OpenDAF

User's guide

Stanislav Ravas <ravas@tind.sk>

(C) 2013-2022 OpenDAF Contributors

Table of contents

1. OpenDAF User Guide 5
1.1 Introduction 5
1.2 Supported platforms 5
2. Core 6
2.1 Entities 6
2.2 Measurements & Commands 7
2.3 Alarms 10
2.4 Connectors & Providers 12
3. Connectors 13
3.1 DAFLink client 13
3.2 EAG3 14
3.3 EBO Breaker receiver 15
3.4 Modbus/RTU Master 16
3.5 File reader 18
3.6 GPIO Connnector 19
3.7 Linux hwmon connnector 20
3.8 Integrated Calculation Engine 21
3.9 IEC-60870-5-101 Controlling Station 22
3.10 IEC-60870-5-104 Controlling Station with passive connection establishment (accept) 24
3.11 IEC-60870-5-104 Controlling Station 26
3.12 IEC-62056-21 Tcp Master 28
3.13 Linux Industrial IO connnector 29
3.14 Kruiz stream receiver 30
3.15 Linux LED subsystem connnector 31
3.16 microELCOR and microELCOR-2 32
3.17 Modbus/RTU Master 33
3.18 Modbus/TCP Master 35
3.19 Monarch over streamd client 37
3.20 MONEZ client 38
3.21 MQTT client 39
3.22 MRE-10DO 10x digital output module 40
3.23 MRE-12AI 12x analog input module 41
3.24 MRE-16DI 16x digital input module 42
3.25 MRE-4AO 4x 0-10V analog output module 43
3.26 I&C O_GATE client 44

(C) 2013-2022 OpenDAF Contributors

3.27 OCPP 1.6 Central System 45
3.28 OGate Gateway client (using notification protocol) 46
3.29 OneWire Connnector 47
3.30 OPC-UA Client 48
3.31 OPC client 50
3.32 SEAK Panter Master 51
3.33 Persistent Storage Connector 53
3.34 PTNet controlling station 54
3.35 PTNet/Tinymesh controlling station 55
3.36 Linux PWM subsystem connnector 56
3.37 QERP 408/416 Power Factor Regulator 57
3.38 Romet ECS 58
3.39 Rotafon digitizer interface 59
3.40 Simatic S7 client 60
3.41 SCORPIO client 62
3.42 Access to OpenDAF internal state 63
3.43 SFLINT Luminance Photometer 64
3.44 Signal generator 65
3.45 SNMP manager 66
3.46 Communication object Per Row SQL client 67
3.47 EBO dosimetry SQL client 68
3.48 SuHub client (using notification protocol) 69
3.49 SWTN Controlling Station 70
3.50 SWTN Controlling Station 72
3.51 Teledosimetry client 73
3.52 TG800 Central Station 74
3.53 Tinymesh master 76
3.54 Teleperm-XS client (via GWD) 77
4. Providers 78
4.1 CHEMIS server 78
4.2 DAFLink provider 79
4.3 IEC-60870-5-101 Controlled Station Unballanced 80
4.4 IEC-60870-5-104 Controlled Station 82
4.5 Kepware CID provider 84
4.6 Kruiz active provider 86
4.7 Modbus/RTU Slave 87
4.8 Modbus/TCP Slave 89
4.9 OGate Gateway Notify server 91

(C) 2013-2022 OpenDAF Contributors

4.10 OPC UA server 92
4.11 SCORPIO server 93
4.12 Sigfox 94
4.13 Communication object Per Row SQL provider 95
4.14 EBO dosimetry SQL provider 96
4.15 SuHub Notify server 97
4.16 SuHub Periodic Notify server 98
5. Network integration APIs 99
5.1 RESTful API version 1 99
5.2 WebSocket API version 1 107
5.3 Transaction Ser/Des 110
5.4 DAFMan RESTful API - version 2 112
6. JavaScript API 130
7. Troubleshooting 131
7.1 `svc.conf` 131
7.2 `svc-recorder.conf` 131
7.3 `connectors.conf` 131
7.4 `connector-<name>.vars` 131
7.5 `providers.conf` 131
7.6 `provider-<name>.vars` 132
7.7 `measurements.csv`, `commands.csv`, `alarms.csv`, `alarm-groups.csv` 132
8. Re-measuring the 2026-10 perf optimizations on the target 133
8.1 What was changed, and which profile line each change targets 133
8.2 1. Before upgrading: confirm the diagnosis on the old build 133
8.3 2. Upgrade 134
8.4 3. After: what the same measurements should show 134
9. Measuring satellite (recorder/dispatcher) intake performance 135
9.1 What changed, what to expect 135
9.2 On a deployment 135
9.3 Loopback smoke (from the build tree, no install) 136

(C) 2013-2022 OpenDAF Contributors

1. OpenDAF User Guide

1.1 Introduction

OpenDAF is an acronym for *Open Data Acquisition Framework*. It is a framework upon which data-acquisition related applications can be built. It is built on a scalable modular architecture providing easily extensible functionality. It can operate on both embedded and server platforms.

The framework is composed of services and modules organized into stacks. Modules and services are configured into the process address space by a service configurator, leading to application architecture determined during deployment. The framework can be extended by creating new modules and services and by configuring them into the process. From the user point of view, framework is configured through its web interface, the WebDAF.

1.2 Supported platforms

Tested images are available for:

- Linux/x86
- Linux/x86-64
- Linux/ARM-v7a
- Linux/ARM-v8 (untested)
- Windows/x86
- Windows/x86-64
- many others possible, but not tested yet

(C) 2013-2022 OpenDAF Contributors

2. Core

2.1 Entities

OpenDAF configuration consists of several configurable named entities. Entity name must conform to C-language symbol name requirements (can contain only letters, digits and the underscore, can't start with digit) and must be unique among entities of the same type.

2.1.1 Entiry Attributes

Entity attributes fall into one of three categories:

- **mandatory** (mandatory user-configured attribute)
- **optional** (optional user-configured attribute)
- **runtime** (runtime-only attribute)

2.1.2 Measurement

By definition, *measurement* is the assignment of a number to a characteristic of an object or event, which can be compared with other objects or events. In the context of *OpenDAF*, *measurement* is always obtained from one of connected downlink devices via assigned *connector*.

More...

2.1.3 Command

In *OpenDAF* context, emitting command is a way to execute some action in connected downlink device via assigned *connector*. It can be used to control downlink device outputs, set setpoints, or invoke any device-specific actions.

More...

2.1.4 Alarm

Alarm is a stateful object representing some alarm condition presence. It can be triggered and acknowledged automatically by *script*, or through *public API*.

More...

2.1.5 Connector

Connector is an instance of a stack of software modules, designed to provide communication with some downlink device. The stack typically implements some standard (i.e. IEC-60870-5-101 controlling station, Modbus/RTU master...) or proprietary communication protocol.

More...

2.1.6 Provider

Provider is an instance of a stack of software modules, designed to handle communication with some uplink device. The stack typically implements some standard (i.e. IEC-60870-5-104 controlled station, Modbus/TCP slave...) or proprietary communication protocol.

More...

(C) 2013-2022 OpenDAF Contributors

2.2 Measurements & Commands

Measurements and *commands* are summarily called *Communication Objects* (*COs*).

2.2.1 Attributes

Overview:

Attribute	Datatype	Measurement	Command	Category
name	string	yes	yes	mandatory
description	string	yes	yes	optional
datatype	enum	yes	yes	mandatory
raw datatype	string	yes	yes	optional
connector name	string	yes	yes	optional
address	string	yes	yes	optional
value	bool, string, number	yes	yes	runtime
quality	number	yes	no	runtime
timestamp	timestamp	yes	yes	runtime
engineering unit	string	yes	yes	optional
engineering range	tuple	yes	yes	optional
raw range	tuple	yes	yes	optional
deadband	string	yes	no	optional
initial value	bool, string, number	no	yes	optional
archivation mode	enum	yes	yes	optional
archivation period	number	yes	yes	optional
archivation value deadband	number,string	yes	yes	optional
archivation time deadband	number	yes	yes	optional
provider addresses	map	yes	yes	optional

Name

[mandatory]

Uniquely identifies *measurement*/*command*. Must conform to C-language symbol name requirements (can contain only letters, digits and the underscore, can't start with digit) and must be unique among all *COs*.

Description

[optional]

Free text description of *CO* purpose.

Datatype

[mandatory]

(C) 2013-2022 OpenDAF Contributors

Datatype used to store measured value expressed in engineering units. Supported datatypes:

Datatype	Code	Storage size [B]	Range
binary	b	1	{0, 1}
quaternary	q	1	{0, 1, 2, 3}
integer	i	4	< -2^31, 2^31-1 >
long	l	8	< -2^63, 2^63-1 >
single-precision float	f	4	see *IEEE-754* standard
double-precision float	d	8	see *IEEE-754* standard
string	s		

Raw datatype

[optional]

Datatype used to obtain raw measured value from the *connector*.

Connector

[optional]

Name of *connector* to be used as a measurement datasource.

Address

[optional]

Address that represents this measurement in the context of *connector* assigned. I.e. Modbus reference, IEC-608750 Information Object Address, Persistor address etc. **Not to be confused with provider addresses!**

Current value

[runtime]

For *measurement*: Current measured value as assigned by its *connector* or other source. Always conforms to the configured datatype.
For *command*: Last value written to this command by some provider, REST API, or other source.

Current quality

[runtime]

Current measured value quality as assigned by its *connector* or other source, represented as a 16-bit unsigned integer value. Quality flags correspond to *OPC-DA 2.05* specification.

Current timestamp

[runtime]

For *measurement*: Current measured value *timestamp* as assigned by its *connector*. Resolution is 1ms. For *command*: *Timestamp* of last command write. Resolution is 1us.

Engineering unit

[optional]

(C) 2013-2022 OpenDAF Contributors

User-configurable engineering unit.

Engineering range

[optional]

Range of *CO* expressed in *engineering units*. Acquired raw samples would be scaled from *raw range* to this *engineering range*.

Raw range

[optional]

Range of *CO* in expressed in *physical units*.

Deadband

[optional]

Measured value *deadband* expressed either as a value in *engineering units* (i.e. 13.56), or a percent from the *engineering range* (i.e. 1.5%).

Initial value

[optional]

Value assigned to the command after startup. Defaults to *empty*.

Archivation mode

mode	description
none	*CO* is not archived
change	*CO* is archived on every value change
periodic	*CO* is archived with fixed archivation period

Archivation period

Used in conjunction with *periodic* archivation mode. Resolution 1ms.

Archivation value deadband

Measured value archive *deadband*. Expressed either as a value in *engineering units* (i.e. 13.56), or a percent from the *engineering range* (i.e. 1.5%). Applies in both *change* and *periodic* archivation modes.

Archivation time deadband

Measured value timestamp archive *deadband*. Applies in both *change* and *periodic* archivation modes.

Provider addresses

Provider address is an assignment of provider-specific *address* to the *communication object*. Most provider implementations provides only those *measurements* having *provider address* for that *provider* assigned.

(C) 2013-2022 OpenDAF Contributors

2.3 Alarms

2.3.1 Attributes

Overview:

Attribute	Datatype	Category
name	string	mandatory
description	string	optional
state	enum	runtime
timestamp	timestamp	runtime
severity	number	mandatory
authority	string	runtime
acknowledge mode	enum	mandatory
archivation mode	enum	mandatory

name

[mandatory]

Uniquely identifies the *alarm*. Must conform to C-language symbol name requirements (can contain only letters, digits and the underscore, can't start with digit) and must be unique among all *alarms*.

Description

[optional]

Free text description of *alarm* purpose.

state

[runtime]

Current alarm state. Possible values:

State	Active	Acknowledged	Description
INACT_ACK	no	yes	Initial state
ACT_UNACK	yes	no	Right after activation
ACT_ACK	yes	yes	Active alarm was acknowledged
INACT_UNACK	no	yes	Alarm deactivated, but not acked yet

timestamp

[runtime]

Timestamp of last alarm state update.

severity

[mandatory]

(C) 2013-2022 OpenDAF Contributors

Positive number describing alarm severity value. Exact values are application specific and freely assignable by system designer. Lower number = more severe alarm.

authority

[runtime]

Identity of latest process or user manipulating alarm state.

acknowledgement mode

[mandatory]

Valid values:

Mode	Description
manual	after deactivation alarm remains in `INACT_UNACK` state until acknowledged
auto	after deactivation alarm goes to state `INACT_ACK` automatically

archivation mode

[mandatory]

mode	description
none	*CO* is not archived
change	*CO* is archived on every value change
periodic	*CO* is archived with fixed archivation period

(C) 2013-2022 OpenDAF Contributors

2.4 Connectors & Providers

Connectors and *Providers* are summarily called *Stack Instantiations* (*SIs*).

2.4.1 Attributes

Overview:

Attribute	Datatype	Connector	Provider	Category
name	string	yes	yes	mandatory
stack name	string	yes	yes	mandatory
configuration variables	map	yes	yes	mandatory

(C) 2013-2022 OpenDAF Contributors

3. Connectors

3.1 DAFLink client

3.1.1 Parameters

Parameter	Datatype	Default Value	Description
DEBUG	enum	0	Enable debugging
DUMP	enum	0	Enable communication dump
PEER_ADDRESSES	list_of_inet_addr		Destination TCP/IP hostname:port, separate multiple entries with comma
TIMEOUT	float	20	Communication timeout [s]
INACTIVITY_TIMEOUT	float	60	Data change inactivity timeout [s] (0 = disabled)
RECONNECT_DELAY	float	10	Reconnection delay after connection lost
RX_MSG_SIZE_LIMIT	integer	1048576	Received message size limit [B]
TX_QUEUE_SIZE_LIMIT	integer	1048576	Transmit message queue size limit [B]

3.1.2 Measurements

This stack supports measurements.

Measurement address

Type: regex

Description:

```
namespace::symbol
Only 'measurements' namespace is available yet
```

3.1.3 Commands

This stack supports commands.

Command address

Type: regex

Description:

```
namespace::symbol
Only 'commands' namespace is available yet
```


(C) 2013-2022 OpenDAF Contributors

3.2 EAG3

3.2.1 Parameters

Parameter	Datatype	Default Value	Description
DEBUG	enum	0	Enable debugging
DUMP	enum	0	Enable communication dump
TTY	tty		Communication interface
SCAN_PERIOD	float	10.0	Scan period [s]
RESPONSE_TIMEOUT	float	0.5	Response timeout [s]
N_SCAN_RETRIES	int	2	Number of scan retries

3.2.2 Measurements

This stack supports measurements.

Measurement address

Type: enum

3.2.3 Commands

This stack supports commands.

Command address

Type: enum

(C) 2013-2022 OpenDAF Contributors

3.3 EBO Breaker receiver

3.3.1 Parameters

Parameter	Datatype	Default Value	Description
DEBUG	enum	0	Enable debugging
DUMP	enum	0	Enable communication dump
RX_ADDRESS	inet_addr		UDP/IP receive address
RX_PORT	integer	5600	UDP/IP receive port
RX_TIMEOUT	float	1	Receive timeout [s]

3.3.2 Measurements

This stack supports measurements.

Measurement address

Type: regex

3.3.3 Commands

Commands are not supported on this stack

(C) 2013-2022 OpenDAF Contributors

3.4 Modbus/RTU Master

3.4.1 Parameters

Parameter	Datatype	Default Value	Description
DEBUG	enum	0	Enable debugging
DUMP	enum	0	Enable communication dump
TTY	tty		Communication interface
BAUDRATE	enum	115200	Baud rate
PARITY	enum	even	Parity
INTERFRAME_DELAY	float	0.0	Interframe delay (0 = 3.5x character-time) [s]
RESPONSE_TIMEOUT	float	0.5	Receive timeout [s]
T_AWAKE	float	5.0	How long slave devices stays awake [s]
T_SETTLE	float	0.1	How long it takes slave device to settle after waking up [s]
WAKE_PACKET_LENGTH	int	10	Wake packet length [B]
N_SCAN_RETRIES	int	2	Number of scan retries

3.4.2 Measurements

This stack supports measurements.

Measurement address

Type: regex

Description:

```
<unit-address>/[datatype@]<address-space>:<reference>, where:
    - unit-address is a modbus slave address
    - datatype is used to define object layout if it spans multiple registers
        - format: <S|U|F|D><L|B><1|2|4>, where:
            - S - signed
            - U - unsigned
            - F - floating point
            - D - discrete
            - L - little-endian (LOW before HIGH)
            - B - big endian (HIGH before LOW)
            - 1 - size is 1 register/coil/discrete input
            - 2 - size is 2 registers/coils/discrete inputs
            - 4 - size is 4 registers/coils/discrete inputs
        - Unsupported datatypes: FL1, FB1, DL4, DB4
    - address-space is modbus address space number (0 - coils, 1 - discrete inputs, 3 - input registers, 4 - holding registers)
    - reference is 1-based register, coil or discrete input number in given address space
```

Examples:

```
3/3:1000 - input register 1000 on slave 3 with default datatype (16-bit unsigned)
7/1:550 - discrete input 550 on slave 7
10/fb4@4:1310 - holding registers 1310,1311,1312,1313 on slave 10 mapped as 64-bit big-endian floating-point value
11/dl2@0:35 - coils 35,36 on slave 11 mapped as 2-bit (quaternary) in little-endian (LSB, MSB) order
```

3.4.3 Commands

This stack supports commands.

(C) 2013-2022 OpenDAF Contributors

Command address

Type: regex

Description:

```
<unit-address>/[datatype@]<address-space>:<reference>, where:
    - unit-address is a modbus slave address
    - datatype is used to define object layout if it spans multiple registers
        - format: <S|U|F|D><L|B><1|2|4>, where:
            - S - signed
            - U - unsigned
            - F - floating point
            - D - discrete
            - L - little-endian (LOW before HIGH)
            - B - big endian (HIGH before LOW)
            - 1 - size is 1 register/coil/discrete input
            - 2 - size is 2 registers/coils/discrete inputs
            - 4 - size is 4 registers/coils/discrete inputs
        - Unsupported datatypes: FL1, FB1, DL4, DB4
    - address-space is modbus address space number (0 - coils, 4 - holding registers)
    - reference is 1-based register, coil or discrete input number in given address space
```

Examples:

```
3/4:1000 - holding register 1000 on slave 3 with default datatype (16-bit unsigned)
7/0:550 - coil 550 on slave 7
10/fb4@4:1310 - holding registers 1310,1311,1312,1313 on slave 10 mapped as 64-bit big-endian floating-point value
11/dl2@0:35 - coils 35,36 on slave 11 mapped as 2-bit (quaternary) in little-endian (LSB, MSB) order
```


(C) 2013-2022 OpenDAF Contributors

3.5 File reader

3.5.1 Parameters

Parameter	Datatype	Default Value	Description
DEBUG	enum	0	Enable debugging
LOCAL_TIMESTAMP	enum	0	Override timestamp with local timestamp
TIMEZONE	string	UTC	Timezone of data in the file according to POSIX standard (keep empty to use system timezone)
SCAN_PERIOD	float	1	File scan period
FILE_PATH	string		Path to file
DIALECT	enum	SHCC	Syntax dialect

3.5.2 Measurements

This stack supports measurements.

Measurement address

Type: string

Description:

```
Syntax depends on concrete dialect
SHCC: <SENSOR>.<ACC|VEL|DISP>
```

Examples:

```
3CA1X2.ACC
3CA1X2.VEL
3CA1X2.DISP
```

3.5.3 Commands

Commands are not supported on this stack

(C) 2013-2022 OpenDAF Contributors

3.6 GPIO Connnector

3.6.1 Parameters

Parameter	Datatype	Default Value	Description
DEBUG	enum	0	Enable debugging

3.6.2 Measurements

This stack supports measurements.

Measurement address

Type: regex

Description:

```
[<di|do|counter>@][!|!!]<pin-id>[/d:<t-debounce>]
di - measurement is a digital input (default)
do - measurement is a digital output readback
counter - measurement is a counter, integer/long datatype required
! - negated input (i.e. NC)
!! - non-negated input (i.e. NO) (default)
<pin-id> - can be one of:
  - symbolic name with dollar prefix (i.e.$GPIO_DI0)
  - hardware-dependent pin number (i.e.215)
<t-debounce> - debounce time in milliseconds
```

Examples:

```
215 - digital input on pin 215, normally-opened
!215 - digital input on pin 215, normally-closed
$GPIO_DI0 - digital input on symbolic pin GPIO_DI0
di@!$GPIO_DI0 - digital input on symbolic pin GPIO_DI0, normally closed
do@$GPIO_DI0 - digital output readback on symbolic pin GPIO_DI0
$GPIO_DI0/d:100 - digital input on symbolic pin GPIO_DI0, with 100ms debouncer
counter@!$GPIO_DI1 - counter on symbolic pin GPIO_DI1, normally closed
```

3.6.3 Commands

This stack supports commands.

