a management pop-up button due to size constraints. Click on the node button while pressing the "Shift" key to open the management pop-up instead of navigating to the node view.

When pressing the "Shift" key while clicking, the "Select all" and "Select none" buttons select only nodes currently on or nodes currently off, respectively.

Node management

On this page the selected node can be controlled and detailed status values and graphs can be seen. By klicking on the arrow, pointing downwards in the upper bar next to the nodename, the other nodes of the unit can be chosen.

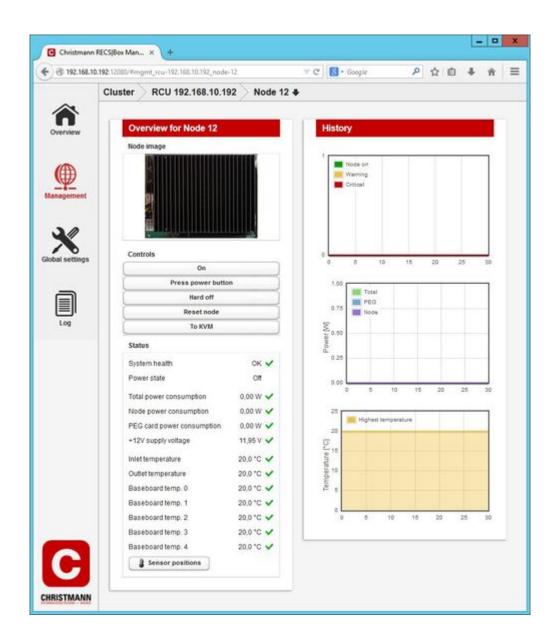


Fig. ##

Global settings

All IPs that are allowed to access the Nagios interface have to be listet here.

The firmware for the whole RECS[®]|Box can be uploaded here by klicking on the "Upload Firmware File" button and selecting the file. The update-process starts right after the file was uploaded. For the update process **all modules will be powered off!**





Log viewer

In the system healths tab of the log page the status changes of the sensors, fan and boards can be seen.

E 192.168.10.192	1200538944			- C 13 - 500		D	☆ €		*	-
Ca the two in the				(1			24			2
	Show logs								Č.	
Overview	System Health & Java Messag	es 🐨 🔂 Emor	Warning	🖌 🛈 Info	Debug	0	Trace	5		
<u>@</u>	MARE 2014-10-24 09:00:23,509 1 MARE 2014-10-24 09:00:23,509 1 INFO 2014-10-24 09:00:23,209 1 INFO 2014-10-24 09:00:24,209 1 INFO 2014-10-24 09:00:28,205 1	nf.chr.rec.tor.TCBDMaster nf.chr.rec.tor.TCBDMaster nf.chr.rec.tor.TCBDMaster nf.chr.rec.tor.TCBDMaster	[Thread-22] B [Thread-22] B [Thread-22] B [Thread-22] B	maith of Powers maith of Powers maith of Powers maith of Powers	upply 1 in RPC upply 2 in RPC upply 1 in RPC upply 2 in RPC	192.165 192.165 192.165 192.165	.10.192 .10.192 .10.193	Le Narr Le Narr Changet Changet		
×										
bal settings										
Log										
	ε.					0		þ		
	Download logs									
С										

Fig.

In the java tab of the log page all messages regarding the software can be found.

192.168.10.19	0.12080/#inst				- C Soogle	p	4	n.	4	*
					(a trate					
	Show logs									1
	System Health	g. Java Messapi	n 🗟 🔂 Emor	🗑 🦲 Warning	2 0 1Hb 0 0 Deb	ug: 🗆	Trac			
Log	IDFR 204-41-00 IRFE 2044-11-00 IRFE 2	0 = 13-13, 526 = 0 0 = 33-23, 571 = 1 0 = 32-23, 972 = 1 0 = 32-23, 972 = 1 0 = 33-23, 973 = 1 0 = 33-23, 974 = 1 0 = 33-23, 984 = 1 0 = 33-24, 984 = 1 0 = 33-24, 001 = 1 0 = 33	1.5. doi: rev. ACMeanse if doi: not rev.	The set of	<pre>r vallable hardware lable backglasse backglasse: 2 backglasse: 2 backglasse: 3 backglasse: 3 backglasse: 3 backglasse: 3 backglasse: 4 backglasse: 4</pre>	perticipe				
	e.								13	
	Download logs	1.0								



Several filters can be set for both tabs at the top. At the bottom the whole log can be downloaded as a ZIP file containing the individual logfiles.