Command address

Type: regex

Description:

```
[<do|counter-reset>@][!|!!]<pin-id>
do - command is a digital output (default)
counter-reset - command is reset for counter measurement on <pin-id>
! - negated input (i.e. NC)
!! - non-negated input (i.e. NO) (default)
<pin-id> - can be one of:
  - symbolic name with dollar prefix (i.e.$GPIO_DI0)
  - hardware-dependent pin number (i.e.215)
```

Examples:

```
96 - digital output on pin 96, nomally-opened
!96 - digital output on pin 96, normally-closed
$GPIO_DO0 - digital output on symbolic pin GPIO_DI0
do@!$GPIO_DO0 - digital output on symbolic pin GPIO_DI0, normally closed
counter-reset@$GPIO_DO1 - reset counter for symbolic pin GPIO_DI1
```


(C) 2013-2022 OpenDAF Contributors

3.7 Linux hwmon connnector

3.7.1 Parameters

Parameter	Datatype	Default Value	Description
DEBUG	enum	0	Enable debugging
SCAN_PERIOD	float	1	Scan period [s]

3.7.2 Measurements

This stack supports measurements.

Measurement address

Type: regex

Description:

```
<chip-pattern>/<feature-name>.<subfeature-name>
```

3.7.3 Commands

This stack supports commands.

Command address

Type: regex

Description:

```
<chip-pattern>/<feature-name>.<subfeature-name>
```


(C) 2013-2022 OpenDAF Contributors

3.8 Integrated Calculation Engine

3.8.1 Parameters

Parameter	Datatype	Default Value	Description
DEBUG	enum	0	Log debug messages
DEBUG_JS	enum	0	Enable Javascript debugging
DEBUG_ADDRESS	regex	0.0.0.0:9091	: to listen for incoming debug connection
INIT_FILE	string		Initialization script file path. Environment variables like $OPENDAF_PREFIX can be used. (leave empty to use no init file)

3.8.2 Measurements

This stack supports measurements.

Measurement address

Type: javascript

Description:

```
Javascript code shall register either periodic or change-sensing callback function
```

Examples:

```
calc.each(1, function() { print(calc.name); return ice.select([ice.mv('M_13CD0001'), ice.mv('M_23CD0001')]); })
```

3.8.3 Commands

This stack supports commands.

Command address

Type: javascript

Description:

```
Javascript function taking 3 arguments (name, v, t)
```

Examples:

```
function(name,v,t) { print('Value ' + v + ' timestamp ' + t + ' written to command ' + name); }
```


(C) 2013-2022 OpenDAF Contributors

3.9 IEC-60870-5-101 Controlling Station

3.9.1 Parameters

Parameter	Datatype	Default Value	Description
DEBUG	enum	0	Enable debugging
DUMP	enum	0	Enable communication dump
TTY	tty		Communication interface
BAUDRATE	enum	115200	Baud rate (must be equal for all stations on the same bus)
PARITY	enum	even	Parity (must be equal for all stations on the same bus)
N_RETRIES	int	2	Maximum number of transaction retries (must be equal for all stations on the same bus)
RX_TIMEOUT	float	0.8	Receive timeout [s] (must be equal for all stations on the same bus)
TX_BUFFER	int	16384	Transmit buffer size
OCTETS_LA	int	1	# of octets of Link Address (must be equal for all stations on the same bus)
OCTETS_CA	int	2	# of octets of Common Address of ASDU
OCTETS_IOA	int	3	# of octets of Information Object Address
OCTETS_COT	int	2	# of octets of Cause Of Transmission
OCTET_ORDER	enum	direct	Order of CA/IOA octets
STATION_ADDRESS	int		Outstation address
POLL_PERIOD	float	1.0	Outstation link-layer poll period [s] (must be equal for all stations on the same bus)
INT_AS_BITSTRING	enum	0	Use TI 51/64 (32-bit bitstring) for integer setpoints
LONG_AS_BITSTRING	enum	0	Use TI 51/64 (32-bit bitstring) for long setpoints
T_EI	float	10	End-Of-Initialization timeout [s] (0 = disabled)
T_CLOCKSYNC	float	3600	Clock synchronization period [s] (0 = disabled)
T_GI	float	60	General interrogation period [s] (0 = disabled)
T_GI_CON	float	10	General interrogation confirmation timeout [s]
T_GI_TERM	float	30	General interrogation termination timeout [s]
UTC_TIMESTAMPS	enum	1	Transmit/receive timestamps in UTC timezone

3.9.2 Measurements

This stack supports measurements.

Measurement address

Type: regex

(C) 2013-2022 OpenDAF Contributors

Description:

```
<CA0>[.<CA1>]/<IOA0>[.<IOA1>[.<IOA2>]]
```

Examples:

```
10.5/1.2.3 - Common Address 10.5, Information Object Address 1.2.3
3/4.5 - Common Address 3, Information Object Address 3.4
16.8/30 - Common Address 16.8, Information Object Address 30
```

3.9.3 Commands

This stack supports commands.

Command address

Type: regex

Description:

```
[TI<TI>@]<CA0>[.<CA1>]/<IOA0>[.<IOA1>[.<IOA2>]]
```

Examples:

```
10.5/1.2.3 - Write to Common Address 10.5, Information Object Address 1.2.3 using default ASDU TI
TI45@10.5/1.2.3 - Write to Common Address 10.5, Information Object Address 1.2.3 using ASDU with TI45 - Single Command
TI48@3/4.5 - Write to Common Address 3, Information Object Address 3.4 using ASDU with TI48 - Set-point command, normalized value
16.8/30 - Write Common Address 16.8, Information Object Address 30 using default ASDU TI
```


(C) 2013-2022 OpenDAF Contributors

3.10 IEC-60870-5-104 Controlling Station with passive connection establishment (accept)

3.10.1 Parameters

Parameter	Datatype	Default Value	Description
DEBUG	enum	0	Enable debugging
DUMP	enum	0	Enable communication dump
HOST	hostname	0.0.0.0	Listen hostname or address (0.0.0.0 = any)
PORT	integer	2404	Listen port
W	int	20	Immediatelly acknowledge frames received after receiving this number of frames (parameter W)
K	int	30	Block frame transmission after reaching this number of unacknowledged frames (parameter K)
TIMEOUT	float	20	Communication timeout [s]
OCTETS_CA	int	2	# of octets of Common Address of ASDU
OCTETS_IOA	int	3	# of octets of Information Object Address
OCTETS_COT	int	2	# of octets of Cause Of Transmission
OCTET_ORDER	enum	direct	Order of CA/IOA octets
TIMESTAMP_COMMANDS	enum	0	Emit commands & setpoints using ASDU types with timestamps
INT_AS_BITSTRING	enum	0	Use TI 51/64 (32-bit bitstring) for integer setpoints
LONG_AS_BITSTRING	enum	0	Use TI 51/64 (32-bit bitstring) for long setpoints
T_EI	float	10	End-Of-Initialization timeout [s] (0 = disabled)
T_GI	float	60	General interrogation period [s] (0 = disabled)
T_GI_CON	float	10	General interrogation confirmation timeout [s]
T_GI_TERM	float	30	General interrogation termination timeout [s]
UTC_TIMESTAMPS	enum	1	Transmit/receive timestamps in UTC timezone

3.10.2 Measurements

This stack supports measurements.

Measurement address

Type: regex

Description:

```
<CA0>[.<CA1>]/<IOA0>[.<IOA1>[.<IOA2>]]
```

Examples:

```
10.5/1.2.3 - Common Address 10.5, Information Object Address 1.2.3
3/4.5 - Common Address 3, Information Object Address 3.4
16.8/30 - Common Address 16.8, Information Object Address 30
```


(C) 2013-2022 OpenDAF Contributors

3.10.3 Commands

This stack supports commands.

Command address

Type: regex

Description:

```
[TI<TI>@]<CA0>[.<CA1>]/<IOA0>[.<IOA1>[.<IOA2>]]
```

Examples:

```
10.5/1.2.3 - Write to Common Address 10.5, Information Object Address 1.2.3 using default ASDU TI
TI45@10.5/1.2.3 - Write to Common Address 10.5, Information Object Address 1.2.3 using ASDU with TI45 - Single Command
TI48@3/4.5 - Write to Common Address 3, Information Object Address 3.4 using ASDU with TI48 - Set-point command, normalized value
16.8/30 - Write Common Address 16.8, Information Object Address 30 using default ASDU TI
```


(C) 2013-2022 OpenDAF Contributors

3.11 IEC-60870-5-104 Controlling Station

3.11.1 Parameters

Parameter	Datatype	Default Value	Description
DEBUG	enum	0	Enable debugging
DUMP	enum	0	Enable communication dump
PEER_ADDRESSES	list_of_inet_addr		Destination TCP/IP hostname:port, separate multiple entries with comma
W	int	20	Immediatelly acknowledge frames received after receiving this number of frames (parameter W)
K	int	30	Block frame transmission after reaching this number of unacknowledged frames (parameter K)
TIMEOUT	float	20	Communication timeout [s]
OCTETS_CA	int	2	# of octets of Common Address of ASDU
OCTETS_IOA	int	3	# of octets of Information Object Address
OCTETS_COT	int	2	# of octets of Cause Of Transmission
OCTET_ORDER	enum	direct	Order of CA/IOA octets
RECONNECT_DELAY	float	10	Reconnection delay after connection lost
TIMESTAMP_COMMANDS	enum	0	Emit commands & setpoints using ASDU types with timestamps
INT_AS_BITSTRING	enum	0	Use TI 51/64 (32-bit bitstring) for integer setpoints
LONG_AS_BITSTRING	enum	0	Use TI 51/64 (32-bit bitstring) for long setpoints
T_EI	float	10	End-Of-Initialization timeout [s] (0 = disabled)
T_GI	float	60	General interrogation period [s] (0 = disabled)
T_GI_CON	float	10	General interrogation confirmation timeout [s]
T_GI_TERM	float	30	General interrogation termination timeout [s]
UTC_TIMESTAMPS	enum	1	Transmit/receive timestamps in UTC timezone

3.11.2 Measurements

This stack supports measurements.

Measurement address

Type: regex

Description:

```
<CA0>[.<CA1>]/<IOA0>[.<IOA1>[.<IOA2>]]
```


(C) 2013-2022 OpenDAF Contributors

Examples:

```
10.5/1.2.3 - Common Address 10.5, Information Object Address 1.2.3
3/4.5 - Common Address 3, Information Object Address 3.4
16.8/30 - Common Address 16.8, Information Object Address 30
```

3.11.3 Commands

This stack supports commands.

Command address

Type: regex

Description:

```
[TI<TI>@]<CA0>[.<CA1>]/<IOA0>[.<IOA1>[.<IOA2>]]
```

Examples:

```
10.5/1.2.3 - Write to Common Address 10.5, Information Object Address 1.2.3 using default ASDU TI
TI45@10.5/1.2.3 - Write to Common Address 10.5, Information Object Address 1.2.3 using ASDU with TI45 - Single Command
TI48@3/4.5 - Write to Common Address 3, Information Object Address 3.4 using ASDU with TI48 - Set-point command, normalized value
16.8/30 - Write Common Address 16.8, Information Object Address 30 using default ASDU TI
```


(C) 2013-2022 OpenDAF Contributors

3.12 IEC-62056-21 Tcp Master

3.12.1 Parameters

Parameter	Datatype	Default Value	Description
DEBUG	enum	0	Enable debugging
DUMP	enum	0	Enable communication dump
PEER_ADDRESSES	list_of_inet_addr		Destination TCP/IP hostname:port, separate multiple entries with comma
SCAN_PERIOD	float	30	Scan period [s]
RESPONSE_TIMEOUT	float	20	Response timeout [s]
RECONNECT_DELAY	float	10	Reconnection delay after connection lost
UTC_TIMESTAMPS	enum	1	Transmit/receive timestamps in UTC timezone
DISCONNECT_AFTER_READOUT	enum	0	Disconnect after readout

3.12.2 Measurements

This stack supports measurements.

Measurement address

Type: regex

Description:

```
[<OBIS.A>-<OBIS.B>:]<OBIS-C>.<OBIS-D>.<OBIS-E>[*<OBIS-F>]
where:
    <OBIS.A> - medium type (0 - generic, 1 - electricity, 4 - heat cost allocator, 5 - cooling, 6 - heat, 7 - gas, 8 - water, 0 - hot water)
    <OBIS.B> - channel
    <OBIS.C> - physical quantity (current, voltage, energy, level, temperature, ...)
    <OBIS.D> - quantity computation result of specific algorithm
    <OBIS.E> - individual components of the measurement defined by codes A..D
    <OBIS.F> - unit/tariff/billing period
```

Examples:

```
1.8.0 - Imported energy in kWh
1.8.0*51 - Imported energy in kWh in period 51
1-0:1.8.0 - Electricity, channel 0, total active energy consumption
```

3.12.3 Commands

Commands are not supported on this stack

(C) 2013-2022 OpenDAF Contributors

3.13 Linux Industrial IO connnector

3.13.1 Parameters

Parameter	Datatype	Default Value	Description
DEBUG	enum	0	Enable debugging
SCAN_PERIOD	float	1	Scan period [s] (set to 0 to disable periodic scan)

3.13.2 Measurements

This stack supports measurements.

Measurement address

Type: regex

Description:

```
<device-name>/<channel-name>[.attr][:<scaled|raw>]
```

3.13.3 Commands

This stack supports commands.

Command address

Type: regex

Description:

```
<device-name>/<channel-name>[.attr][:<scaled|raw>]
```


(C) 2013-2022 OpenDAF Contributors

3.14 Kruiz stream receiver

3.14.1 Parameters

Parameter	Datatype	Default Value	Description
DEBUG	enum	0	Enable debugging
DUMP	enum	0	Enable communication dump
HOST	hostname	0.0.0.0	Listen hostname or address (0.0.0.0 = any)
PORT	integer	15001	Listen port
FRAME_SIZE	integer	128	Manual frame size incl. header
OVERRIDE_TIMESTAMP	enum	0	Override outgoing timestamps with current timestamp
RX_TIMEOUT	float	1	Receive timeout [s]

3.14.2 Measurements

This stack supports measurements.

Measurement address

Type: regex

Description:

```
<float32|int16>@<offset>
Offset 0 points to first position in frame payload, not frame header!
```

3.14.3 Commands

Commands are not supported on this stack

(C) 2013-2022 OpenDAF Contributors

3.15 Linux LED subsystem connnector

3.15.1 Parameters

Parameter	Datatype	Default Value	Description
DEBUG	enum	0	Enable debugging

3.15.2 Measurements

Measurements are not supported on this stack

3.15.3 Commands

This stack supports commands.

Command address

Type: regex

Description:

```
<led-name>
```


(C) 2013-2022 OpenDAF Contributors

3.16 microELCOR and microELCOR-2

3.16.1 Parameters

Parameter	Datatype	Default Value	Description
DEBUG	enum	0	Enable debugging
DUMP	enum	0	Enable communication dump
TTY	tty		Communication interface
BAUDRATE	enum	9600	Baud rate
SCAN_PERIOD	float	60.0	Scan period [s]
RESPONSE_TIMEOUT	float	2.0	Response timeout [s]
N_SCAN_RETRIES	int	2	Number of scan retries
WAKE_TIME	float	0.5	Time between rising CTS and starting request transmission [s]
TRIGGER	enum	auto	Readout trigger (auto = each SCAN_PERIOD, manual = by writing start command)
ELCOR_TYPE	enum	microelcor2	Elcor device type

3.16.2 Measurements

This stack supports measurements.

Measurement address

Type: enum

Description:

```
Value label exactly spelled exactly as in document 'Popis systému µ-ELCOR a microELCOR-2 - provozní režim'
```

3.16.3 Commands

This stack supports commands.

Command address

Type: enum

(C) 2013-2022 OpenDAF Contributors

3.17 Modbus/RTU Master

3.17.1 Parameters

Parameter	Datatype	Default Value	Description
DEBUG	enum	0	Enable debugging
DUMP	enum	0	Enable communication dump
TTY	tty		Communication interface
BAUDRATE	enum	115200	Baud rate
PARITY	enum	even	Parity
INTERFRAME_DELAY	float	0.0	Interframe delay (0 = 3.5x character-time) [s]
SCAN_PERIOD	float	1	Scan period [s]
RESPONSE_TIMEOUT	float	0.5	Receive timeout [s]
N_SUCCESSIVE_FAILURES_TO_IGNORE	integer	0	Number of successive scan failures to ignore
TRIGGER	enum	auto	Scan trigger
WRITE_MODE	enum	change	Default command write mode
MERGE_WRITES	enum	1	Merge writes to successive registers
PREFER_FC6	enum	0	Prefer FC6 over FC16 for 1-register writes
LOCKING	enum	0	Lock tty during modbus transactions

3.17.2 Measurements

This stack supports measurements.

Measurement address

Type: regex

Description:

```
<unit-address>/[datatype@]<address-space>:<reference>, where:
    - unit-address is a modbus slave address
    - datatype is used to define object layout if it spans multiple registers
        - format: <S|U|F|X|D><L|B><1|2|4>, where:
            - S - signed
            - U - unsigned
            - F - floating point
            - X - floating point (order 2143)
            - D - discrete
            - L - little-endian (LOW before HIGH)
            - B - big endian (HIGH before LOW)
            - 1 - size is 1 register/coil/discrete input
            - 2 - size is 2 registers/coils/discrete inputs
            - 4 - size is 4 registers/coils/discrete inputs
        - Unsupported datatypes: FL1, FB1, XL1, XB1, DL4, DB4
    - address-space is modbus address space number (0 - coils, 1 - discrete inputs, 3 - input registers, 4 - holding registers)
    - reference is 1-based register, coil or discrete input number in given address space
```


(C) 2013-2022 OpenDAF Contributors

Examples:

```
3/3:1000 - input register 1000 on slave 3 with default datatype (16-bit unsigned)
7/1:550 - discrete input 550 on slave 7
10/fb4@4:1310 - holding registers 1310,1311,1312,1313 on slave 10 mapped as 64-bit big-endian floating-point value
11/dl2@0:35 - coils 35,36 on slave 11 mapped as 2-bit (quaternary) in little-endian (LSB, MSB) order
```

3.17.3 Commands

This stack supports commands.

Command address

Type: regex

Description:

```
<unit-address>/[datatype@]<address-space>:<reference>[!<write-mode>], where:
    - unit-address is a modbus slave address
    - datatype is used to define object layout if it spans multiple registers
        - format: <S|U|F|X|D><L|B><1|2|4>, where:
            - S - signed
            - U - unsigned
            - F - floating point
            - X - floating point (order 2143)
            - D - discrete
            - L - little-endian (LOW before HIGH)
            - B - big endian (HIGH before LOW)
            - 1 - size is 1 register/coil/discrete input
            - 2 - size is 2 registers/coils/discrete inputs
            - 4 - size is 4 registers/coils/discrete inputs
        - Unsupported datatypes: FL1, FB1, XL1, XB1, DL4, DB4
    - address-space is modbus address space number (0 - coils, 4 - holding registers)
    - reference is 1-based register, coil or discrete input number in given address space
    - write-mode is one of: change, trigger, change-and-trigger

special addresses: @trigger
```

Examples:

```
3/4:1000 - holding register 1000 on slave 3 with default datatype (16-bit unsigned)
7/0:550 - coil 550 on slave 7
10/fb4@4:1310 - holding registers 1310,1311,1312,1313 on slave 10 mapped as 64-bit big-endian floating-point value
11/dl2@0:35 - coils 35,36 on slave 11 mapped as 2-bit (quaternary) in little-endian (LSB, MSB) order
@trigger - triggering command, shall be binary
```


(C) 2013-2022 OpenDAF Contributors

3.18 Modbus/TCP Master

3.18.1 Parameters

Parameter	Datatype	Default Value	Description
DEBUG	enum	0	Enable debugging
DUMP	enum	0	Enable communication dump
HOST	hostname		Destination TCP/IP hostname or address
PORT	integer	502	Destination TCP/IP port
SCAN_PERIOD	float	1	Scan period [s]
RESPONSE_TIMEOUT	float	0.5	Receive timeout [s]
N_SUCCESSIVE_FAILURES_TO_IGNORE	integer	0	Number of successive scan failures to ignore
RECONNECT_DELAY	float	10	Reconnection delay after connection lost
TRIGGER	enum	auto	Scan trigger
WRITE_MODE	enum	change	Default command write mode
MERGE_WRITES	enum	1	Merge writes to successive registers
PREFER_FC6	enum	0	Prefer FC6 over FC16 for 1-register writes

3.18.2 Measurements

This stack supports measurements.

Measurement address

Type: regex

Description:

```
<unit-address>/[datatype@]<address-space>:<reference>, where:
    - unit-address is a modbus slave address
    - datatype is used to define object layout if it spans multiple registers
        - format: <S|U|F|X|D><L|B><1|2|4>, where:
            - S - signed
            - U - unsigned
            - F - floating point
            - X - floating point (order 2143)
            - D - discrete
            - L - little-endian (LOW before HIGH)
            - B - big endian (HIGH before LOW)
            - 1 - size is 1 register/coil/discrete input
            - 2 - size is 2 registers/coils/discrete inputs
            - 4 - size is 4 registers/coils/discrete inputs
        - Unsupported datatypes: FL1, FB1, XL1, XB1, DL4, DB4
    - address-space is modbus address space number (0 - coils, 1 - discrete inputs, 3 - input registers, 4 - holding registers)
    - reference is 1-based register, coil or discrete input number in given address space
```

Examples:

```
3/3:1000 - input register 1000 on slave 3 with default datatype (16-bit unsigned)
7/1:550 - discrete input 550 on slave 7
10/fb4@4:1310 - holding registers 1310,1311,1312,1313 on slave 10 mapped as 64-bit big-endian floating-point value
11/dl2@0:35 - coils 35,36 on slave 11 mapped as 2-bit (quaternary) in little-endian (LSB, MSB) order
```


(C) 2013-2022 OpenDAF Contributors

3.18.3 Commands

This stack supports commands.

Command address

Type: regex

Description:

```
<unit-address>/[datatype@]<address-space>:<reference>[!<write-mode>], where:
    - unit-address is a modbus slave address
    - datatype is used to define object layout if it spans multiple registers
        - format: <S|U|F|X|D><L|B><1|2|4>, where:
            - S - signed
            - U - unsigned
            - F - floating point
            - X - floating point (order 2143)
            - D - discrete
            - L - little-endian (LOW before HIGH)
            - B - big endian (HIGH before LOW)
            - 1 - size is 1 register/coil/discrete input
            - 2 - size is 2 registers/coils/discrete inputs
            - 4 - size is 4 registers/coils/discrete inputs
        - Unsupported datatypes: FL1, FB1, XL1, XB1, DL4, DB4
    - address-space is modbus address space number (0 - coils, 4 - holding registers)
    - reference is 1-based register, coil or discrete input number in given address space
    - write-mode is one of: change, trigger, change-and-trigger

special addresses: @trigger
```

Examples:

```
3/4:1000 - holding register 1000 on slave 3 with default datatype (16-bit unsigned)
7/0:550 - coil 550 on slave 7
10/fb4@4:1310 - holding registers 1310,1311,1312,1313 on slave 10 mapped as 64-bit big-endian floating-point value
11/dl2@0:35 - coils 35,36 on slave 11 mapped as 2-bit (quaternary) in little-endian (LSB, MSB) order
@trigger - triggering command, shall be binary
```


(C) 2013-2022 OpenDAF Contributors

3.19 Monarch over streamd client

3.19.1 Parameters

Parameter	Datatype	Default Value	Description
DEBUG	enum	0	Enable debugging
DUMP	enum	0	Enable communication dump
PEER_ADDRESSES	list_of_inet_addr		Destination TCP/IP hostname:port, separate multiple entries with comma
INACTIVITY_TIMEOUT	float	30.0	Inactivity timeout [s]
RECONNECT_DELAY	float	10	Reconnection delay after connection lost
KEEP_VALUE_ON_DISCONNECT	enum	1	When connection is disconnected, only measurement qualities are updated, not values reset
CHANGETIME_FALLBACK	enum	1	When source time is 0, fallback to change time (system time)

3.19.2 Measurements

This stack supports measurements.

Measurement address

Type: regex

Description:

```
Monarch record identification - DBno:object/key
```

3.19.3 Commands

This stack supports commands.

Command address

Type: regex

Description:

```
Monarch record identification - DBno:object/key
```


(C) 2013-2022 OpenDAF Contributors

3.20 MONEZ client

3.20.1 Parameters

Parameter	Datatype	Default Value	Description
DEBUG	enum	0	Enable debugging
DUMP	enum	0	Enable communication dump
HOST	hostname	0.0.0.0	Listen hostname or address (0.0.0.0 = any)
PORT	integer	71	Listen port
INACTIVITY_TIMEOUT	float	1.0	Inactivity timeout [s]

3.20.2 Measurements

This stack supports measurements.

Measurement address

Type: enum

Description:

```
Enter MONEZ measurement name
```

Examples:

3.20.3 Commands

Commands are not supported on this stack

(C) 2013-2022 OpenDAF Contributors

3.21 MQTT client

3.21.1 Parameters

Parameter	Datatype	Default Value	Description
DEBUG	enum	0	Enable debugging
DUMP	enum	0	Enable communication dump
BROKERS	list_of_inet_addr		Broker TCP/IP hostname:port, separate multiple entries with comma
KEEPALIVE_INTERVAL	float	10	Keep-alive interval [s]
RECONNECT_DELAY	float	10	Reconnection delay after connection lost
QOS	enum	2	MQTT publish/subscribe QoS
USERNAME	string		Optional username
PASSWORD	string		Optional password

3.21.2 Measurements

This stack supports measurements.

Measurement address

Type: regex

Description:

```
[[<format>[::<xpath>]]@]topic
```

Examples:

```
/ra/group/A4KVB1
raw@/ra/group/A4KVB1
json::/dynamic_priority@/ra/group/A4KVB1
```

3.21.3 Commands

This stack supports commands.

Command address

Type: regex

Description:

```
[[<format>[::<pattern>]]@][!]topic
```

Examples:

```
!/ra/group/A4KVB1
raw@/ra/group/A4KVB1
json::{"value": $value, "timestamp": $timestamp}@/ra/group/A4KVB1
```


(C) 2013-2022 OpenDAF Contributors

3.22 MRE-10DO 10x digital output module

3.22.1 Parameters

Parameter	Datatype	Default Value	Description
ADDRESS	int	112	Module address (112 + as configured via jumpers)
DEBUG	enum	0	Enable debugging

3.22.2 Measurements

Measurements are not supported on this stack

3.22.3 Commands

This stack supports commands.

Command address

Type: int

Description:

```
Digital output number
```

Examples:

```
0 - first digital output
9 - last digital output
```


(C) 2013-2022 OpenDAF Contributors

3.23 MRE-12AI 12x analog input module

3.23.1 Parameters

Parameter	Datatype	Default Value	Description
ADDRESS	int	80	Module address (80 + as configured via jumpers)
DEBUG	enum	0	Enable debugging
SCAN_PERIOD	float	1.0	Scan period [s]

3.23.2 Measurements

This stack supports measurements.

Measurement address

Type: regex

Description:

```
<input-range>@<position>, where:
    - <input-range> must correspond to module hardware and is one of:
        - 4-20mA
        - +-2.5V
        - pt100
        - ntc:<R0>:<BETA>
    - <position> is the input position, 0 - 11
```

Examples:

```
4-20mA@0 - 4-20 mA current loop on input 0
4-20mA@6 - 4-20 mA current loop on input 6
+-2.5V@3 - ±2.5 V range on input 3
pt100@10 - PT100 RTD on input 10
ntc:10000:3955@7 - NTC with 10kOhm nominal resistance and Beta=3955 on input 7
ntc:1000:3528@8 - NTC with 1kOhm nominal resistance and Beta=3528 on input 8
```

3.23.3 Commands

Commands are not supported on this stack

(C) 2013-2022 OpenDAF Contributors

3.24 MRE-16DI 16x digital input module

3.24.1 Parameters

Parameter	Datatype	Default Value	Description
ADDRESS	int	88	Module address (88 + as configured via jumpers)
DEBUG	enum	0	Enable debugging
SCAN_PERIOD	float	1.0	Scan period [s]

3.24.2 Measurements

This stack supports measurements.

Measurement address

Type: int

Description:

```
Digital input number
```

Examples:

```
0 - first digital input
15 - last digital input
```

3.24.3 Commands

Commands are not supported on this stack

(C) 2013-2022 OpenDAF Contributors

3.25 MRE-4AO 4x 0-10V analog output module

3.25.1 Parameters

Parameter	Datatype	Default Value	Description
ADDRESS	int	96	Module address (96 + as configured via jumpers)
DEBUG	enum	0	Enable debugging

3.25.2 Measurements

Measurements are not supported on this stack

3.25.3 Commands

This stack supports commands.

Command address

Type: regex

Description:

```
Analog output number or ! (eeprom write command)
```

Examples:

```
0 - first analog output
3 - last analog output
! - write current state to eeprom
```


(C) 2013-2022 OpenDAF Contributors

3.26 I&C O_GATE client

3.26.1 Parameters

Parameter	Datatype	Default Value	Description
DEBUG	enum	0	Enable debugging
DUMP	enum	0	Enable communication dump
PEER_ADDRESSES	list_of_inet_addr		Destination TCP/IP hostname:port, separate multiple entries with comma
SAMPLING_PERIOD	float	1	Sampling period [s]
INACTIVITY_TIMEOUT	float	1.0	Inactivity timeout [s]
RECONNECT_DELAY	float	10	Reconnection delay after connection lost

3.26.2 Measurements

This stack supports measurements.

Measurement address

Type: regex

Description:

```
Enter P_LDB tagname
```

Examples:

3.26.3 Commands

Commands are not supported on this stack

(C) 2013-2022 OpenDAF Contributors

3.27 OCPP 1.6 Central System

3.27.1 Parameters

Parameter	Datatype	Default Value	Description
DEBUG	enum	0	Enable debugging
HOST	hostname	0.0.0.0	Listen hostname or address (0.0.0.0 = any)
PORT	integer	9090	Listen port
CONN_LIMIT	integer	4	Maximum number of parallel client connections

3.27.2 Measurements

This stack supports measurements.

Measurement address

Type: regex

Description:

```
<connection-point>/<measurement-structured-path>]
```

3.27.3 Commands

Commands are not supported on this stack

(C) 2013-2022 OpenDAF Contributors

3.28 OGate Gateway client (using notification protocol)

3.28.1 Parameters

Parameter	Datatype	Default Value	Description
DEBUG	enum	0	Enable debugging
DUMP	enum	0	Enable communication dump
PEER_ADDRESSES	list_of_inet_addr		Destination TCP/IP hostname:port, separate multiple entries with comma
SPR	boolean	False	Seconds-only precision
VPL	boolean	True	Variable packet length
INACTIVITY_TIMEOUT	float	10	Inactivity timeout [s]
COMMIT_TIMEOUT	float	0.25	Commit timeout [s]
COMMIT_LIMIT	integer	1000	Commit limit
RECONNECT_DELAY	float	10	Reconnection delay after connection lost
QUANTIZE_QUALITY	boolean	False	Enable quality quantization (filtering-out everything except for quality class)
OVERRIDE_TIMESTAMP	boolean	False	Override measurement timestamps with local timestamp

3.28.2 Measurements

This stack supports measurements.

Measurement address

Type: regex

Description:

```
Enter OGate Gateway tagname
```

Examples:

3.28.3 Commands

Commands are not supported on this stack

(C) 2013-2022 OpenDAF Contributors

3.29 OneWire Connnector

3.29.1 Parameters

Parameter	Datatype	Default Value	Description
DEBUG	enum	0	Enable debugging
SCAN_PERIOD	float	10	Scan period [s]
THREAD_POOL_SIZE	int	1	Number of scan threads

3.29.2 Measurements

This stack supports measurements.

Measurement address

Type: regex-literal

Description:

```
The regex will be matched against onewire device node strings formatted as XX-YYYYYYYYYYY where:
- XX represents the device family as a hexadecimal number
- YYYYYYYYYYY represents the device ID as a hexadecimal number
```

Examples:

```
10\-.* - match any device of 0x10 family
10\-C03F.* - match any device of 0x10 family, whose ID upper bytes are 0xC0 and 0x3F
..\-CDAB014522FE - match device with ID CDAB014522FE regardless of family
(10|28)\-.* - match any device of 0x10 or 0x28 family
```

3.29.3 Commands

Commands are not supported on this stack

(C) 2013-2022 OpenDAF Contributors

3.30 OPC-UA Client

3.30.1 Parameters

Parameter	Datatype	Default Value	Description
URL	string		OPC-UA server endpoint URL(s), comma-separated for round-robin failover
DEFAULT_NAMESPACE	integer	1	Default namespace
USERNAME	string		Username (leave empty for anonymous access)
PASSWORD	string		Password
TIMEOUT	float	5	Network timeout [s]
SCAN_PERIOD	float	1	Poll period [s]
RECONNECT_INTERVAL	float	1	Reconnection interval after connection loss [s]
MAX_ITEMS_PER_SUB	integer	5000	Maximum monitored items per OPC-UA subscription
DEBUG	enum	0	Enable debugging

3.30.2 Measurements

This stack supports measurements.

Measurement address

Type: regex

Description:

```
[uatype@]<nodeId>

uatype (optional): boolean | sbyte | byte | int16 | uint16 | int32 | uint32 | int64 | uint64 | float | double | string
  If omitted, the CO's configured raw datatype is used.

nodeId: OPC-UA node identifier string
  ns=<index>;i=<number>   - numeric node id
  ns=<index>;s=<name>     - string node id
  ns=<index>;<name>       - string node id
  i=<number>              - numeric node id in default namespace
  s=<name>                - string node id in default namespace
  <name>                  - string node id in default namespace
```

Examples:

```
ns=2;i=1001
ns=2;s=Sensors.Temperature
float@ns=2;s=Room.Humidity
int32@i=12345
Room.Humidity
```

3.30.3 Commands

This stack supports commands.

Command address

Type: regex

(C) 2013-2022 OpenDAF Contributors

Description:

```
[uatype@]<nodeId>

uatype (optional): boolean | sbyte | byte | int16 | uint16 | int32 | uint32 | int64 | uint64 | float | double | string
  If omitted, the CO's configured raw datatype is used.

nodeId: OPC-UA node identifier string
  ns=<index>;i=<number>   - numeric node id
  ns=<index>;s=<name>     - string node id
  ns=<index>;<name>       - string node id
  i=<number>              - numeric node id in default namespace
  s=<name>                - string node id in default namespace
  <name>                  - string node id in default namespace
```

Examples:

```
ns=2;i=2001
ns=2;s=Actuators.Setpoint
double@ns=2;s=Controller.Setpoint
Controller.Setpoint
```


(C) 2013-2022 OpenDAF Contributors

3.31 OPC client

3.31.1 Parameters

Parameter	Datatype	Default Value	Description
DEBUG	enum	0	Enable debugging
DUMP	enum	0	Enable communication dump
PEER_ADDRESSES	list_of_inet_addr		Destination TCP/IP hostname:port, separate multiple entries with comma
SAMPLING_PERIOD	float	1	Sampling period [s]
INACTIVITY_TIMEOUT	float	1.0	Inactivity timeout [s]
RECONNECT_DELAY	float	10	Reconnection delay after connection lost
OPC_HOSTNAME	string	localhost	OPC server hostname (use localhost for OPC server running on the same machine as the OPC agent)
OPC_SERVER	string		OPC server application ID (i.e. Kepware.KEPServerEX.V5)

3.31.2 Measurements

This stack supports measurements.

Measurement address

Type: string

Description:

```
Enter OPC tagname
```

3.31.3 Commands

This stack supports commands.

Command address

Type: string

Description:

```
Enter OPC tagname
```


(C) 2013-2022 OpenDAF Contributors

3.32 SEAK Panter Master

3.32.1 Parameters

Parameter	Datatype	Default Value	Description
DEBUG	enum	0	Enable debugging
DUMP_TTY	enum	0	Dump raw data as received/transmitted from/to tty
DUMP_FRAMES	enum	0	Dump received/transmitted modbus frames
TTY	tty		Communication interface
SCAN_PERIOD	float	10.0	Periodic scan period [s]
STATUS_QUERY_PERIOD	float	0.2	Status query period [s]
RESET_HOLDOFF	float	5	Reset hold-off time [s]
RESPONSE_TIMEOUT	float	0.5	Receive timeout [s]
PERMANENT_ERROR_TIMEOUT	float	600	Permanent error timeout [s]
REQUEST_TTL	float	600	Request time-to-live [s]
EXECUTION_DELAY	float	0	Delay between command write and its execution [s]
CONTROLLER_QUEUE	integer	16384	Per-controller transmit queue size [B]
BROADCAST_REPS	integer	0	Number of broadcast commands repetitions (adjust CONTROLLER_QUEUE and REQUEST_TTL accordingly)
MODULATION_RETRIES	integer	2	Number of modulation retries after modulator exception
BAUDRATE	enum	9600	Baud rate
PARITY	enum	even	Parity
INTERFRAME_DELAY	float	0.0	Interframe delay (0 = 3.5x character-time) [s]
ASYNCH_ENABLE	enum	0	Enable asynchronous reception
LOCKING	enum	0	Lock tty during bus transactions
REPORT_CHANGES_ONLY	enum	1	Report command counters, permanent error and queue state only on change (0 restores the legacy 1 Hz streams)
PUBLISH_INTERNAL_FSM_STATES	enum	0	Publish internal FSM states (every transition incl. internal queries; 0 publishes only external state changes)

3.32.2 Measurements

This stack supports measurements.

(C) 2013-2022 OpenDAF Contributors

Measurement address

Type: string

Description:

Examples:

3.32.3 Commands

This stack supports commands.

Command address

Type: string

Examples:

(C) 2013-2022 OpenDAF Contributors

3.33 Persistent Storage Connector

3.33.1 Parameters

Parameter	Datatype	Default Value	Description
DEBUG	enum	0	Enable debugging
STORAGE_PATH	string	/srv	Storage path
MAP_SIZE	int	4	Map size [MB]

3.33.2 Measurements

This stack supports measurements.

Measurement address

Type: regex

3.33.3 Commands

This stack supports commands.

Command address

Type: regex

(C) 2013-2022 OpenDAF Contributors

3.34 PTNet controlling station

3.34.1 Parameters

Parameter	Datatype	Default Value	Description
DEBUG	enum	0	Enable debugging
DUMP	enum	0	Enable communication dump
LOG_ASDUS	enum	0	Enable human-readable ASDU logging
PEER_ADDRESSES	list_of_inet_addr		Destination TCP/IP hostname:port, separate multiple entries with comma
RECONNECT_DELAY	float	10	Reconnection delay after connection lost
T_BACKEND_TIMEOUT	float	900	Invalidate measurements only if backend is disconnected longer than configured value [s] (0 = disable)
T_IC_PERIOD	float	60	Automatic interrogation period [s] (0 = disabled)
T_IC_TIMEOUT	float	5	Interrogation command termination timeout [s]
T_RD_TIMEOUT	float	1	Read command completion timeout [s]

3.34.2 Measurements

This stack supports measurements.

Measurement address

Type: string

3.34.3 Commands

This stack supports commands.

Command address

Type: string

(C) 2013-2022 OpenDAF Contributors

3.35 PTNet/Tinymesh controlling station

3.35.1 Parameters

Parameter	Datatype	Default Value	Description
DEBUG	enum	0	Enable debugging
DUMP	enum	0	Enable communication dump
LOG_ASDUS	enum	0	Enable human-readable ASDU logging
PEER_ADDRESSES	list_of_inet_addr		Destination TCP/IP hostname:port, separate multiple entries with comma
RECONNECT_DELAY	float	10	Reconnection delay after connection lost
T_PRM_TIMEOUT	float	5	Primary message ack timeout
T_IC_PERIOD	float	60	Automatic interrogation period (0 = disabled)
T_IC_TIMEOUT	float	5	Interrogation command termination timeout
T_RD_TIMEOUT	float	1	Read command completion timeout
TRANSPORT_MTU	int	21	MTU transport layer would accept
MULTIPLE_DOWNLINK_IES	enum	0	Allow multiple information elements in one downlink packet

3.35.2 Measurements

This stack supports measurements.