Redfish API

The documentation of the RECS[®] |Box Redfish API can be seen at Github.

REST API

Access

The RECS[®]|Box Management API is accessible via the IP-Address or the hostname of the TOR-Master of the cluster. The basic URL of the API has the format https://TOR-Master/REST/ or http://TOR-Master/REST/.

Accessing the REST API requires HTTP Basic authentication. The authenticated user has to be in the "Admin" or "User" group to be able to execute the POST/PUT management calls.

Components

The RECS[®]|Box Management API makes all hardware components in the cluster available as XML trees in software. The following components are supported by the API:

Attribute	Description
node	A single node
backplane	A backplane can be equipped with zero or more baseboards
baseboard	A baseboard can be equipped with zero or more nodes
rcu	A RECS [®] Box Computing Unit (RCU) can be equipped with zero or more baseboards
rack	A rack consists of several RCUs

Many resources also return lists of components. These are named according to the scheme <component name>List (e.g. nodeList, rcuList) and contain the elements of the list.

Example of a backplaneList:

```
<backplaneList>
<backplane position="1" id="RCU_84055620466592_BP_1"
infrastructurePower="0.0">
<temperatures>24.0</temperatures>
<temperatures>25.0</temperatures>
<temperatures>26.0</temperatures>
<temperatures>27.0</temperatures>
<temperatures>28.0</temperatures>
</backplane>
```

</backplaneList>

Node

Example XML:

<node baseBoardPosition="0" maxPowerUsage="44"
actualNodePowerUsage="32.426884399865166"
actualPEGPowerUsage="15.12053962324833" actualPowerUsage="47.54742402311349"
architecture="x86"
baseBoardId="RCU_84055620466592_BB_1" health="0K"
id="RCU_84055620466592_BB_1_0" inletTemperature="20.0"
lastSensorUpdate="1465470151268" macAddressCompute="70:b3:d5:56:40:48"
outletTemperature="20.0" state="1"
highestTemperature="20.0" voltage="12.072700851453936"/>

The following table shows the possible attributes (some are optional) and their meaning:

Attribute	Description	Unit	Data type
id	Unique ID for referencing the component	-	String
actualPowerUsage	Actual power consumption of a node (Node + PEG)	W	Double
actualNodePowerUsage	Actual power consumption of a node (Node only)	W	Double
actualPEGPowerUsage	Actual power consumption of a PEG card	W	Double
maxPowerUsage	Maximum power the node can draw	W	Integer
baseBoardId	ID of the baseboard which hosts the node	-	String
baseBoardPosition	Position of the node on the baseboard	-	Integer
state	Power state of the node (0=Off, 1=On, 2=Soft-off, 3=Standby, 4=Hibernate)	-	Integer
architecture	Architecture (x86, arm, UNKNOWN)	-	String
health	Health status of the node (OK, Warning, Critical)	-	String
inletTemperature	Temperature of the inlet air	°C	Double
outletTemperature	Temperature of the outlet air	°C	Double
highestTemperature	Highest temperature measured on the node's baseboard	°C	Double
voltage	Supply voltage of the baseboard	V	Double
lastSensorUpdate	Timestamp of the last sensor update	ms	Long
macAddressCompute	MAC address of the NIC connected to the compute network (optional)	-	String
macAddressMgmt	MAC address of the NIC connected to the management network (optional)	-	String

In accordance to the component node the API offers nodeList which returns multiple instances of node.

Backplane

Example XML:

<pre><backplane <="" id="RCU_84055620466592_BP_1" position="1" pre=""></backplane></pre>
<pre>infrastructurePower="0.0"></pre>
<temperatures>24.0</temperatures>
<temperatures>25.0</temperatures>
<temperatures>26.0</temperatures>
<temperatures>27.0</temperatures>
<temperatures>28.0</temperatures>

The attributes have the following meaning:

Attribute	Description	Unit	Data type
id	Unique ID for referencing the component	-	String
postcion	Position of the backplane in the RECS [®] Box Computing Unit	-	Integer
infrastructurePower	Power usage of the infrastructure components on the backplane	w	Double
temperatures	List of temperatures measured on the backplane	°C	Double

In accordance to the component backplane the API offers backplaneList which returns multiple instances of backplane.