Measurement address

Type: string

3.35.3 Commands

This stack supports commands.

Command address

Type: string

(C) 2013-2022 OpenDAF Contributors

3.36 Linux PWM subsystem connnector

3.36.1 Parameters

Parameter	Datatype	Default Value	Description
DEBUG	enum	0	Enable debugging
PERIOD	float	1000000	PWM period [ns]

3.36.2 Measurements

Measurements are not supported on this stack

3.36.3 Commands

This stack supports commands.

Command address

Type: regex

Description:

```
[!]chip-number/pwm-index
```


(C) 2013-2022 OpenDAF Contributors

3.37 QERP 408/416 Power Factor Regulator

3.37.1 Parameters

Parameter	Datatype	Default Value	Description
DEBUG	enum	0	Enable debugging
DUMP	enum	0	Enable communication dump
TTY	tty		Communication interface
SCAN_PERIOD	float	10.0	Scan period [s]
RESPONSE_TIMEOUT	float	0.5	Response timeout [s]

3.37.2 Measurements

This stack supports measurements.

Measurement address

Type: regex

3.37.3 Commands

Commands are not supported on this stack

(C) 2013-2022 OpenDAF Contributors

3.38 Romet ECS

3.38.1 Parameters

Parameter	Datatype	Default Value	Description
DEBUG	enum	0	Enable debugging
DUMP	enum	0	Enable communication dump
TTY	tty		Communication interface

3.38.2 Measurements

This stack supports measurements.

Measurement address

Type: enum

3.38.3 Commands

This stack supports commands.

Command address

Type: enum

(C) 2013-2022 OpenDAF Contributors

3.39 Rotafon digitizer interface

3.39.1 Parameters

Parameter	Datatype	Default Value	Description
DEBUG	enum	0	Enable debugging
DUMP_BYTES	enum	0	Enable communication dump on TTY level
DUMP_PACKETS	enum	0	Enable communication dump on datalink level
TTY	tty	ttyROT	Communication interface
BAUDRATE	enum	3000000	Baud rate
PARITY	enum	none	Parity
T_RX_TIMEOUT	float	0.25	Frame reception timeout [s]
T_RSP_TIMEOUT	float	1	Response timeout [s]
AUTOSTART	enum	1	Start sampling after connect
RECORDS_PER_TXN	int	10	Number of records to pack into one transaction
SAMPLE_RATE	float	250	Digitizer sample rate [Hz]
EST_ENABLE	enum	0	Enable timestamp estimator
EST_LOCK_LATENCY	int	1	Timestamp estimator - lock latency [record]
EST_LOCK_VALID_FOR	int	2	Timestamp estimator - lock valid for [record]
EST_LOCK_INVALID_AFTER	int	20	Timestamp estimator - lock invalid after [record]

3.39.2 Measurements

This stack supports measurements.

Measurement address

Type: enum

3.39.3 Commands

This stack supports commands.

Command address

Type: enum

(C) 2013-2022 OpenDAF Contributors

3.40 Simatic S7 client

3.40.1 Parameters

Parameter	Datatype	Default Value	Description
DEBUG	enum	0	Enable debugging
DUMP	enum	0	Enable communication dump
PEER_ADDRESSES	list_of_inet_addr		Destination TCP/IP hostname:port, separate multiple entries with comma
PORT	int	102	Default destination TCP port
SCAN_PERIOD	float	0.25	Scan period [s]
RESPONSE_TIMEOUT	float	0.25	Receive timeout [s]
RECONNECT_DELAY	float	2	Reconnection delay after connection lost
DISCONNECT_STOPPED	enum		Disconnect from PLC when it enters STOP state
BIT_QUALITY_LOGIC	enum		Bit quality logic (applies only if quality is bit, negative - false is good, positive - true is good)
VENDOR_QUALITY	enum	1	Store quality in vendor quality bits
RACK	int	0	PLC rack
SLOT	int	1	PLC slot
PDU_SIZE_LIMIT	int	240	PDU size limit
PDU_MAX_VARS	int	10	Maximum variables per PDU
MIN_READ_COUNT	enum	0	Minimize read count by reading unconfigured areas
ADDRESS	string		Destination TCP/IP hostname or address (legacy, use PEER_ADDRESSES)

3.40.2 Measurements

This stack supports measurements.

Measurement address

Type: regex

Description:

Address consists of mandatory value reference and optional timestamp and quality references:
`<value-reference>[,t=<timestamp-reference>][,q=<quality-reference>]`

(C) 2013-2022 OpenDAF Contributors

There are two supported reference syntaxes:

1. Simatic syntax - see vendor documentation
2. Native syntax: `<type>@<area>[<block>:]<offset>[.<bit|length>]` where:
   - `type` is one of `bit`, `bool`, `byte`, `usint`, `word`, `uint`, `counter`, `timer`, `dword`, `udint`, `char`, `sint`, `int`, `dint`, `float`, `dtl`, `string`, `s5t`
   - `area` is one of `pe`, `pa`, `mk`, `db`, `ct`, `tm`
   - `block` is block number
   - `offset` is offset in block
   - `bit|length` bit number for `bit` and `bool` types, mandatory maximum string length for `string` type

Examples:

```
float@DB20:150
bit@DB81:107.0
word@DB80:126,t=dtl@DB80:0
bit@DB80:124.2,t=dtl@DB80:0
float@DB400:916,t=dtl@DB400:0,q=byte@DB400:904
string@DB250:10.25
DB10.DBX148.2
DB10.I274
DB69.S112.10
```

3.40.3 Commands

This stack supports commands.

Command address

Type: regex

Description:

Address consists of mandatory value reference and optional timestamp and quality references:
`<value-reference>[,t=<timestamp-reference>]`

There are two supported reference syntaxes:

1. Simatic syntax - see vendor documentation
2. Native syntax: `<type>@<area>[<block>:]<offset>[.<bit|length>]` where:
   - `type` is one of `bit`, `bool`, `byte`, `usint`, `word`, `uint`, `counter`, `timer`, `dword`, `udint`, `char`, `sint`, `int`, `dint`, `float`, `dtl`, `string`
   - `area` is one of `pe`, `pa`, `mk`, `db`, `ct`, `tm`
   - `block` is block number
   - `offset` is offset in block
   - `bit|length` bit number for `bit` and `bool` types, mandatory maximum string length for `string` type

Examples:

```
bit@DB350:200.0
bit@DB350:100.2,t=dtl@DB350:0
word@DB350:392
string@DB350:10.25
DB10.DBX148.2
DB10.I274
DB69.S112.10
```


(C) 2013-2022 OpenDAF Contributors

3.41 SCORPIO client

3.41.1 Parameters

Parameter	Datatype	Default Value	Description
DEBUG	enum	0	Enable debugging
DUMP	enum	0	Enable communication dump
PEER_ADDRESSES	list_of_inet_addr		Destination TCP/IP hostname:port, separate multiple entries with comma
INACTIVITY_TIMEOUT	float	60.0	Inactivity timeout [s]
RECONNECT_DELAY	float	10	Reconnection delay after connection lost
BLOCK	integer	3	Plant block number to read values from

3.41.2 Measurements

This stack supports measurements.

Measurement address

Type: regex

Description:

```
<array>,<index> or <array>_<type><index>
```

Examples:

3.41.3 Commands

Commands are not supported on this stack

(C) 2013-2022 OpenDAF Contributors

3.42 Access to OpenDAF internal state

3.42.1 Parameters

Parameter	Datatype	Default Value	Description

3.42.2 Measurements

This stack supports measurements.

Measurement address

Type: regex

Description:

```
Supported addresses:
quality(measurement-name)
```

3.42.3 Commands

Commands are not supported on this stack

(C) 2013-2022 OpenDAF Contributors

3.43 SFLINT Luminance Photometer

3.43.1 Parameters

Parameter	Datatype	Default Value	Description
DEBUG	enum	0	Enable debugging
DUMP	enum	0	Enable communication dump
TTY	tty		Communication interface
BAUDRATE	enum	115200	Baud rate
PARITY	enum	none	Parity
SCAN_PERIOD	float	10.0	Scan period [s]
RESPONSE_TIMEOUT	float	0.5	Response timeout [s]
LOCKING	enum	0	Lock tty during transactions

3.43.2 Measurements

This stack supports measurements.

Measurement address

Type: regex

Description:

```
<address>/<ea|avg_ea|t_cycle>
```

Examples:

```
66/ea - ambient illuminance on sensor 66
66/avg_ea - average ambient illuminance on sensor 66
66/t_cycle - evaluation period on sensor 66
```

3.43.3 Commands

This stack supports commands.

Command address

Type: regex

Description:

```
<address>/t_cycle
```

Examples:

```
66/t_cycle - evaluation period setpoint on sensor 66
```


(C) 2013-2022 OpenDAF Contributors

3.44 Signal generator

3.44.1 Parameters

Parameter	Datatype	Default Value	Description
MODE	enum	measurements	Generator mode

3.44.2 Measurements

This stack supports measurements.

Measurement address

Type: regex

Examples:

```
constant(i10)
square(period)
sin(period, nStepsPerPeriod, phase[deg])
saw(period, nStepsPerPeriod)
```

3.44.3 Commands

This stack supports commands.

Command address

Type: regex

Examples:

```
constant(i10)
square(period)
sin(period, nStepsPerPeriod, phase[deg])
saw(period, nStepsPerPeriod)
```


(C) 2013-2022 OpenDAF Contributors

3.45 SNMP manager

3.45.1 Parameters

Parameter	Datatype	Default Value	Description
PEER	regex		Destination agent address
COMMUNITY	string	public	SNMP community
SCAN_PERIOD	float	1	Scan period [s]
TIMEOUT	float	10.0	Response timeout [s]
RETRIES	int	3	Number of query retries
VERSION	enum	2c	SNMP protocol version
N_OIDS_PER_PDU	int	16	Number of OIDs requested in one PDU
DEBUG	enum	0	Enable debugging

3.45.2 Measurements

This stack supports measurements.

Measurement address

Type: ASN.1

Description:

```
ASN.1 object specification. Always prefix symbolic name with MIB name.
```

Examples:

```
.1.3.6.1.3.10
.1.3.6.1.2.1.1
RFC1213-MIB::sysDescr.0
HOST-RESOURCES-MIB::hrSystemUptime.0
```

3.45.3 Commands

Commands are not supported on this stack

(C) 2013-2022 OpenDAF Contributors

3.46 Communication object Per Row SQL client

3.46.1 Parameters

Parameter	Datatype	Default Value	Description
DEBUG	enum	0	Enable debugging
SAMPLING_PERIOD	float	1	Sampling period [s]
CONNECTION_STRING	string		Database connection string (://)
TABLE	regex		Database table name
COL_NAME	regex		Identification column name
COL_VALUE	regex		Value column name
COL_TIMESTAMP	regex		Timestamp column name (leave empty to use local timestamps)
COL_QUALITY	regex		Quality column name (leave empty to always use good quality)
WATCHDOG	string		Watchdog row identification (leave empty to disable watchdog)
MAX_WATCHDOG_AGE	float	60	Maximum watchdog age
LOCK	string		Session lock (consider using ODBCLock when connecting to ODBC datasource)

3.46.2 Measurements

This stack supports measurements.

Measurement address

Type: string

Description:

```
Row identifier to lookup in identification column
```

Examples:

3.46.3 Commands

Commands are not supported on this stack

(C) 2013-2022 OpenDAF Contributors

3.47 EBO dosimetry SQL client

3.47.1 Parameters

Parameter	Datatype	Default Value	Description
DEBUG	enum	0	Enable debugging
SAMPLING_PERIOD	float	1	Sampling period [s]
CONNECTION_STRING	regex		Database connection string
TABLE	regex		Database table name
COLUMN_NAME_PRECISION	integer	2	Column name precision (1 -> V1, 2 -> V01, 3 -> V001)
MAX_AGE	float	60	Maximum table row timestamp age (timespan from last timestamp change)
LOCK	string	ODBCLock	Session lock (consider using ODBCLock when connecting to ODBC datasource)

3.47.2 Measurements

This stack supports measurements.

Measurement address

Type: int

Description:

```
Column name derived index (V245 -> 245, V002 -> 2)
```

Examples:

3.47.3 Commands

Commands are not supported on this stack

(C) 2013-2022 OpenDAF Contributors

3.48 SuHub client (using notification protocol)

3.48.1 Parameters

Parameter	Datatype	Default Value	Description
DEBUG	enum	0	Enable debugging
DUMP	enum	0	Enable communication dump
PEER_ADDRESSES	list_of_inet_addr		Destination TCP/IP hostname:port, separate multiple entries with comma
SPR	boolean	False	Seconds-only precision
INACTIVITY_TIMEOUT	float	10	Inactivity timeout [s]
RECONNECT_DELAY	float	10	Reconnection delay after connection lost
OVERRIDE_TIMESTAMP	boolean	False	Override measurement timestamps with local timestamp

3.48.2 Measurements

This stack supports measurements.

Measurement address

Type: regex

Description:

```
Enter SuHub tagname
```

Examples:

3.48.3 Commands

Commands are not supported on this stack

(C) 2013-2022 OpenDAF Contributors

3.49 SWTN Controlling Station

3.49.1 Parameters

Parameter	Datatype	Default Value	Description
DEBUG	enum	0	Enable debugging
DUMP	enum	0	Enable communication dump
TTY	tty		Communication interface
BAUDRATE	enum	19200	Baud rate
PARITY	enum	none	Parity
T_RX_TIMEOUT	float	1	Frame reception timeout [s]
T_TX_IDLE	float	0.064	After-frame idle period [s]
MTU	int	32	Link-layer frame MTU [B]
T_FRAME_TTL	float	10	Frame time-to-live [s]
T_PAYLOAD_TTL	float	60	Application-layer payload time-to-live [s]
T_FORCED_COMMAND_REFRESH	float	3600	Forced command refresh period [s]
T_MEASUREMENT_VALID	float	3600	Measurement validity period after latest update [s]

3.49.2 Measurements

This stack supports measurements.

Measurement address

Type: regex

Description:

```
<UID3>.<UID2>.<UID1>.<UID0>/<IOA> or 0x<UID hex>/<IOA>
```

Examples:

```
10.5.3.8/123 - UID 10.5.3.8 (0x0A050308), Information Object Address 123
0xAB01E0CD/25 - UID 171.1.224.205 (0xAB01E0CD), Information Object Address 25
```

3.49.3 Commands

This stack supports commands.

Command address

Type: regex

Description:

```
[TI<TI>@]<UID3>.<UID2>.<UID1>.<UID0>/<IOA> or [TI<TI>@]0x<UID hex>/<IOA>
```


(C) 2013-2022 OpenDAF Contributors

Examples:

```
10.5.3.8/123 - UID 10.5.3.8 (0x0A050308), Information Object Address 123, use implicit information element type
0xAB01E0CD/25 - UID 171.1.224.205 (0xAB01E0CD), Information Object Address 25, use implicit information element type
TI86@10.5.3.8/123 - UID 10.5.3.8 (0x0A050308), Information Object Address 123, use information element type 86
TI53@0xAB01E0CD/25 - UID 171.1.224.205 (0xAB01E0CD), Information Object Address 25, use information element type 53
```


(C) 2013-2022 OpenDAF Contributors

3.50 SWTN Controlling Station

3.50.1 Parameters

Parameter	Datatype	Default Value	Description
DEBUG	enum	0	Enable debugging
DUMP	enum	0	Enable communication dump
PEER_ADDRESSES	list_of_inet_addr		Destination TCP/IP hostname:port, separate multiple entries with comma
RECONNECT_DELAY	float	10	Reconnection delay after connection lost
T_FRAME_TTL	float	10	Frame time-to-live [s]
T_PAYLOAD_TTL	float	60	Application-layer payload time-to-live [s]
T_FORCED_COMMAND_REFRESH	float	3600	Forced command refresh period [s]
T_MEASUREMENT_VALID	float	3600	Measurement validity period after latest update [s]

3.50.2 Measurements

This stack supports measurements.

Measurement address

Type: regex

Description:

```
<UID3>.<UID2>.<UID1>.<UID0>/<IOA> or 0x<UID hex>/<IOA>
```

Examples:

```
10.5.3.8/123 - UID 10.5.3.8 (0x0A050308), Information Object Address 123
0xAB01E0CD/25 - UID 171.1.224.205 (0xAB01E0CD), Information Object Address 25
```

3.50.3 Commands

This stack supports commands.

Command address

Type: regex

Description:

```
[TI<TI>@]<UID3>.<UID2>.<UID1>.<UID0>/<IOA> or [TI<TI>@]0x<UID hex>/<IOA>
```

Examples:

```
10.5.3.8/123 - UID 10.5.3.8 (0x0A050308), Information Object Address 123, use implicit information element type
0xAB01E0CD/25 - UID 171.1.224.205 (0xAB01E0CD), Information Object Address 25, use implicit information element type
TI86@10.5.3.8/123 - UID 10.5.3.8 (0x0A050308), Information Object Address 123, use information element type 86
TI53@0xAB01E0CD/25 - UID 171.1.224.205 (0xAB01E0CD), Information Object Address 25, use information element type 53
```


(C) 2013-2022 OpenDAF Contributors

3.51 Teledosimetry client

3.51.1 Parameters

Parameter	Datatype	Default Value	Description
DEBUG	enum	0	Enable debugging
DUMP	enum	0	Enable communication dump
PEER_ADDRESSES	list_of_inet_addr		Destination TCP/IP hostname:port, separate multiple entries with comma
RECONNECT_DELAY	float	10	Reconnection delay after connection lost
LOCAL_TIMESTAMP	enum	1	Override timestamp with local timestamp
INACTIVITY_TIMEOUT	float	300	Inactivity timeout [s]
USERNAME	string		User name
PASSWORD	string		Password
CLIENT_ID	integer		Client ID
CLIENT_TYPE	integer		Client type

3.51.2 Measurements

This stack supports measurements.

Measurement address

Type: regex

Description:

```
Enter P_LDB tagname
```

Examples:

3.51.3 Commands

Commands are not supported on this stack

(C) 2013-2022 OpenDAF Contributors

3.52 TG800 Central Station

3.52.1 Parameters

Parameter	Datatype	Default Value	Description
DEBUG	enum	0	Enable debugging
DUMP	enum	0	Enable communication dump
TTY	tty		Communication interface
BAUDRATE	enum	2400	Baud rate (must be equal for all stations on the same bus)
PARITY	enum	even	Parity (must be equal for all stations on the same bus)
FLOWCONTROL	enum	rts-set	Flow control mode using RTS/CTS lines (use rts-set on OpenBox RTUs)
N_RETRIES	int	2	Maximum number of transaction retries (must be equal for all stations on the same bus)
RX_TIMEOUT	float	0.8	Receive timeout [s] (must be equal for all stations on the same bus)
TX_BUFFER	int	16384	Transmit buffer size
LINK_ADDRESS	int		Remote substation link address
POLL_PERIOD	float	1.0	Outstation link-layer poll period [s] (must be equal for all stations on the same bus)
T_CLOCKSYNC	float	3600	Clock synchronization period [s] (0 = disabled)
T_GA	float	60	General interrogation period [s] (0 = execute only once)
T_GA_TERM	float	10	General interrogation termination timeout [s]
GA_SYSIND_VALIDATE	enum	1	System Indications GA response validation enable
UTC_TIMESTAMPS	enum	1	Transmit/receive timestamps in UTC timezone
DEFAULT_STATION	int	0	Default station address
DEFAULT_CUBICLE	int	0	Default cubicle number
DEFAULT_CHASSIS	int	0	Default chassis number
STATION_NUMBER	int	1022	This station number
OVERRIDE_TIMESTAMPS	enum	0	Override timestamps on data received

3.52.2 Measurements

This stack supports measurements.

Measurement address

Type: regex

(C) 2013-2022 OpenDAF Contributors

Description:

```
[<IAC>@][HW:][[[<station>.]<cubicle>.]<chassis>.]<slot>.<point>
[<IAC>@]SW:[<station>.]<point>
```

Examples:

```
1.5 - hardware address, implicit IAC, default station, cubicle and chassis, slot 1, point 5
HW:3.0.2.2.0 - hardware address, implicit IAC, station 3, cubicle 0, chassis 2, slot 2, point 0
SW:10 - software address, implicit IAC, default station, point 10
3@SW:63.2400 - software address, IAC 3 - measured value, station 63, point 2400
```

3.52.3 Commands

This stack supports commands.

Command address

Type: regex

Description:

```
[<IAC>@][HW:][[[<station>.]<cubicle>.]<chassis>.]<slot>.<point>
[<IAC>@]SW:[<station>.]<point>
```

Examples:

```
1.5 - hardware address, implicit IAC, default station, cubicle and chassis, slot 1, point 5
128@HW:3.0.2.2.0 - hardware address, IAC 128 - pulse command, statio 3, cubicle 0, chassis 2, slot 2, point 0
SW:10 - software address, implicit IAC, default station, point 10
131@SW:63.2400 - software address, IAC 131 - setpoing, station 63, point 2400
```


(C) 2013-2022 OpenDAF Contributors

3.53 Tinymesh master

3.53.1 Parameters

Parameter	Datatype	Default Value	Description
DEBUG	enum	0	Enable debugging
DUMP	enum	0	Enable communication dump
PEER_ADDRESSES	list_of_inet_addr		Destination TCP/IP hostname:port, separate multiple entries with comma
RECONNECT_DELAY	float	10	Reconnection delay after connection lost
T_RD_TIMEOUT	float	1	Read command completion timeout [s]

3.53.2 Measurements

This stack supports measurements.

Measurement address

Type: regex

Examples:

```
mv 00:cd:a0:3e/adc1
mv 00:cd:a0:3e/di3
```

3.53.3 Commands

This stack supports commands.

Command address

Type: regex

Examples:

```
ex ff:ff:ff:ff/pwm
c_cfg 00:cd:a0:3e/95
```


(C) 2013-2022 OpenDAF Contributors

3.54 Teleperm-XS client (via GWD)

3.54.1 Parameters

Parameter	Datatype	Default Value	Description
DEBUG	enum	0	Enable debugging
DUMP	enum	0	Enable communication dump
PEER_ADDRESSES	list_of_inet_addr		Destination TCP/IP hostname:port, separate multiple entries with comma
ACNT	integer		Number of analog blocks, must match acnt in gwd.conf
BCNT	integer		Number of binary blocks, must match bcnt in gwd.conf
SIGNAL_BASE	integer	1	Signal numbering base index
RECONNECT_DELAY	float	10	Reconnection delay after connection lost
INACTIVITY_TIMEOUT	float	5	Inactivity timeout [s]
ALARM_HOLD_TIME	float	2	Alarm signal hold time [s]
LOCAL_TIMESTAMP	enum	0	Override timestamp with local timestamp
MAP_TO_BAD_QUALITY	enum	0	Map TXS error flags to BAD quality (uncertain otherwise)

3.54.2 Measurements

This stack supports measurements.

Measurement address

Type: regex

Description:

```
[!|#]DB<db-number>,<signal-number>
! - alarm on true
# - alarm on false
```

Examples:

```
DB153,A6
DB111,D1
!DB111,D1
#DB111,D1
```

3.54.3 Commands

Commands are not supported on this stack

(C) 2013-2022 OpenDAF Contributors

4. Providers

4.1 CHEMIS server

4.1.1 Parameters

Parameter	Datatype	Default Value	Description
DEBUG	enum	0	Enable debugging
DUMP	enum	0	Enable communication dump
INACTIVITY_TIMEOUT	float	30	Rx inactivity timeout [s]
HOST	hostname	0.0.0.0	Listen hostname or address (0.0.0.0 = any)
PORT	integer	25000	Listen port
CONNECTION_LIMIT	integer	0	Maximum number of simultaneous connections (0 = no limit)
ADDRESS_FALLBACK	enum	0	Use communication object name when provider address is not present

4.1.2 Measurements

This stack supports measurements.

Measurement address

Type: regex

Description:

```
Signal name to be presented to the client, will be left-padded to 7 characters by spaces
```

Examples:

```
3UA1001
4UR2003
3SS1024
```

4.1.3 Commands

Commands are not supported on this stack

(C) 2013-2022 OpenDAF Contributors

4.2 DAFLink provider

4.2.1 Parameters

Parameter	Datatype	Default Value	Description
DEBUG	enum	0	Enable debugging
DUMP	enum	0	Enable communication dump
TIMEOUT	float	20	Communication timeout [s]
HOST	hostname	0.0.0.0	Listen hostname or address (0.0.0.0 = any)
PORT	integer	2012	Listen port
RX_MSG_SIZE_LIMIT	integer	1048576	Received message size limit [B]
TX_QUEUE_SIZE_LIMIT	integer	1048576	Transmit message queue size limit [B]
ADDRESS_FALLBACK	enum	0	Use communication object name when provider address is not present

4.2.2 Measurements

This stack supports measurements.

Measurement address

Type: string

Description:

```
Ignored, uses measurement name directly
```

4.2.3 Commands

This stack supports commands.

Command address

Type: string

Description:

```
Ignored, uses command name directly
```


(C) 2013-2022 OpenDAF Contributors

4.3 IEC-60870-5-101 Controlled Station Unballanced

4.3.1 Parameters

Parameter	Datatype	Default Value	Description
DEBUG	enum	0	Enable debugging
DUMP	enum	0	Enable communication dump
TTY	tty		Communication interface
BAUDRATE	enum	115200	Baud rate
PARITY	enum	even	Parity
OCTETS_LA	int	1	# of octets of Link Address
RX_TIMEOUT	float	1.0	Receive timeout [s]
STATION_ADDRESS	int		Station address
TX_BUFFER	int	16384	Transmit buffer size
CASDU	iec-60870-5-structured-address		Default Common Address of ASDU
OCTETS_CA	int	2	# of octets of Common Address of ASDU
OCTETS_IOA	int	3	# of octets of Information Object Address
OCTETS_COT	int	2	# of octets of Cause Of Transmission
OCTET_ORDER	enum	direct	Order of CA/IOA octets

4.3.2 Measurements

This stack supports measurements.

Measurement address

Type: regex

Description:

```
[TI<TI>@][<CA0>[.<CA1>]/]<IOA0>[.<IOA1>[.<IOA2>]]
```

Examples:

```
1.2.3 - Provide as Information Object Address 1.2.3 on default CA
4.5 - Provide as Information Object Address 3.4 on default CA
30 - Provide as Information Object Address 30 on default CA
5.40/1.2.3 - Provide as Information Object Address 1.2.3 on CA 5.40
```

4.3.3 Commands

This stack supports commands.

Command address

Type: regex

(C) 2013-2022 OpenDAF Contributors

Description:

```
[<CA0>[.<CA1>]/]<IOA0>[.<IOA1>[.<IOA2>]]
```

Examples:

```
1.2.3 - Accept command from Information Object Address 1.2.3 on default CA
4.5 - Accept command from Information Object Address 3.4 on default CA
30 - Accept command from Information Object Address 30 on default CA
5.30/1.2.30 - Accept command from Information Object Address 1.2.30 on CA 5.30
```


(C) 2013-2022 OpenDAF Contributors

4.4 IEC-60870-5-104 Controlled Station

4.4.1 Parameters

Parameter	Datatype	Default Value	Description
DEBUG	enum	0	Enable debugging
DUMP	enum	0	Enable communication dump
W	int	20	Immediatelly acknowledge frames received after receiving this number of frames (parameter W)
K	int	30	Block frame transmission after reaching this number of unacknowledged frames (parameter K)
TIMEOUT	float	20	Communication timeout [s]
CASDU	iec-60870-5-structured-address		Default Common Address of ASDU
OCTETS_CA	int	2	# of octets of Common Address of ASDU
OCTETS_IOA	int	3	# of octets of Information Object Address
OCTETS_COT	int	2	# of octets of Cause Of Transmission
OCTET_ORDER	enum	direct	Order of CA/IOA octets
HOST	hostname	0.0.0.0	Listen hostname or address (0.0.0.0 = any)
PORT	int	2404	Listen port
ACT_TERM_DELAY	float	1	Delay between command ACT and TERM
CON_TIME_SYNC	enum	0	Confirm time synchronization (it won't be performed anyway)
UTC_TIMESTAMPS	enum	0	Report timestamps in UTC timezone
TIMEZONE	string		Local timezone according to *inverted* POSIX standard (keep empty to use system timezone)
TX_QUEUE_SIZE	int		Transmit queue size [B]
RX_QUEUE_SIZE	int		Receive queue size [B]

4.4.2 Measurements

This stack supports measurements.

Measurement address

Type: regex

Description:

```
[TI<TI>@][<CA0>[.<CA1>]/]<IOA0>[.<IOA1>[.<IOA2>]]
```

Examples:

```
1.2.3 - Provide as Information Object Address 1.2.3 on default CA
4.5 - Provide as Information Object Address 3.4 on default CA
```


(C) 2013-2022 OpenDAF Contributors

```
30 - Provide as Information Object Address 30 on default CA
5.40/1.2.3 - Provide as Information Object Address 1.2.3 on CA 5.40
```

4.4.3 Commands

This stack supports commands.

Command address

Type: regex

Description:

```
[<CA0>[.<CA1>]/]<IOA0>[.<IOA1>[.<IOA2>]]
```

Examples:

```
1.2.3 - Accept command from Information Object Address 1.2.3 on default CA
4.5 - Accept command from Information Object Address 3.4 on default CA
30 - Accept command from Information Object Address 30 on default CA
5.30/1.2.30 - Accept command from Information Object Address 1.2.30 on CA 5.30
```


(C) 2013-2022 OpenDAF Contributors

4.5 Kepware CID provider

4.5.1 Parameters

Parameter	Datatype	Default Value	Description
DEBUG	enum	0	Enable debugging
DEBUG_OBJECTS	string		Space-separated list of objects to debug in-depth
CAPACITY	integer	100000	Maximum memory map capacity
CFG_NAME	regex	OpenDAF	Kepware configuration name
DEVICE	string	Image	Kepware device name
MEASUREMENT_GROUP	string	Measurements	Group to place measurements in (keep empty to not create measurement group)
COMMAND_GROUP	string	Commands	Group to place commands in (keep empty to not create command group)
SCAN_PERIOD	float	0.25	Memory map scan period
ADDRESS_FALLBACK	enum	0	Use communication object name when provider address is not present
USE_UPDATE_TIMESTAMP	enum	1	Update read data timestamp in each update cycle with current timestamp

4.5.2 Measurements

This stack supports measurements.

Measurement address

Type: regex

Description:

```
Name can't contain:
- periods
- double quotation marks
- leading underscores
- leading or trailing spaces
```

Examples:

```
Tag 1
TAG__10!__20$
M_10,20
```

4.5.3 Commands

This stack supports commands.

Command address

Type: regex

Description:

```
Name can't contain:
- periods
```


(C) 2013-2022 OpenDAF Contributors

```
- double quotation marks
- leading underscores
- leading or trailing spaces
```

Examples:

```
Command 1
SETPOINT__10!__20$
SP_10,20
```


(C) 2013-2022 OpenDAF Contributors

4.6 Kruiz active provider

4.6.1 Parameters

Parameter	Datatype	Default Value	Description
DEBUG	enum	0	Enable debugging
DUMP	enum	0	Enable communication dump
HOST	hostname	localhost	Destination TCP/IP hostname or address
PORT	integer	2400	Destination TCP/IP port
FRAME_SIZE	integer	0	Manual frame size incl. header (keep 0 to autofit)
RECONNECT_DELAY	float	10	Reconnection delay after connection lost
NPP_OBJECT	integer	2	NPP object number
NPP_BLOCK	integer	3	NPP reactor block number
SERVER_TYPE	integer	0	Server type (transmitted in KRUIZ header)
SERVER_ROLE	enum	1	Server role (1 = primary, 2 = backup)
TX_PERIOD	float	1	Periodic transmit period [s]
OVERRIDE_TIMESTAMP	enum	0	Override outgoing timestamps with current timestamp

4.6.2 Measurements

This stack supports measurements.

Measurement address

Type: regex

Description:

```
<float32|int16>@<offset>
Offset 0 points to first position in frame payload, not frame header!
```

4.6.3 Commands

Commands are not supported on this stack

(C) 2013-2022 OpenDAF Contributors

4.7 Modbus/RTU Slave

4.7.1 Parameters

Parameter	Datatype	Default Value	Description
DEBUG	enum	0	Enable debugging
DUMP	enum	0	Enable communication dump
TTY	tty		Communication interface
BAUDRATE	enum	115200	Baud rate
PARITY	enum	even	Parity
ADDRESS	int	1	Station address
READ_NO_ILLEGAL_ADDRESS	enum	0	Do not return Illegal Data Address on invalid read, return zeroes instead

4.7.2 Measurements

This stack supports measurements.

Measurement address

Type: regex

Description:

```
[datatype@]<address-space>:<reference>, where:
    - datatype is used to define object layout if it spans multiple registers
        - format: <S|U|F|D><L|B><1|2|4>, where:
            - S - signed
            - U - unsigned
            - F - floating point
            - D - discrete
            - L - little-endian (LOW before HIGH)
            - B - big endian (HIGH before LOW)
            - 1 - size is 1 register/coil/discrete input
            - 2 - size is 2 registers/coils/discrete inputs
            - 4 - size is 4 registers/coils/discrete inputs
        - Unsupported datatypes: FL1, FB1, DL4, DB4
    - address-space is modbus address space number (0 - coils, 1 - discrete inputs, 3 - input registers, 4 - holding registers)
    - reference is 1-based register, coil or discrete input number in given address space
```

Examples:

```
3:1000 - measurement mapped onto input register 1000 with default datatype (16-bit unsigned)
1:550 - measurement mapped onto discrete input 550
fb4@4:1310 - measurement mapped onto 64-bit big-endian floating-point value on holding registers 1310,1311,1312,1313
dl2@0:35 - measurement mapped onto 2-bit (quaternary) in little-endian (LSB, MSB) order on coils 35,36
```

4.7.3 Commands

This stack supports commands.

Command address

Type: regex

(C) 2013-2022 OpenDAF Contributors

Description:

```
[datatype@]<address-space>:<reference>, where:
    - datatype is used to define object layout if it spans multiple registers
        - format: <S|U|F|D><L|B><1|2|4>, where:
            - S - signed
            - U - unsigned
            - F - floating point
            - D - discrete
            - L - little-endian (LOW before HIGH)
            - B - big endian (HIGH before LOW)
            - 1 - size is 1 register/coil/discrete input
            - 2 - size is 2 registers/coils/discrete inputs
            - 4 - size is 4 registers/coils/discrete inputs
        - Unsupported datatypes: FL1, FB1, DL4, DB4
    - address-space is modbus address space number (0 - coils, 4 - holding registers)
    - reference is 1-based register, coil or discrete input number in given address space
```

Examples:

```
4:1000 - command mapped onto holding register 1000 with default datatype (16-bit unsigned)
0:550 - command mapped onto coil 550
fb4@4:1310 - command mapped onto 64-bit big-endian floating-point value on holding registers 1310,1311,1312,1313
dl2@0:35 - command mapped onto 2-bit (quaternary) in little-endian (LSB, MSB) order on coils 35,36
```


(C) 2013-2022 OpenDAF Contributors

4.8 Modbus/TCP Slave

4.8.1 Parameters

Parameter	Datatype	Default Value	Description
DEBUG	enum	0	Enable debugging
DUMP	enum	0	Enable communication dump
HOST	hostname	0.0.0.0	Listen hostname or address (0.0.0.0 = any)
PORT	integer	502	Listen port
READ_NO_ILLEGAL_ADDRESS	enum	0	Do not return Illegal Data Address on invalid read, return zeroes instead

4.8.2 Measurements

This stack supports measurements.

Measurement address

Type: regex

Description:

```
[datatype@]<address-space>:<reference>, where:
    - datatype is used to define object layout if it spans multiple registers
        - format: <S|U|F|D><L|B><1|2|4>, where:
            - S - signed
            - U - unsigned
            - F - floating point
            - D - discrete
            - L - little-endian (LOW before HIGH)
            - B - big endian (HIGH before LOW)
            - 1 - size is 1 register/coil/discrete input
            - 2 - size is 2 registers/coils/discrete inputs
            - 4 - size is 4 registers/coils/discrete inputs
        - Unsupported datatypes: FL1, FB1, DL4, DB4
    - address-space is modbus address space number (0 - coils, 1 - discrete inputs, 3 - input registers, 4 - holding registers)
    - reference is 1-based register, coil or discrete input number in given address space
```

Examples:

```
3:1000 - measurement mapped onto input register 1000 with default datatype (16-bit unsigned)
1:550 - measurement mapped onto discrete input 550
fb4@4:1310 - measurement mapped onto 64-bit big-endian floating-point value on holding registers 1310,1311,1312,1313
dl2@0:35 - measurement mapped onto 2-bit (quaternary) in little-endian (LSB, MSB) order on coils 35,36
```

4.8.3 Commands

This stack supports commands.

Command address

Type: regex

Description:

```
[datatype@]<address-space>:<reference>, where:
    - datatype is used to define object layout if it spans multiple registers
        - format: <S|U|F|D><L|B><1|2|4>, where:
            - S - signed
            - U - unsigned
            - F - floating point
            - D - discrete
```


(C) 2013-2022 OpenDAF Contributors

```
            - L - little-endian (LOW before HIGH)
            - B - big endian (HIGH before LOW)
            - 1 - size is 1 register/coil/discrete input
            - 2 - size is 2 registers/coils/discrete inputs
            - 4 - size is 4 registers/coils/discrete inputs
        - Unsupported datatypes: FL1, FB1, DL4, DB4
    - address-space is modbus address space number (0 - coils, 4 - holding registers)
    - reference is 1-based register, coil or discrete input number in given address space
```

Examples:

```
4:1000 - command mapped onto holding register 1000 with default datatype (16-bit unsigned)
0:550 - command mapped onto coil 550 on slave 7
fb4@4:1310 - command mapped onto 64-bit big-endian floating-point value on holding registers 1310,1311,1312,1313
dl2@0:35 - command mapped onto 2-bit (quaternary) in little-endian (LSB, MSB) order on coils 35,36
```


(C) 2013-2022 OpenDAF Contributors

4.9 OGate Gateway Notify server

4.9.1 Parameters

Parameter	Datatype	Default Value	Description
DEBUG	enum	0	Enable debugging
DUMP	enum	0	Enable communication dump
INACTIVITY_TIMEOUT	float	30	Rx inactivity timeout [s]
HOST	hostname	0.0.0.0	Listen hostname or address (0.0.0.0 = any)
PORT	integer	60000	Listen port
ADDRESS_FALLBACK	enum	0	Use communication object name when provider address is not present
OVERRIDE_INITIAL_TIMESTAMPS	enum	0	Override initial update timestamps with current timestamp

4.9.2 Measurements

This stack supports measurements.

Measurement address

Type: regex

Description:

```
Any name except for '@@'
```

4.9.3 Commands

Commands are not supported on this stack

(C) 2013-2022 OpenDAF Contributors

4.10 OPC UA server

4.10.1 Parameters

Parameter	Datatype	Default Value	Description
DEBUG	enum	0	Enable debugging
ADDRESS_FALLBACK	enum	1	Use communication object name when provider address is not present
JOIN_OBJECTS	enum	1	Join measurements and commands with the same provider address into one item
PORT	integer	4840	TCP/IP listen port
NAMESPACE	string		Custom namespace (keep empty to use default ns 1)

4.10.2 Measurements

This stack supports measurements.

Measurement address

Type: regex

Description:

```
OPC-UA item name with optional type prefix. Folders are created using dot character.
```

Examples:

```
Tag1
boolean@machine1.di.di3
double@3UA0010
```

4.10.3 Commands

This stack supports commands.

Command address

Type: regex

Description:

```
OPC-UA item name with optional type prefix. Folders are created using dot character.
```

Examples:

```
Tag1
boolean@machine1.di.di3
double@3UA0010
```


(C) 2013-2022 OpenDAF Contributors

4.11 SCORPIO server

4.11.1 Parameters

Parameter	Datatype	Default Value	Description
DEBUG	enum	0	Enable debugging
DUMP	enum	0	Enable communication dump
INACTIVITY_TIMEOUT	float	30	Rx inactivity timeout [s]
HOST	hostname	0.0.0.0	Listen hostname or address (0.0.0.0 = any)
PORT	integer	12102	Listen port
CONNECTION_LIMIT	integer	0	Maximum number of simultaneous connections (0 = no limit)
TX_PERIOD	float	4.0	Transmission period [s]
OVERRIDE_TIMESTAMPS	enum	0	Override timestamps with current

4.11.2 Measurements

This stack supports measurements.