Baseboard

Example XML:

```
<br/><baseBoard rcuPosition="6" baseboardType="APLS" id="RCU_84055620466592_BB_6"<br/>infrastructurePower="9.8" rcuId="RCU_84055620466592"><br/><nodeId>RCU_84055620466592_BB_6_1</nodeId><br/><nodeId>RCU_84055620466592_BB_6_2</nodeId><br/><nodeId>RCU_84055620466592_BB_6_3</nodeId><br/><temperatures>20.0</temperatures><br/><temperatures>20.0</temperatures><br/><temperatures>20.0</temperatures><br/><temperatures>20.0</temperatures><br/><temperatures>20.0</temperatures><br/><temperatures>20.0</temperatures><br/><temperatures>20.0</temperatures><br/><temperatures>20.0</temperatures><br/><temperatures>20.0</temperatures><br/><temperatures>20.0</temperatures><br/><temperatures>20.0</temperatures><br/><temperatures>20.0</temperatures><br/><temperatures>20.0</temperatures><br/><temperatures>20.0</temperatures><br/><temperatures>20.0</temperatures><br/><temperatures>20.0</temperatures><br/><temperatures>20.0</temperatures><br/><temperatures>20.0</temperatures><br/><temperatures>20.0</temperatures><br/><temperatures>20.0</temperatures><br/><temperatures>20.0</temperatures><br/><temperatures>20.0</temperatures><br/><temperatures>20.0</temperatures><br/><temperatures>20.0</temperatures><br/><temperatures>20.0</temperatures><br/><temperatures>20.0</temperatures><br/><temperatures>20.0</temperatures><br/><temperatures>20.0</temperatures><br/><temperatures>20.0</temperatures><br/><temperatures>20.0</temperatures><br/><temperatures>20.0</temperatures><br/><temperatures>20.0</temperatures><br/><temperatures>20.0</temperatures><br/><temperatures>20.0</temperatures><br/><temperatures>20.0</temperatures><br/><temperatures>20.0</temperatures><br/><temperatures>20.0</temperatures><br/><temperatures>20.0</temperatures><br/><temperatures>20.0</temperatures><br/><temperatures>20.0</temperatures><br/><temperatures>20.0</temperatures>20.0</temperatures>20.0</temperatures>20.0</temperatures>20.0</temperatures>20.0</temperatures>20.0</temperatures>20.0</temperatures>20.0</temperatures>20.0</temperatures>20.0</temperatures>20.0</temperatures
```

The attributes have the following meaning:

Attribute	Description	Unit	Data type
id	Unique ID for referencing the component	-	String
rcuId	Unique ID of the RECS [®] Box Computing Unit hosting the baseboard	-	String
rcuPosition	Position of the baseboard inside the RECS [®] Box Computing Unit	-	Integer

Attribute	Description	Unit	Data type
infrastructurePower	Power usage of the infrastructure components on the baseboard	1	Double
baseboardType	Type of the baseboard (CXP, APLS)	-	String
nodeId	List of IDs of the nodes installed on the baseboard	-	String
temperatures	List of temperatures measured on the backplane	°C	Double

In accordance to the component baseboard the API offers baseboardList which returns multiple instances of baseboard.

RCU

Example XML:

```
<rcu rcuType="ANTARES" fanSpeed="60" rackId="RCK_1" name="RECSMaster (RCU)
on 192.168.56.195" rackPosition="0" id="RCU_84055620466592">
<backplaneId>RCU_84055620466592_BP_1</backplaneId>
<baseBoardId>RCU_84055620466592_BB_1</baseBoardId>
<baseBoardId>RCU_84055620466592_BB_2</baseBoardId>
<baseBoardId>RCU_84055620466592_BB_3</baseBoardId>
<baseBoardId>RCU_84055620466592_BB_3</baseBoardId>
<baseBoardId>RCU_84055620466592_BB_4</baseBoardId>
<baseBoardId>RCU_84055620466592_BB_5</baseBoardId>
<baseBoardId>RCU_84055620466592_BB_5</baseBoardId>
<baseBoardId>RCU_84055620466592_BB_6</baseBoardId>
</rcu>
```

The attributes have the following meaning:

Attribute	Description	Unit	Data type
id	Unique ID for referencing the component	-	String
rackId	ID of the rack which hosts the RECS [®] Box Computing Unit	-	String
rackPosition	Position of the RECS [®] Box Computing Unit in the rack	-	Integer
name	Name of the RECS [®] Box Computing Unit	-	String
ip	IP address of the RECS [®] Box Computing Unit	-	String
rcuType	Type of the RECS [®] Box Computing Unit (SIRIUS, ARNEB, ANTARES)	-	String
kvmNode	ID of the node to which the KVM system is switched (optional)	-	String
fanSpeed	Current speed setting of the fans in the RECS [®] Box Computing Unit	%	Integer
backplaneId	List of IDs of backplanes which are installed in the RECS [®] Box Computing Unit	-	String
baseBoardId	List of IDs of baseboards which are installed in the RECS [®] Box Computing Unit	-	String

In accordance to the component rcu the API offers rcuList which returns multiple instances of rcu.

Rack

Example XML:

```
<rack description="Default rack" id="RCK_1">
<rcuId>RCU_84055620466592</rcuId>
</rack>
```

The attributes have the following meaning:

Attribute	Description	Unit	Data type
id	Unique ID for referencing the component	-	String
description	Description of the rack	-	String
rcuId	List of IDs of RECS [®] Box Computing Units which are installed in the rack	-	String

In accordance to the component rack the API offers rackList which returns multiple instances of rack.

Resources

The resources are split into monitoring resources (for pure information gathering) and management resources (for changing the system configuration or state).

Monitoring

For monitoring the following resources are available:

Attribute	e Description	
/node	Returns a nodeList with all nodes of the cluster	GET
/node/{node_id}	Returns information about the node with the given ID	GET
/baseboard	Returns a baseboardList with all baseboards of the cluster	GET
/baseboard/{baseboard_id}	Returns information about the baseboard with the given ID	GET
/baseboard/{baseboard_id}/node	Returns a nodeList with all nodes that are installed on the baseboard with the given ID	GET
/backplane	Returns a backplaneList with all backplanes of the cluster	GET
/backplane/{backplane_id}	Returns information about the backplane with the given ID	GET
/rcu	Returns an rcuList with all RECS [®] Box Computing Units of the cluster	GET
/rcu/{rcu_id}	Returns information about the RECS [®] Box Computing Unit with the given ID	GET
/rcu/{rcu_id}/baseboard	Returns a baseboardList with all baseboards that are installed in the RECS [®] Box Computing Unit with the given ID	GET

ttribute Description		HTTP Method	
/rcu/{rcu_id}/backplane	Returns a backplaneList with all backplanes that are installed in the RECS [®] Box Computing Unit with the given ID	GET	
/rcu/{rcu_id}/node	Returns a nodeList with all nodes that are installed in the RECS [®] Box Computing Unit with the given ID	GET	
/rack	Returns a rackList with all racks of the cluster	GET	
/rack/{rack_id}	Returns information about the rack with the given ID	GET	
/rack/{rack_id}/rcu	Returns a rcuList with all RECS [®] Box Computing Units that are installed in the rack with the given ID	GET	

Management

The management of individual components can be found under the "manage" path of the component.

Attribute	Description	HTTP method	Parameter
/node/{node_id}/manage/power_on	Turns on the node with the given ID and returns updated node XML	POST	
/node/{node_id}/manage/power_off	Turns off the node with the given ID and returns updated node XML	POST	
/node/{node_id}/manage/reset	Resets the node with the given ID and returns updated node XML	POST	
/node/{node_id}/manage/select_kvm	Switches the KVM port of the RECS [®] Box Computing Unit containing the node to the node with the given ID and returns updated node XML	PUT	
/rcu/{rcu_id}/manage/set_fans	Sets the overall fan speed of the RCU with the given ID and returns the curent status of the RCU	PUT	percent={value}

Errors

Information about the success or failure of management requests are returned via HTTP status codes. Please have a look at RFC2616 for an overview about the defined HTTP status codes.

From:

https://recswiki.christmann.info/wiki/ - RECS®|Box Wiki

Permanent link: https://recswiki.christmann.info/wiki/doku.php?id=doc_recs4:software_interface&rev=1608117562

Last update: 2020/12/16 11:19