Measurement address

Type: regex

Description:

```
Signal name to be presented to the SCORPIO
```

Examples:

```
A1001
D0200
Z0035
```

4.11.3 Commands

Commands are not supported on this stack

(C) 2013-2022 OpenDAF Contributors

4.12 Sigfox

4.12.1 Parameters

Parameter	Datatype	Default Value	Description
DEBUG	enum	0	Enable debugging
DUMP	enum	0	Enable communication dump
HOST	hostname	localhost	Destination TCP/IP hostname or address
PORT	integer	2400	Destination TCP/IP port
RECONNECT_DELAY	float	10	Reconnection delay after connection lost

4.12.2 Measurements

This stack supports measurements.

Measurement address

Type: enum

4.12.3 Commands

Commands are not supported on this stack

(C) 2013-2022 OpenDAF Contributors

4.13 Communication object Per Row SQL provider

4.13.1 Parameters

Parameter	Datatype	Default Value	Description
DEBUG	enum	0	Enable debugging
SAMPLING_PERIOD	float	5	Sampling period [s] (when using master/backup configuration, don't set equal sampling period for both master and slave)
CONNECTION_STRING	regex		Database connection string
TABLE	regex		Database table name
COL_ID_NAME	regex		Name of column holding measurement identification (column to be matched against measurement provider address)
COL_ID_VALUE	regex		Name of column holding measurement value (leave empty to skip)
COL_ID_TIMESTAMP	regex		Name of column holding measurement timestamp (leave empty to skip)
COL_ID_QUALITY	regex		Name of column holding measurement quality (leave empty to skip)
UTC	enum		Write timestamps in UTC timezone
LOCK	string	ODBCLock	Session lock (consider using ODBCLock when connecting to ODBC datasource)

4.13.2 Measurements

This stack supports measurements.

Measurement address

Type: string

Description:

```
Row identifier to lookup in identification column
```

4.13.3 Commands

Commands are not supported on this stack

(C) 2013-2022 OpenDAF Contributors

4.14 EBO dosimetry SQL provider

4.14.1 Parameters

Parameter	Datatype	Default Value	Description
DEBUG	enum	0	Enable debugging
SAMPLING_PERIOD	float	5	Sampling period [s] (when using master/backup configuration, don't set equal sampling period for both master and slave)
CONNECTION_STRING	regex		Database connection string
TABLE	regex		Database table name
COLUMN_NAME_PRECISION	integer	2	Column name precision (1 -> V1, 2 -> V01, 3 -> V001)
ROLE	enum	master	Provider role
LOCK	string	ODBCLock	Session lock (consider using ODBCLock when connecting to ODBC datasource)

4.14.2 Measurements

This stack supports measurements.

Measurement address

Type: integer

Description:

```
Integer measurement index
```

4.14.3 Commands

Commands are not supported on this stack

(C) 2013-2022 OpenDAF Contributors

4.15 SuHub Notify server

4.15.1 Parameters

Parameter	Datatype	Default Value	Description
DEBUG	enum	0	Enable debugging
DUMP	enum	0	Enable communication dump
INACTIVITY_TIMEOUT	float	30	Rx inactivity timeout [s]
HOST	hostname	0.0.0.0	Listen hostname or address (0.0.0.0 = any)
PORT	integer	61000	Listen port
ADDRESS_FALLBACK	enum	0	Use communication object name when provider address is not present
OVERRIDE_INITIAL_TIMESTAMPS	enum	0	Override initial update timestamps with current timestamp

4.15.2 Measurements

This stack supports measurements.

Measurement address

Type: string

Description:

```
Any name
```

4.15.3 Commands

Commands are not supported on this stack

(C) 2013-2022 OpenDAF Contributors

4.16 SuHub Periodic Notify server

4.16.1 Parameters

Parameter	Datatype	Default Value	Description
DEBUG	enum	0	Enable debugging
DUMP	enum	0	Enable communication dump
INACTIVITY_TIMEOUT	float	30	Rx inactivity timeout [s]
HOST	hostname	0.0.0.0	Listen hostname or address (0.0.0.0 = any)
PORT	integer	60006	Listen port
ADDRESS_FALLBACK	enum	0	Use communication object name when provider address is not present

4.16.2 Measurements

This stack supports measurements.

Measurement address

Type: regex

Description:

```
Any name except for '@@'
```

4.16.3 Commands

Commands are not supported on this stack

(C) 2013-2022 OpenDAF Contributors

5. Network integration APIs

5.1 RESTful API version 1

This HTTP-based API is arguably the most easily usable among available APIs. It's designed according to RESTful design principle.

Usually it's accessible at http://server/opendaf/ or https://server/opendaf/.

5.1.1 Resource and method summary

Resource	GET	PUT	POST	DELETE
`/measurements/`	get array of VTQ+			
`/measurements/:name`	get VTQ and more	start simulation		stop simulation
`/commands/`	get array of VT+			
`/commands/:name`	get VT and more	write value		
`/alarms/`	get array of alms			
`/alarms/:name`	get alarm			
`/alarms/:name/activate`			activate alarm	
`/alarms/:name/deactivate`			deactivate alarm	
`/alarms/:name/acknowledge`			acknowledge alarm	
`/archive/measurements/:name/:t_since/:t_until`	read meas.archive			
`/archive/commands/:name/:t_since/:t_until`	read cmd. archive			
`/archive/alarms/:name/active/:t_since/:t_until`	read alm. archive			

5.1.2 Resource details

This chapter covers details of individual resources and methods.

Path parameters

Name	Data type	Description
:name	identifier	resource name
:t_since	number	time interval start as number of seconds since unix epoch
:t_until	number	time interval end as number of seconds since unix epoch

(C) 2013-2022 OpenDAF Contributors

Query parameters

When parameters are referenced in subsequent chapters, required ones are marked with asterisk(*).

Name	Data type	Description
names	comma-separated list of identifiers	Limit operation to names listed
fields	comma-separated list of strings	Limit result JSON to fields listed
value	prefixed-value	Supply value to operation
timestamp	number	Timestamp as floating-point seconds since unix epoch
quality	number	Quality flags according to OPC-DA 2.05 standard
valid-for	number	Operation validity in seconds
resample	number	Resample to given sampling rate in seconds
format	string	Possible values: `json`, `csv`, `bin-v1`
warp_head	boolean	Generate synthetic sample at *:t_since*
q_at_least	string	Possible values: `good`, `uncertain`, `bad`

Measurements

`GET /measurements/`

Get properties of measurements. Result schema is measurements.json.

- query parameters: *names, fields*
- available fields: *name, vtq, address, datatype, raw-datatype, eu-range, raw-range, description*

Measurement

`GET /measurements/:name`

Get properties of the measurement. Result schema is measurement.json.

`PUT /measurements/:name`

Start simulation on the measurement.

- query parameters: *value(*), timestamp, quality, valid-for*

`DELETE /measurements/:name`

Stop simulation on the measurement.

`POST /measurements/:name`

Overwrite measurement value, timestamp, quality.

- query parameters: *value(*), timestamp, quality*

Commands

`GET /commands/`

Get properties of commands. Result schema is commands.json.

- query parameters: *names, fields*
- available fields: *name, vt, address, datatype, raw-datatype, eu-range, raw-range, description*

(C) 2013-2022 OpenDAF Contributors

Command

`GET /command/:name`

Get properties of the command. Result schema is command.json.

`PUT /commands/:name`

Write new command value.

- query parameters: *value(*)*

Alarms

`GET /alarms/`

Get properties of alarms. Result schema is alarms.json.

- query parameters: *names, fields*
- available fields: *name, severity, t, authority, state, description*

Alarm

`GET /alarms/:name`

Get properties of the alarm. Result schema is alarm.json.

`POST /alarms/:name/activate`

Activate the alarm.

`POST /alarms/:name/deactivate`

Deactivate the alarm.

`POST /alarms/:name/acknowledge`

Measurement archive

`GET /archive/measurements/:name/:t_since/:t_until`

Get records from archive for measurement *:name* since *:t_since* until *:t_until*.

- query parameters: *resample, format, warp_head, q_at_least*

Result schema is results-measurement.

Command archive

`GET /archive/commands/:name/:t_since/:t_until`

Get records from archive for command *:name* since *:t_since* until *:t_until*.

- query parameters: *resample, format, warp_head*

Result schema is results-command.

`GET /archive/alarms/:name/:t_since/:t_until`

Get records from archive for alarm *:name* since *:t_since* until *:t_until*.

- query parameters: *format*

Result schema is results-alarm.

(C) 2013-2022 OpenDAF Contributors

5.1.3 JSON schemas

Download schemas archive here

Top-level document schemas

COMMUNICATION_OBJECT.JSON

This is not true top-level document, but is a base schema for measurement.json and command.json.

```
{
    "$schema": "http://json-schema.org/draft-07/schema#",
    "type": "object",
    "properties": {
        "name": {
            "title": "Unique name",
            "$ref": "identifier.json"
        },
        "address": {
            "title": "Connector address",
            "type": "string"
        },
        "datatype": {
            "title": "Datatype",
            "$ref": "datatype.json"
        },
        "raw-datatype": {
            "title": "Raw datatype",
            "$ref": "datatype.json"
        },
        "eu-range": {
            "title": "Engineering range",
            "$ref": "range.json"
        },
        "raw-range": {
            "title": "Raw range",
            "$ref": "range.json"
        },
        "eu": {
            "title": "Engineering units",
            "type": "string"
        },
        "description": {
            "title": "Description",
            "type": "string"
        }
    },
    "required": [
        "name",
        "address",
        "datatype",
        "raw-datatype",
        "eu-range",
        "raw-range"
    ]
}
```

MEASUREMENT.JSON

```
{
    "$schema": "http://json-schema.org/draft-07/schema#",
    "type": "object",
    "allOf": [
        { "$ref": "communication_object.json" },
        {
            "type": "object",
            "properties": {
                "vtq": { "$ref": "vtq.json" }
            },
            "required": [ "vtq" ]
        }
    ]
}
```

MEASUREMENTS.JSON

```
{
    "$schema": "http://json-schema.org/draft-07/schema#",
    "title":    "Map of measurement (name -> measurement)",
    "type":     "object",
    "patternProperties": {
        "[_a-zA-Z][_a-zA-Z0-9]": { "$ref": "measurement.json" }
    }
}
```


(C) 2013-2022 OpenDAF Contributors

COMMAND.JSON

```
{
    "$schema": "http://json-schema.org/draft-07/schema#",
    "type": "object",
    "allOf": [
        { "$ref": "communication_object.json" },
        {
            "type": "object",
            "properties": {
                "vt": { "$ref": "vt.json" }
            },
            "required": [ "vt" ]
        }
    ]
}
```

COMMANDS.JSON

```
{
    "$schema": "http://json-schema.org/draft-07/schema#",
    "title":    "Map of commands (name -> command)",
    "type":     "object",
    "patternProperties": {
        "[_a-zA-Z][_a-zA-Z0-9]": { "$ref": "command.json" }
    }
}
```

Top-level archive document schemas

RESULTS-MEASUREMENT.JSON

```
{
    "title":    "archive query measurement history results",
    "type":     "object",
    "patternProperties": {
        "[_a-zA-Z][_a-zA-Z0-9]": {
            "type": "array",
            "items": {
                "type": "array",
                "items": [
                    {
                        "title": "value with type prefix",
                        "type": "string"
                    },
                    {
                        "title": "timestamp",
                        "type": "array",
                        "items": [
                            {
                                "title": "seconds since the epoch",
                                "type": "integer",
                                "minimum": 0
                            },
                            {
                                "title": "microseconds",
                                "type": "integer",
                                "minimum": 0,
                                "maximum": 999999
                            }
                        ]
                    },
                    {
                        "title": "quality according to OPC DA 2.05",
                        "type": "integer",
                        "minimum": 0,
                        "maximum": 255
                    }
                ],
                "additionalItems": false
            }
        }
    }
}
```


(C) 2013-2022 OpenDAF Contributors

RESULTS-COMMAND.JSON

```
{
    "title":    "archive query command history results",
    "type":     "object",
    "patternProperties": {
        "[_a-zA-Z][_a-zA-Z0-9]": {
            "type": "array",
            "items": {
                "type": "array",
                "items": [
                    {
                        "title": "value with type prefix, or null",
                        "type": ["string", "null"]
                    },
                    {
                        "title": "timestamp",
                        "type": "array",
                        "items": [
                            {
                                "title": "seconds since the epoch",
                                "type": "integer",
                                "minimum": 0
                            },
                            {
                                "title": "microseconds",
                                "type": "integer",
                                "minimum": 0,
                                "maximum": 999999
                            }
                        ]
                    }
                ],
                "additionalItems": false
            }
        }
    }
}
```

RESULTS-ALARM.JSON

```
{
    "title":    "archive query alarm history results",
    "type":     "object",
    "patternProperties": {
        "[_a-zA-Z][_a-zA-Z0-9]": {
            "type": "array",
            "items": {
                "type": "object",
                "properties": {
                    "t": {
                        "title": "timestamp",
                        "type": "array",
                        "items": [
                            {
                                "title": "seconds since the epoch",
                                "type": "integer",
                                "minimum": 0
                            },
                            {
                                "title": "microseconds",
                                "type": "integer",
                                "minimum": 0,
                                "maximum": 999999
                            }
                        ]
                    },
                    "state": { "enum": ["INACT_ACK", "ACT_UNACK", "ACT_ACK", "INACT_UNACK"] },
                    "description": { "type": "string" },
                    "authority": { "type": "string" }
                },
                "additionalProperties": false,
                "required": [ "t", "state", "description", "authority" ]
            }
        }
    }
}
```

Basic type schemas

DATATYPE.JSON

```
{
    "$schema": "http://json-schema.org/draft-07/schema#",
```


(C) 2013-2022 OpenDAF Contributors

```
    "enum": [ "b", "q", "i", "l", "f", "d", "s" ]
}
```

IDENTIFIER.JSON

```
{
    "$schema": "http://json-schema.org/draft-07/schema#",
    "pattern": "^[0-9_a-zA-Z]{1,64}$"
}
```

PREFIXED_VALUE.JSON

String starting with datatype character, followed by formatted value.

```
{
    "$schema": "http://json-schema.org/draft-07/schema#",
    "anyOf": [
        { "pattern": "^b[01]$" },
        { "pattern": "^q[0123]$" },
        { "pattern": "^[il][+-]?[0-9]+$" },
        { "pattern": "^[fd][+-]?[0-9]*\\.?[0-9]+$" },
        { "pattern": "^s" }
    ]
}
```

QUALITY.JSON

Number representing measurement quality according to OPC-DA 2.05 standard.

```
{
    "$schema": "http://json-schema.org/draft-07/schema#",
    "title": "Measurement quality according to OPC DA 2.05",
    "type": "integer",
    "minimum": 0,
    "maximum": 65535,
    "description": "good: (quality & 0xC0) == 0xC0, uncertain: (quality & 0xC0) == 0x40, bad: (quality & 0xC0) == 0x00"
}
```

TIMESTAMP.JSON

2-element array containing number of seconds and microseconds since the unix epoch (1970-01-01 01:00 UTC).

```
{
    "$schema": "http://json-schema.org/draft-07/schema#",
    "title": "timestamp",
    "type": "array",
    "items": [
        {
            "title": "seconds since the epoch",
            "type": "integer",
            "minimum": 0
        },
        {
            "title": "microseconds",
            "type": "integer",
            "minimum": 0,
            "maximum": 999999
        }
    ]
}
```

RANGE.JSON

2-element array `[low, high]` holding range of some value.

```
{
    "$schema": "http://json-schema.org/draft-07/schema#",
    "type": "array",
    "items": {
        "anyOf" : [
            { "$ref": "prefixed_value.json" },
            { "type": "null" }
        ]
    },
    "minItems": 2,
```


(C) 2013-2022 OpenDAF Contributors

```
    "maxItems": 2
}
```

VT.JSON

2-element array `[value,timestamp]` holding command value and timestamp.

```
{
    "$schema": "http://json-schema.org/draft-07/schema#",
    "title": "value-timestamp tuple",
    "type": "array",
    "items": [
        { "$ref": "prefixed_value.json" },
        { "$ref": "timestamp.json" }
    ],
    "additionalItems": false
}
```

VTQ.JSON

3-element array `[value,timestamp,quality]` holding measurement value, timestamp and quality.

```
{
    "$schema": "http://json-schema.org/draft-07/schema#",
    "title": "value-timestamp-quality tuple",
    "type": "array",
    "items": [
        { "$ref": "prefixed_value.json" },
        { "$ref": "timestamp.json" },
        { "$ref": "quality.json" }
    ],
    "additionalItems": false
}
```

SEVERITY.JSON

Numeric alarm severity value.

```
{
    "$schema": "http://json-schema.org/draft-07/schema#",
    "title": "Alarm severity",
    "type": "integer",
    "minimum": 0,
    "description": "Alarm with severity 0 is the most severe. Severity decreased with its value rising."
}
```


(C) 2013-2022 OpenDAF Contributors

5.2 WebSocket API version 1

WebSocket API is designed to allow real-time streaming of events happening in OpenDAF runtime.

Usually it's accessible at http://server/opendaf/ws/opendaf or https://server/opendaf/ws/opendaf.

5.2.1 Overview

WS-API communication protocol is mostly event based. Client has to setup watches on objects of interest - measurements, commands, alarms, function modules. Once the watch is setup, WS-API server will transmit current object state, and begin transmitting state change notifications. When the client is no longer interested in given object, it can be unwatched and server will stop transmitting state changes.

5.2.2 Requests

Requests are JSON objects issued by the client. Each request contains *request_id* assigned by the client. Server executes the request and returns a response.

See request / response schema files `request.json` and `response.json` here

Requests summary

Request (`request.json`)	**Response (`response.json`)**	**Description**
#/types/**Heartbeat**	#/types/**Response**	Perform connection test
#/types/**WatchMeasurements**	#/types/**Response**	Cancel measurements watch
#/types/**UnwatchMeasurements**	#/types/**Response**	Setup measurements watch
#/types/**WatchCommands**	#/types/**Response**	Setup commands watch
#/types/**UnwatchCommands**	#/types/**Response**	Cancel commands watch
#/types/**WatchFunctionModules**	#/types/**Response**	Setup function modules watch
#/types/**UnwatchFunctionModules**	#/types/**Response**	Cancel function modules watch
#/types/**WatchAlarms**	#/types/**Response**	Setup alarm state change watch
#/types/**UnwatchAlarms**	#/types/**Response**	Cancel alarm state change watch
#/types/**AckAlarms**	#/types/**Response**	Acknowledge alarms
#/types/**OperateAlarms**	#/types/**Response**	Perform bulk operations on alarms
#/types/**WriteCommands**	#/types/**Response**	Perform command write transaction
#/types/**ReadDatabase**	#/types/**DatabaseResponse**	Read configuration database

WatchMeasurements / UnwatchMeasurements

WatchMeasuremenets request sets up watch on measurements listed. When measurement value, timestamp or quality changes, client is notified with *MeasurementUpdateNotification*. Initial notification is transmitted by server right after receiving the request.

UnwatchMeasurements cancels watch on measurements listed.

(C) 2013-2022 OpenDAF Contributors

WatchCommands / UnwatchCommands

WatchCommands request sets up watch on commands listed. When command value is written, client is notified with *CommandUpdateNotification*. Initial notification is transmitted by server right after receiving the request.

UnwatchCommands cancels watch on commands listed.

WatchFunctionModules / UnwatchFunctionModules

WatchFunctionModules request sets up watch on function modules listed. When function module state changes, client is notified with *FunctionModuleUpdateNotification*. Initial notification is transmitted by server right after receiving the request.

UnwatchFunctionModules cancels watch on function modules listed.

WatchAlarms / UnwatchAlarms

WatchAlarms request sets up watch on alarms listed. When alarm state changes, client is notified with *AlarmStateChangeNotification*. Initial notification is transmitted by server right after receiving the request.

UnwatchAlarms cancels watch on alarms listed.

AckAlarms

AckAlarms acknowledges alarms listed. Optional `timestamp` (`[seconds, microseconds]`) sets the time of the operation; server's current time is used when omitted.

OperateAlarms

OperateAlarms performs bulk activation / deactivations / acknowledge on alarms listed. Optional `timestamp` (`[seconds, microseconds]`) sets the time of the operations; server's current time is used when omitted.

WriteCommands

WriteCommands performs command write transaction.

ReadDatabase

Server responds to *ReadDatabase* request with *DatabaseResponse* containing definitions of objects requested.

5.2.3 Sollicited response

Solicited response (response to request) is a JSON object always containing: - integer property *request_id* mirrored from the originating request - boolean property *result*, true = request succeeded, false = request failed

See schema file `response.json` here

5.2.4 Unsolicited notification

Unsolicited (spontaneous) notifications don't carry *request_id* property. Server transmits them when watched event occurs.

See schema file `response.json` here

(C) 2013-2022 OpenDAF Contributors

Notification summary

Notification (`response.json`)	Transmission trigger
#/types/**MeasurementUpdateNotification**	Measurement value/timestamp/quality change
#/types/**CommandUpdateNotification**	Command value/timestamp change
#/types/**FunctionModuleUpdateNotification**	Function module state change
#/types/**AlarmStateChangeNotification**	Alarm state change

(C) 2013-2022 OpenDAF Contributors

5.3 Transaction Ser/Des

5.3.1 1. Introduction

This document contains description of raw transaction protocol as provided by transaction-serdes modules. With default configuration, transaction-serdes is listening on 127.0.0.1:6378/tcp.

As this module provides unlimited access to OpenDAF transaction mechanism, it's imperative not to expose this interface to the world, i.e. transaction-serdes shall not listen on 0.0.0.0:6378/tcp. External clients shall access this interface via ssh tunnel. User credentials are of course required for such access.

5.3.2 2. Protocol description

Protocol is a symmetric, very native, low-level, binary, and its description will be similar :-) It's purpose is a high-speed transaction streaming between OpenDAF components. Symmetric means client and server both use the same stream structure and can both transmit and receive transactions.

There is no padding / alignment in serialized data structures.

2.1 Notations used

notation	description
`{ type }`	describes serialized layout of given structure type
`[ type ]`	represents one instance of serialized structure of given type
`type name`	represents discrete field of given data type named "name" (discrete data types have serialized layout identical to in-memory layout)
`[N * type]`	represents N instances of discrete type or serialized structures type
`<type1|type2|...|typeN>`	represents any of discrete or serialized structure types mentioned
`//`	line comment
`/* */`	block comment

2.2 Datatypes used

datatype	description
string	variable-length null-terminated string
uint16_t, uint32_t	standard C types
int	signed integer type, 32-bit on both 32-bit and 64-bit platforms
float	IEEE-754 32-bit float
double	IEEE-754 64-bit float

2.3 Protocol stream structure

Protocol stream carries only `[ TransactionContainer ]` entities.

```
{ TransactionContainer }
    - int transaction_type          // 0x0001 - SetTransactionMaskTransaction
                                    // 0x0100 - MeasurementUpdateTransaction
                                    // 0x0200 - CommandWriteTransaction
                                    // 0x0400 - InitialMeasurementUpdateTransaction
                                    // 0x0800 - InitialCommandWriteTransaction
                                    // 0x1000 - InitialAlarmStateChangeTransaction
```


(C) 2013-2022 OpenDAF Contributors

```
                                // 0x2000 - AlarmStateChangeTransaction
    - uint32_t transaction_size     // size of transaction field following
    - <[SetTransactionMaskTransaction]|[MeasurementUpdateTransaction]|[CommandWriteTransaction]|AlarmStateChangeTransaction]> transaction
```

2.4 Transaction layouts

```
{ SetTransactionMaskTransaction }
    - uint64_t mask         // bitmask of transactions peer should start transmitting to us

{ MeasurementUpdateTransaction }
    - uint32_t n_updates    // # of updates following
    - [ n_updates * [ MeasurementUpdate ] ] updates

{ InitialMeasurementUpdateTransaction }
    - [ MeasurementUpdateTransaction] initial_state

{ CommandWriteTransaction }
    - uint32_t n_writes     // # of writes following
    - [ n_writes * [ CommandWrite] ] writes

{ InitialCommandWriteTransaction }
    - [ CommandWriteTransaction ] initial_state

{ AlarmStateChangeTransaction }
    - uint32_t n_changes    // # of changes following
    - [ Timestamp ] timestamp
    - string authority
    - [ n_changes * [ AlarmStateChange ] ]
```

2.5 Auxiliary structures layout

```
{ InitialAlarmStateChangeTransaction }
    - [ AlarmStateChangeTransaction ] initial_state

{ MeasurementUpdate }
    - string measurement_name
    - [ VTQ ] vtq

{ CommandWrite }
    - string command_name
    - [ VT ] vt

{ AlarmStateChange }
    - string alarm_name
    - int state         // 0 - INACT_ACK, 1 - ACT_UNACK, 2 - ACT_ACK, 3 - INACT_UNACK

{ VTQ }
    - [ VT ] vt
    - uint16_t quality      // quality & 0x00C0 => C0 = good, 00 = bad, 80 = uncertain, 40 = unused

{ VT }
    - [ Value ] value
    - [ T ] timestamp

{ T }
    - uint64_t value    // [] = us

{ Value }
    - uint8_t datatype                                  // 0 - empty, 1 - integer, 2 - long, 3 - float, 4 - double, 5 - binary, 6 - quaternary, 7 - string
    - <int|int64_t|float|double|uint8_t|string> value   // non-trivial datatype mapping: empty -> nothing, binary -> uint8_t, quaternary -> uint8_t
```

5.3.3 3. Usage scenarios

3.1 Realtime data acquisition

This scenario assumes clients wants to read all data from OpenDAF in realtime.

Sequence of events: 1. client connects to transaction serdes 2. client transmits SetTransactionMaskTransaction with either mask = 0x0100 or mask = 0x0500 3. server transmits SetTransactionMaskTransaction too, client can safely skip it 4. server transmits InitialMeasurementUpdateTransaction (if 0x0400 is present in mask) 5. servers continuously transmits MeasurementUpdateTransactions

(C) 2013-2022 OpenDAF Contributors

5.4 DAFMan RESTful API - version 2

This HTTP-based API provides access to DAFMan's database actions. Usually it's accessible at http://server/dafman/v2/ or https://server/dafman/v2/.

(C) 2013-2022 OpenDAF Contributors

5.4.1 Resource and method summary

(C) 2013-2022 OpenDAF Contributors

Resource	GET	PUT	POST	DELETE
`/alarms/`	get List of Alarms			
`/alarms/names`	get List of Alarms' names			
`/alarms/:name`	get Alarm	Create / update Alarm		Delete Alarm
`/commands/`	get List of Commands			
`/commands/names`	get List of Commands' names			
`/commands/:name`	get Command	Create / update Command		Delete Command
`/connectors/`	get List of Connectors			
`/connectors/names`	get List of Connectors' names			
`/connectors/:name`	get Connector	Create / update Connector		Delete Connector
`/cfgdb/database`	get dafman database		Replace database	
`/cfgdb/render`	get render status		Render config	
`/cfgfb/txn`			Perform ops in one transaction	
`/function-modules/`	get List of Function modules			
`/function-modules/names`	get List of Function modules' names			
`/function-modules/:name`	get Function module	Create / update Function module		Delete Alarm
`/function-modules/:name/executable`	get Function module executable	Replace executable		Delete executable
`/function-modules/:name/executable-stat`	get Function module executable stat			
`/measurements/`	get List of Measurements			
`/measurements/names`	get List of Measurements' names			
`/measurements/:name`	get Measurement	Create / update Measurement		Delete Measurement
`/providers/`	get List of Providers			
`/providers/names`	get List of Providers' names			
`/providers/:name`	get Provider	Create / update Provider		Delete Provider

(C) 2013-2022 OpenDAF Contributors

Resource	GET	PUT	POST	DELETE
`/stacks/connector/`	get List of Connectors' Stacks			
`/stacks/connector/names`	get List of Connectors' Stack names			
`/stacks/connector/:name`	get Connector Stack			
`/stacks/provider/`	get List of Providers' Stacks			
`/stacks/provider/names`	get List of Providers' Stacks-names			
`/stacks/provider/:name`	get Provider Stack			

5.4.2 Resource details

This chapter covers details of individual resources and methods.

Path parameters

Name	Data type	Description
:name	identifier	Object / resource name (unique identificator)

Query parameters

When parameters are referenced in subsequent chapters, required ones marked with asterisk(*).

Name	Data type	Description
names	comma-separated list of identifiers	Limit operation to names listed
force	boolean	Force operation
commit	boolean	Commit(1) or Rollback(0) transaction
render	boolean	Render database on success

Alarms

`GET /alarms/names`

Get names of all alarms as `Array<String>`.

Status Code	Response	Description
200	`Array<String>`	Success
500	Error	Cannot get list of Alarms' names

`GET /alarms/`

(C) 2013-2022 OpenDAF Contributors

Get properties of alarms as `Array<Object>`. Result schema is alarms.json.

- query parameters: *names*

Status Code	Response	Description
200	`Array<Object>`	Success
400	Bad Request	Query parameters invalid
500	Error	Cannot get list of Alarms

Alarm

`GET /alarms/:name`

Get properties of the alarm as `Object`. Result schema is alarm.json.

Status Code	Response	Description
200	`Object`	Success
404	Not Found	Alarm does not exists
500	Error	Cannot get Alarm

`PUT /alarms/:name`

Create or update the alarm configuration. Request schema is alarm.json.

Status Code	Response	Description
204		Success
400	Bad Request	JSON invalid
500	Error	Cannot create / update Alarm

`DELETE /alarms/:name`

Delete the alarm configuration from database.

Status Code	Response	Description
204		Success
404	Not Found	Alarm does not exists
500	Error	Cannot delete Alarm

Commands

`GET /commands/names`

Get names of all commands as `Array<String>`.

Status Code	Response	Description
200	`Array<String>`	Success
500	Error	Cannot get list of Commands' names

`GET /commands/`

(C) 2013-2022 OpenDAF Contributors

Get properties of commands as `Array<Object>`. Result schema is commands.json.

- query parameters: *names*

Status Code	Response	Description
200	`Array<Object>`	Success
400	Bad Request	Query parameters invalid
500	Error	Cannot get list of Commands

Command

`GET /commands/:name`

Get properties of the command as `Object`. Result schema is command.json.

Status Code	Response	Description
200	`Object`	Success
404	Not Found	Command does not exists
500	Error	Cannot get Command

`PUT /commands/:name`

Create or update the command configuration. Request schema is command.json.

Status Code	Response	Description
204		Success
400	Bad Request	JSON invalid
500	Error	Cannot create / update Command

`DELETE /commands/:name`

Delete the command configuration from database.

Status Code	Response	Description
204		Success
404	Not Found	Command does not exists
500	Error	Cannot delete Command

Database

`GET /cfgdb/database`

Get dafman sqlite3 database as `Content-type: application/x-sqlite3`.

Status Code	Response	Description
200	`dafman.sqlite3`	Success
500	Error	Cannot get database

`POST /cfgdb/database`

(C) 2013-2022 OpenDAF Contributors

Replace configuration database. Request file must be a valid `.sqlite3` file and headers should be set to `Content-type: application/x-sqlite3`

Status Code	Response	Description
204		Success
500	Error	Replace failed

`POST /cfgdb/txn`

Perform operations defined in request body. Request body schema is txn.json

- query parameters: *commit, render*

Status Code	Response	Description
204		Operations performed
400	Error	Request does not conform to schema
500	Error	Operation failed due to server error

Measurements

`GET /measurements/names`

Get names of all measurements as `Array<String>`.

Status Code	Response	Description
200	`Array<String>`	Success
500	Error	Cannot get list of Measurements' names

`GET /measurements/`

Get properties of measurements as `Array<Object>`. Result schema is measurements.json.

- query parameters: *names*

Status Code	Response	Description
200	`Array<Object>`	Success
400	Bad Request	Query parameters invalid
500	Error	Cannot get list of Measurements

Measurement

`GET /measurements/:name`

Get properties of the Measurement as `Object`. Result schema is measurement.json.

Status Code	Response	Description
200	`Object`	Success
404	Not Found	Measurement does not exists
500	Error	Cannot get Measurement

`PUT /measurements/:name`

(C) 2013-2022 OpenDAF Contributors

Create or update the Measurement configuration. Request schema is measurement.json.

Status Code	Response	Description
204		Success
400	Bad Request	JSON invalid
500	Error	Cannot create / update Measurement

`DELETE /measurements/:name`

Delete the Measurement configuration from database.

Status Code	Response	Description
204		Success
404	Not Found	Measurement does not exists
500	Error	Cannot delete Measurement
### Render		

`GET /cfgdb/render`

Get configuration render status. Result schema is render.json.

Status Code	Response	Description
200	`Object`	Success
500	Error	Cannot get render status

`POST /cfgdb/render`

Render configuration database.

- query parameters: *force* - Force configuration render - render even when configuration is up to date.

Status Code	Response	Description
204		Success
500	Error	Render failed

Providers

`GET /providers/names`

Get names of all providers as `Array<String>`.

Status Code	Response	Description
200	`Array<String>`	Success
500	Error	Cannot get list of Providers' names

`GET /providers/`

(C) 2013-2022 OpenDAF Contributors

Get properties of providers as `Array<Object>`. Result schema is stack_instantiations.json.

- query parameters: *names*

Status Code	Response	Description
200	`Array<Object>`	Success
400	Bad Request	Query parameters invalid
500	Error	Cannot get list of Providers

Provider

`GET /providers/:name`

Get properties of the Provider as `Object`. Result schema is stack_instantiation.json.

Status Code	Response	Description
200	`Object`	Success
404	Not Found	Provider does not exists
500	Error	Cannot get Provider

`PUT /providers/:name`

Create or update the Provider configuration. Request schema is stack_instantiation.json.

Status Code	Response	Description
204		Success
400	Bad Request	JSON invalid
500	Error	Cannot create / update Provider

`DELETE /providers/:name`

Delete the Provider configuration from database.

Status Code	Response	Description
204		Success
404	Not Found	Provider does not exists
500	Error	Cannot delete Provider

Connectors

`GET /connectors/names`

Get names of all connectors as `Array<String>`.

Status Code	Response	Description
200	`Array<String>`	Success
500	Error	Cannot get list of Connectors' names

`GET /connectors/`

(C) 2013-2022 OpenDAF Contributors

Get properties of connectors as `Array<Object>`. Result schema is stack_instantiations.json.

- query parameters: *names*

Status Code	Response	Description
200	`Array<Object>`	Success
400	Bad Request	Query parameters invalid
500	Error	Cannot get list of Connectors

Connector

`GET /connectors/:name`

Get properties of the Connector as `Object`. Result schema is stack_instantiation.json.

Status Code	Response	Description
200	`Object`	Success
404	Not Found	Connector does not exists
500	Error	Cannot get Connector

`PUT /connectors/:name`

Create or update the Connector configuration. Request schema is stack_instantiation.json.

Status Code	Response	Description
204		Success
400	Bad Request	JSON invalid
500	Error	Cannot create / update Connector

`DELETE /connectors/:name`

Delete the Connector configuration from database.

Status Code	Response	Description
204		Success
404	Not Found	Connector does not exists
500	Error	Cannot delete Connector

Function modules

`GET /function-modules/names`

Get names of all function modules as `Array<String>`.

Status Code	Response	Description
200	`Array<String>`	Success
500	Error	Cannot get list of Function modules' names

`GET /function-modules/`

(C) 2013-2022 OpenDAF Contributors

Get properties of function modules as `Array<Object>`. Result schema is function_modules.json.

- query parameters: *names*

Status Code	Response	Description
200	`Array<Object>`	Success
400	Bad Request	Query parameters invalid
500	Error	Cannot get list of Function modules

Function module

`GET /function-modules/:name`

Get properties of the Function module as `Object`. Result schema is function_modules.json.

Status Code	Response	Description
200	`Object`	Success
404	Not Found	Function module does not exists
500	Error	Cannot get Function module

`PUT /function-modules/:name`

Create or update the Function module configuration. Request schema is function_modules.json.

Status Code	Response	Description
204		Success
400	Bad Request	JSON invalid
500	Error	Cannot create / update Function module

`DELETE /function-modules/:name`

Delete the Function module configuration from database.

Status Code	Response	Description
204		Success
404	Not Found	Function module does not exists
500	Error	Cannot delete Function module

`GET /function-modules/:name/executable`

Get Function module executable as `Content-type: application/octet-stream`.

Status Code	Response	Description
200	`Object`	Success
404	Not Found	Function module does not exists
500	Error	Cannot get executable

`PUT /function-modules/:name/executable`

(C) 2013-2022 OpenDAF Contributors

Replace Function module executable. Request headers should be set to `Content-type: application/octet-stream`

Status Code	Response	Description
204		Success
400	Bad Request	JSON invalid
500	Error	Executable replace failed

`DELETE /function-modules/:name/executable`

Delete Function module executable.

Status Code	Response	Description
204		Success
403	Forbidden	Deletion forbidden by OS
404	Not Found	Function module does not exists
500	Error	Cannot delete executable

`GET /function-modules/:name/executable-stat`

Get Function module executable stat as `Object`. Result schema is executable_stat.json.

Status Code	Response	Description
200	`Object`	Success
404	Not Found	Function module does not exists
500	Error	Cannot get executable stat

Stacks

`GET /stacks/connector/names` `GET /stacks/provider/names`

Get names of all stacks for Connector or Provider as `Array<String>`.

Status Code	Response	Description
200	`Array<String>`	Success
500	Error	Cannot get list of Stacks' names

`GET /stacks/connector/` `GET /stacks/provider/`

Get properties of stack Connector or Provider as `Array<Object>`. Result schema is stacks.json.

- query parameters: *names*

Status Code	Response	Description
200	`Array<Object>`	Success
400	Bad Request	Query parameters invalid
500	Error	Cannot get list of Stacks

Stack

`GET /stacks/connector/:name` `GET /stacks/provider/:name`

(C) 2013-2022 OpenDAF Contributors

Get properties of the Stack as `Object`. Result schema is stack.json.

Status Code	Response	Description
200	`Object`	Success
404	Not Found	Stack does not exists
500	Error	Cannot get Stack

5.4.3 JSON schemas

Download schemas archive here

Top-level document schemas

ALARM.JSON

```
{
    "$schema": "http://json-schema.org/draft-07/schema#",
    "title": "Alarm schema",
    "type": "object",
    "properties": {
        "name": {
            "title": "Unique name",
            "$ref": "identifier.json"
        },
        "description": {
            "title": "Alarm description",
            "type": ["string", "null"]
        },
        "severity": { "$ref": "severity.json" },
        "archMode": {
            "title": "Archivation mode",
            "type": "string",
            "enum": [ "none", "change" ]
        },
        "ackMode": {
            "title": "Acknowledge mode",
            "type": "string",
            "enum": [ "manual", "auto" ]
        },
        "enabled": {
            "type": "boolean"
        },
        "trgMeasurement": {
            "type": ["string", "null"],
            "title": "Trigger measurement"
        },
        "trgOperator": {
            "type": ["string", "null"],
            "title": "Trigger operator",
            "enum": [ "lt", "gt", "le", "ge", "eq", "ne", null ]
        },
        "trgRefValue": {
            "type": ["string", "null"],
            "title": "Reference value"
        },
        "trgHysteresis": {
            "type": ["string", "null"],
            "title": "Alarm hysteresis"
        },
        "groupName": {
            "type": ["string", "null"],
            "title": "Alarm group name"
        },
        "properties": { "$ref": "properties.json" }
    },
    "required": [ "name", "severity", "archMode", "ackMode", "enabled" ]
}
```

ALARMS.JSON

```
{
    "$schema": "http://json-schema.org/draft-07/schema#",
    "title": "Map of alarms (name -> alarm)",
    "type": "object",
    "patternProperties": {
        "[_a-zA-Z0-9]": { "$ref": "alarm.json" }
    }
}
```


(C) 2013-2022 OpenDAF Contributors

RENDER.JSON

```
{
    "$schema": "http://json-schema.org/draft-07/schema#",
    "title": "Render schema",
    "type": "object",
    "properties": {
        "up_to_date": {
            "title": "Configuration render up-to-date status",
            "type": "boolean"
        }
    },
    "required": [ ]
}
```

COMMUNICATION_OBJECT.JSON

This is not true top-level document, but is a base schema for measurement.json and command.json.

```
{
    "$schema": "http://json-schema.org/draft-07/schema#",
    "title": "Communication object schema",
    "type": "object",
    "properties": {
        "name": {
            "title": "Unique name",
            "$ref": "identifier.json"
        },
        "address": { "type": ["string", "null"] },
        "description": { "type": ["string", "null"] },
        "datatype": {
            "title": "Datatype",
            "$ref": "datatype.json"
        },
        "rawDatatype": {
            "title": "Raw datatype",
            "anyOf": [ { "$ref": "datatype.json" }, { "type": "null" } ]
        },
        "connectorName": {
            "title": "Name of Connector",
            "type": ["string", "null"]
        },
        "rawRangeLo": {
            "title": "Raw range low",
            "type": ["string", "null"]
        },
        "rawRangeHi": {
            "title": "Raw range high",
            "type": ["string", "null"]
        },
        "euRangeLo": {
            "title": "Engineering unit range low",
            "type": ["string", "null"]
        },
        "euRangeHi": {
            "title": "Engineering unit range high",
            "type": ["string", "null"]
        },
        "providerAddresses": {
            "title": "Provider Addresses",
            "type": "object",
            "patternProperties": {
                ".{1,}": { "type": ["string", "null"] }
            }
        },
        "archMode": {
            "title": "Archivation mode",
            "type": "string",
            "enum": [ "none", "change" ]
        },
        "archPeriod": {
            "title": "Archivation period",
            "type": "integer"
        },
        "archValueDeadband": {
            "title": "Archivation value deadband",
            "type": ["string", "null"]
        },
        "archTimeDeadband": {
            "title": "Archivation time deadband [ms]",
            "type": "integer"
        },
        "leader": { "type": ["string", "null"] },
        "stackUmask": {
            "title": "Stack umask (disabled modules / features)",
            "type": "integer"
        },
```


(C) 2013-2022 OpenDAF Contributors

```
        "eu": {
            "title": "Engineering unit",
            "type": ["string", "null"]
        },
        "enabled": { "type": "boolean" },
        "properties": { "$ref": "properties.json" }
    },
    "required": [ "name", "datatype", "archMode", "stackUmask", "enabled" ]
}
```

MEASUREMENT.JSON

```
{
    "$schema": "http://json-schema.org/draft-07/schema#",
    "title": "Measurement schema",
    "type": "object",
    "allOf": [
        { "$ref": "communication_object.json" },
        {
            "type": "object",
            "properties": {
                "deadband": {
                    "type": ["string", "null"]
                 }
            },
            "required": []
        }
    ]
}
```

MEASUREMENTS.JSON

```
{
    "$schema": "http://json-schema.org/draft-07/schema#",
    "title":    "Map of measurements (name -> measurement)",
    "type":     "object",
    "patternProperties": {
        "[_a-zA-Z0-9]": { "$ref": "measurement.json" }
    }
}
```

COMMAND.JSON

```
{
    "$schema": "http://json-schema.org/draft-07/schema#",
    "title": "Command schema",
    "type": "object",
    "allOf": [
        { "$ref": "communication_object.json" },
        {
            "type": "object",
            "properties": {
                "initialValue": {
                    "title": "Command initial value",
                    "type": ["string", "null"]
                 }
            },
            "required": []
        }
    ]
}
```

COMMANDS.JSON

```
{
    "$schema": "http://json-schema.org/draft-07/schema#",
    "title":    "Map of commands (name -> command)",
    "type":     "object",
    "patternProperties": {
        "[_a-zA-Z0-9]": { "$ref": "command.json" }
    }
}
```

FUNCTION_MODULE.JSON

```
{
    "$schema": "http://json-schema.org/draft-07/schema#",
    "title": "Function module schema",
    "type": "object",
    "properties": {
```


(C) 2013-2022 OpenDAF Contributors

```
        "name": {
            "title": "Unique name",
            "$ref": "identifier.json"
        },
        "description": {
            "title": "Function module description",
            "type": "string"
        },
        "executable": {
            "title": "Executable image path",
            "type": "string"
        },
        "debug": { "type": "boolean" },
        "respawn": { "type": "boolean" },
        "enabled": { "type": "boolean" },
        "termToKill": {
            "title": "Term to kill [ms]",
            "type": "integer"
        },
        "queryTimeout": {
            "title": "Query Timeout [ms]",
            "type": "integer"
        },
        "runMode": {
            "title": "Run mode",
            "type": "string"
        },
        "properties": { "$ref": "properties.json" },
        "env": {
            "title": "Environment variables",
            "type": "object",
            "patternProperties": {
                ".{1,}": { "type": ["string", "null"] }
            }
        }
    },
    "required": [ "name", "executable", "debug", "respawn",  "termToKill", "queryTimeout", "runMode", "env" ]
}
```

FUNCTION_MODULES.JSON

```
{
    "$schema": "http://json-schema.org/draft-07/schema#",
    "title": "Map of function modules (name -> function module)",
    "type": "object",
    "patternProperties": {
        "[_a-zA-Z0-9]": { "$ref": "function_module.json" }
    }
}
```

STACK_INSTANTIATION.JSON

This is mutual schema for Connector and Provider.

```
{
    "$schema": "http://json-schema.org/draft-07/schema#",
    "title": "Stack Instantiation (Connector / Provider) schema",
    "type": "object",
    "properties": {
        "name": {
            "title": "Unique name",
            "$ref": "identifier.json"
        },
        "stackName": {
            "title": "Stack name",
            "type": "string"
        },
        "wdtMeasurement": {
            "title": "Watchdog measurement name",
            "type": ["string", "null"]
        },
        "vars": {
            "title": "Variables",
            "type": "object",
            "patternProperties": {
                ".{1,}": { "type": ["string", "null"] }
            }
        },
        "properties": { "$ref": "properties.json" }
    },
    "required": [ "name", "stackName", "vars" ]
}
```


(C) 2013-2022 OpenDAF Contributors

STACK_INSTANTIATIONS.JSON

This is mutual schema for Connectors and Providers

```
{
    "$schema": "http://json-schema.org/draft-07/schema#",
    "title": "Map of stack instantiations (name -> connector / provider)",
    "type": "object",
    "patternProperties": {
        "[_a-zA-Z0-9]": { "$ref": "stack_instantiation.json" }
    }
}
```

STACK.JSON

```
{
    "$schema": "http://json-schema.org/draft-07/schema#",
    "title": "Stack schema",
    "type": "object",
    "properties": {
        "name": {
            "title": "Unique name",
            "$ref": "identifier.json"
        },
        "description": {
            "title": "Stack description",
            "type": "object"
        },
        "defaults": {
            "title": "Stack defaults",
            "type": "object"
        }
    },
    "required": [ "name" ]
}
```

STACKS.JSON

```
{
    "$schema": "http://json-schema.org/draft-07/schema#",
    "title": "Map of stacks (name -> stack)",
    "type": "object",
    "patternProperties": {
        "[_a-zA-Z0-9]": { "$ref": "stack.json" }
    }
}
```

EXECUTABLE_STAT.JSON

```
{
    "$schema": "http://json-schema.org/draft-07/schema#",
    "title": "Function module executable stat schema",
    "type": "object",
    "properties": {
        "mode": { "type": "integer" },
        "uid": { "type": "integer" },
        "gid": { "type": "integer" },
        "size": { "type": "integer" },
        "atime": { "type": "integer" },
        "mtime": { "type": "integer" },
        "ctime": { "type": "integer" }
    },
    "required": []
}
```

TXN.JSON

```
```

Basic type schemas

DATATYPE.JSON

```
{
    "$schema": "http://json-schema.org/draft-07/schema#",
    "enum": [ "b", "q", "i", "l", "f", "d", "s" ]
}
```


(C) 2013-2022 OpenDAF Contributors

```
```

IDENTIFIER.JSON

```
{
    "$schema": "http://json-schema.org/draft-07/schema#",
    "pattern": "^[_a-zA-Z0-9]{1,64}$"
}
```

SEVERITY.JSON

Numeric alarm severity value.

```
{
    "$schema": "http://json-schema.org/draft-07/schema#",
    "title": "Alarm severity",
    "type": "integer",
    "minimum": 0,
    "description": "Alarm with severity 0 is the most severe. Severity decreased with its value rising."
}
```

PROPERTIES.JSON

Additional objects' properties

```
{
    "$schema": "http://json-schema.org/draft-07/schema#",
    "title":    "Map of properties ( Map<String, String> ))",
    "type":     "object",
  "patternProperties": {
    ".{1,}": { "type": ["string", "null"] }
  }
}
```


(C) 2013-2022 OpenDAF Contributors

6. JavaScript API

OpenDAF runtime provides following built-in modules:

- `opendaf` (runtime data access)
- `dafman` (configuration database access)
- `ice` (*Integrated Calculation Engine* functions)
- `file` (file access)
- `os` (operating system functions)
- `net` (network functions)

Module APIs are documented here.

(C) 2013-2022 OpenDAF Contributors

7. Troubleshooting

The fact that OpenDAF is of a highly modular architecture can be readily used to help hunt down many kind of issues.

By selectively disabling/enabling individual modules, performance bottlenecks can be quickly found, and bugs quickly isolated.

This disabling/enabling of individual modules (and even measurements, commands, alarms) is achieved by editing configuration files in `/etc/opendaf/`.

The easiest way do disable the module is to comment it out with `#` on line beginning.

Files can be regenerated with `dafman render --force`, if they are changed beyond repair.

To apply changes, opendaf processes must be restarted:

```
systemctl restart opendaf
systemctl restart opendaf-archive
```

In following sections individual file groups will be described.

7.1 `svc.conf`

This is realtime opendaf process service configuration files. Only service libraries configured in this file are loaded into the process.

Interesting services:

- Logger (logging configuration)
- Core (transaction execution - enable logging with `--dump-transactions 1`)
- MeasurementStack (enable/disable individual layers)
- CommandStack (enable/disable individual layers)

7.2 `svc-recorder.conf`

Service configuration file of opendaf archivation process.

Interesting services:

- Logger (logging configuration)
- Core (transaction execution - enable logging with `--dump-transactions 1`)

7.3 `connectors.conf`

Connectors are implemented as services, and each connector has an entry in this file. Each connector can be individually disabled.

7.4 `connector-<name>.vars`

Configuration variables of given connector. To quickly experiment with connector settings, they can be changed here without the need to change configuration database.

7.5 `providers.conf`

Providers are implemented as services, and each provider has an entry in this file. Each provider can be individually disabled.

(C) 2013-2022 OpenDAF Contributors

7.6 provider-<name>.vars

Configuration variables of given provider. To quickly experiment with provider settings, they can be changed here without the need to change configuration database.

7.7 measurements.csv, commands.csv, alarms.csv, alarm-groups.csv

Data tables of individual object classes.

By manipulating rows in these tables, various setups can be tested quickly, without need to change configuration database.

(C) 2013-2022 OpenDAF Contributors

8. Re-measuring the 2026-10 perf optimizations on the target

The `perf-opt` branch (commits `13eb0832` .. `071e543a`) attacks the hot spots found in the ARM32 profile at `/home/stano/projekty/falcon/bugs/20261006/prof/perf-report.txt` (33K cpu-clock samples of the `opendaf` nucleus, ~15 modbus units polled at 4 Hz plus panter modulators). This guide says how to confirm the diagnosis on the OLD binaries, what to measure, and what the NEW binaries should show.

8.1 What was changed, and which profile line each change targets

Change	Commit	Profile cost it targets
Transaction kind tag replaces `dynamic_cast` on the commit path	`perf(core): transaction kind tag`	`dynamic_cast` + `type_info::operator==` + `strcmp` ≈ 6.5% (ARM EABI compares type names by string; the C→* follower casts failed on every measurement commit)
Followers subscribe to their leader object, not the global commit signal	`perf(core): followers subscribe per leader object`	unresolved libopendaf-core cluster `0x4e494` / `0x4e4ac` / `0x4e4dc` ≈ 14.5% (boost::signals2 emission loop) + pthread mutex ≈ 5.7% + command-csv-loader self 5.2% (its hidden template copies of the slot code)
Edge-triggered stream-head notify (TX_COMPLETE/LOCK forwarded again since review round 13: the edge already coalesces their notifies)	`perf(stack): edge-triggered stream-head notify`	`unix_stream_sendmsg` 3.3% kernel (one notify-pipe byte per enqueue, ~160/s; Dev_Poll never coalesces)
Cached scan request blocks, `--locking` bypass, exact-length RX copy, blocade deadline timer	`perf(modbus-master): cached scan requests, ...`	per-cycle allocations/copies on the ~60 cycles/s serial path
Panter transaction coalescing, dead METER_READOUT query, subscriber-aware scan + bug fixes	`fix/perf(panter): ...`	5-6 transactions+notifies per internal query → 1; one wasted serial round-trip per scan period
`REPORT_CHANGES_ONLY=1`, `PUBLISH_INTERNAL_FSM_STATES=0` (new defaults)	`feat(panter): ...`	1 Hz unchanged-counter pushes, FSM 20→30→20 flip per query

8.2 1. Before upgrading: confirm the diagnosis on the old build

The 14.5% cluster is attributed to boost::signals2 emission hidden under `-fvisibility=hidden`. Confirm with the unstripped old library (take it from the matching `-dbg` package or the build tree):

```
addr2line -f -C -e libopendaf-core.so.2.13.3 0x4e494 0x4e4ac 0x4e4dc
```

Expected: frames inside `boost::signals2::detail::signal_impl::operator()` / slot iteration instantiated for `core::Event<void(core::Transaction*)>`. If these resolve somewhere else entirely, stop and re-profile before trusting the expectations below.

Baseline the notify-pipe traffic (writes of 1 byte to the reactor notify fd) and overall CPU on the running old daemon:

```
pidof dafmand                       # -> PID
strace -f -c -e trace=write,sendmsg -p $PID & sleep 60; kill %1
pidstat -p $PID 10 6                # %CPU baseline
```


(C) 2013-2022 OpenDAF Contributors

Record: writes/s on the notify fd (expect in the order of ~160/s with the falcon config), total %CPU, and keep the perf invocation used for the original report so the after-run is comparable:

```
perf record -F 997 -a -g -- sleep 60
perf report --sort dso,symbol > perf-report-before.txt
```

8.3 2. Upgrade

Build `perf-opt` as RelWithDebInfo (or keep the unstripped libs around — otherwise the after-profile is as unreadable as the before one). No configuration changes are required: the new panter parameters default to the optimized behavior, `LOCKING` defaults to 0 which now also short-circuits the in-stack lock messages.

To restore any legacy stream while validating:

- `REPORT_CHANGES_ONLY=0` — counters and permanent-error republished every tick as before, 1 Hz execution-queue timer back.
- `PUBLISH_INTERNAL_FSM_STATES=1` — FSM state published on every transition including the internal 20→30→20 flips.
- `LOCKING=1` — full lock message flow (only meaningful with a locking serial-datalink setup).

Known behaviour, panter: each request/response cycle now reaches the repository as ONE transaction, and a transaction commits all or nothing. A single misconfigured measurement (datatype not matching its value, e.g. a stackUmask that skips the scaler) therefore discards the whole cycle's updates - counters, FSM state, permanent error, ID data - not only its own. With `REPORT_CHANGES_ONLY=1` those values come back only on their next change. The log shows both sides; grep for:

```
Measurement(<name>): Datatype <x>, can't update with <y>!
MeasurementUpdateTransaction rolled back: N of M updates invalid, all M discarded!
```

Fix the named measurement's datatype or stackUmask.

Behavioral notes for whoever validates SCADA-side: - Follower updates now always commit AFTER their leader's transaction (previously the order depended on subscription order). - `AC_MV_ID_INFO` arrives once per interrogated node (it was duplicated). - Panter stored-data values now pass raw→EU processing and deduplication (previously committed directly, bypassing the stack). - Delayed panter commands execute punctually instead of up to 1 s late. - The periodic METER_READOUT query (FC3 @ 0x65) is gone — its response was never parsed.

8.4 3. After: what the same measurements should show

Repeat the perf record, strace and pidstat runs with identical durations and load, then compare:

Metric	Before	Expected after
libopendaf-core unresolved cluster (`0x4e494` area)	≈ 14.5%	collapses to low single digits (emission loop now runs per-leader, not per-commit × per-follower)
`dynamic_cast` / `type_info` / `strcmp` triplet	≈ 6.5%	essentially gone from the commit path
`pthread_mutex_lock/unlock`	≈ 5.7%	roughly halved (far fewer signal slots per commit)
command-csv-loader self time	≈ 5.2%	≈ 0 (loaders now call core factories; no local template copies — verify with `nm -D` that the loader .so only imports `createCommandFollower`)
notify-fd writes/s	~160/s	tens/s (edge-triggered; one wakeup per burst)
panter transactions per internal query	5–6	1
overall dafmand %CPU	baseline	down roughly a quarter to a third

If a line does not move as expected, diff the per-DSO view (`perf diff perf-before.data perf-after.data`) before looking for new causes — the usual culprit is comparing builds with different visibility/strip settings rather than a regression.

(C) 2013-2022 OpenDAF Contributors

9. Measuring satellite (recorder/dispatcher) intake performance

The recorder and dispatcher ingest transactions from the main opendaf process over transaction-serdes on TCP loopback :6378. This guide covers measuring the 2026-10 intake optimizations on a deployment, and reproducing the loopback smoke locally.

9.1 What changed, what to expect

Change	Where	Expected effect
Geometric MOPS growth + snapshot pre-sizing	main, serialization	time-to-snapshot after satellite (re)connect drops from O(N²) memcpy to linear; large plants no longer stall the main reactor per reconnect
Serialize-once fan-out	main, per commit	serialization CPU halves with two satellites
One chunk per transaction + sendv + edge-armed WRITE_MASK + TCP_NODELAY (opt-in: `--tcp-nodelay=1` on the serdes links only)	main, socket writer	send syscalls per transaction: ≥2 → 1; epoll_ctl per message → per burst; tens of ms Nagle latency gone
Periodic deserialization watchdog	satellites	two timer-heap ops per received transaction → none (5 s tick; timeout detection granularity 10–15 s)
Recycled 64 KiB receive block	satellites	one alloc/free pair per readable event → none
Unknown-object tolerance	satellites	CSV drift no longer causes a 5 s reconnect + full-snapshot storm; rate-limited warnings name the object
Dispatcher `--request AUTO`	dispatcher	without process-variable rule templates: the measurement/command streams are not requested at all — the dominant dispatcher load disappears on both ends
Archive first-touch filtering, in-place deadband, RTTI-free sampling	recorder	per-archived-commit allocations drop; string/none objects stop at the first stage

9.2 On a deployment

Baseline before upgrading, repeat after, same durations:

```
PID=$(pidof dafmand || pidof opendaf)       # main process
# syscall profile of the serdes link (sends, epoll_ctl):
strace -f -c -p $PID & sleep 60; kill %1
# CPU of all three processes:
pidstat -p $(pgrep -d, -f 'opendaf|dafmand') 10 6
```

Time-to-snapshot: restart one satellite and measure from its "Connector(...): Connected" log line to the recorder's "Archive: N initial samples taken" (journalctl timestamps).

Dispatcher advice: after start it logs `Dispatcher: Intake advice: alarm state changes only` or `... + measurement/command streams (rule templates read process variables)`. Verify it matches the site's rules; `--request AUTO` falls back to the full legacy mask until the dispatcher publishes advice.

(C) 2013-2022 OpenDAF Contributors

9.3 Loopback smoke (from the build tree, no install)

1. Scratch profile: a cfgroot with `measurements.csv` (a few hundred rows, any fake connector name), empty `commands.csv` / `alarms.csv` / `alarm-groups.csv`, and three service confs modeled on `src/lib/dafman/share/svc-recorder-unix.conf` / `etc/opendaf/default/svc-dispatcher.conf`:
2. main: Core + csv loaders + `opendaf-stream-socket-acceptor` on `127.0.0.1:16378` with a `trnserdes.conf` gestalt + CivetWeb + `opendaf-rest-api-measurements`;
3. recorder/dispatcher: Core + csv loaders + SourceStream (`socket-stream-connection` + `transaction-serdes --des`) + connector to `127.0.0.1:16378` + Archive (scratch LMDB dir) or Dispatcher (empty rules dir, `--request AUTO`).
4. `export LD_LIBRARY_PATH=$(find build/src -name "libopendaf-*.so" -printf '%h\n' | sort -u | paste -sd:)`
5. Run `build/src/opendaf/opendaf -f <conf>` for each process; the REST API injects live commits: `curl -X POST "http://127.0.0.1:18090/measurements/M_SIM_1?value=f1.5"`.
6. Pass criteria: recorder logs the full snapshot ("N initial samples taken") and its `data.mdb` grows with injected updates; the dispatcher logs its intake advice; no ERROR/WARNING in any log; `strace -c` on the main process shows roughly one `sendto`/`write` per injected commit per full-mask satellite.

(C) 2013-2022 OpenDAF Contributors