

Filter plugin OPCUA to Pivot

This plugin intends at converting PIVOT data from/to [OPCUA north plugin](#).

The filter implements the Fledge Filter plugin interface (see [filter_plugins](#)).

Filter configuration

See [Plugins configuration design](#) for examples and details

The "config" parameter of "plugin_init" call shall include :

- A "exchanged_data" category with the same content as provided to the OPCUA north plugin. This section is mandatory so that the filter plugin can convert the PIVOT type to OPCUA type.

Notes :

- All types not listed in this table are not supported in current version.
- The current implementation provides a default mapping rule for each known type, but some new rules might be added in the future and configured using the "alternate_mapping_rule" option in its parameters.
- While converting from PIVOT to OPCUA, the PIVOT object is "standalone" and the plugin does not need any configuration from this transformation. The "Exchanged Data" section is only used by conversion from OPCUA to PIVOT.

Filter interface

The <Root> key of PIVOT object can be:

Reading key	Content
PIVOT.GTIS	Tele Signal
PIVOT.GTIM	Tele Measurement
PIVOT.GTIC	Tele Control

In the following table: <type> is equal to

- "SpsTyp", "DpsTyp" (for GTIS)
- "MvTyp" (for GTIM)
- "SpcTyp", "DpcTyp", "IncType" or "ApcTyp" (for GTIC)

The "BscTyp" is not supported currently.

Type of data

The plugin is not responsible for checking that the conversion matches its actual role: typically if the plugin is attached to a North OPCUA service, it will be expected to convert OPCUA to PIVOT, but will anyway convert PIVOT to OPCUA if receiving PIVOT instead of OPCUA as reading. This consistency must be ensured by configuration.

Thus the rules for the conversion is:

- For each object in the reading is modified as follow:
 - If the object name is "**PIVOT**" and contains the "**GTIM**" or "**GTIS**" key, then the reading must be converted from a PIVOT object to an OPCUA "data object".
 - If the object name is "**PIVOT**" and contains the "**GTIC**" key, then the reading must be converted from a PIVOT object to an OPCUA "reply object".
 - If the object name is "**opcua_operation**", then the reading must be converted from an OPCUA control request (operation) to a PIVOT object.
 - Otherwise the reading is not modified.

Converting monitoring data types

For TM/TS, the filter receives PIVOT reading (from south) and creates a corresponding OPC UA reading. The created OPCUA reading has the following JSON equivalent format ("data object"):

```

{
  "data_object":{
    "do_cot": <int>,
    "do_confirmation": <int>,
    "do_comingfrom": <string>,
    "do_id": <string>,
    "do_type": <string>,
    "do_quality": <int>,
    "do_ts_quality": <int>,
    "do_value_quality": <string>,
    "do_source": <string>,
    "do_ts": <int>,
    "do_ts_org": <string>,
    "do_ts_validity": <string>,
    "do_value": <depends on data type...>
  }
}

```

The content under <Root> will convert the pivot to a opcua object as follow:

Key	Type	Default Value	opcua.data_object. <...>	Note
<Root>.Cause.stVal	Integer	Mandatory	do_cot	See Cause of Transmission
<Root>.Confirmation.stVal	Boolean	false	do_confirmation	
<Root>.ComingFrom	String	Mandatory	do_comingfrom	Any protocol name ("iec104", "opcua", ...)
<Root>.Identifier	String	Mandatory	do_id do_type	do_id = PIVOT ID do_type = "opcua_sps" "opcua_dps" "opcua_mvi" "opcua_mv". Other types not supported in current version.
<Root>.<type>.q.DetailsQuality <Root>.<type>.q.test <Root>.<type>.q.operatorBlocked	PIVOT : Object OPCUA : Integer	0	do_quality	OR-Mask of following values: 0x0001 = badReference 0x0002 = failure 0x0004 = inconsistent 0x0008 = innaccurate 0x0010 = oldData 0x0020 = oscillatory 0x0040 = outOfRange 0x0080 = overflow 0x1000 = test 0x2000 = operator blocked
<Root>.<type>.t.TimeQuality	Integer	0	do_ts_quality	OR-Mask of following values: 0x01 = clockFailure 0x02 = clockNotSynch 0x04 = leapSecondKnown
<Root>.<type>.q.Source	String	"process"	do_source	"process" "substituted"
<Root>.<type>.q.Validity	String	Mandatory	do_value_quality	"good" "invalid" "reserved" "questionable"
<Root>.<type>.t.SecondSinceEpoch	Integer	0	do_ts	Unit = seconds since Epoch (January, 1st 1970)
<Root>.TmOrg.stVal	String	"genuine"	do_ts_org	"genuine" "substituted"
<Root>.TmValidity.stVal	String	"good"	do_ts_validity	"good" "invalid" "reserved" "questionable"
<Root>.SpsTyp.stVal	Boolean	Mandatory	do_value	
<Root>.DpsTyp.stVal	String	Mandatory	do_value	See CDC double point status (DPSTyp)
<Root>.MvTyp.mag.f	Float	Mandatory	do_value	
<Root>.MvTyp.mag.i	Integer	Mandatory	do_value	

Converting commands data types

Converting commands implies two different use case:

- While sending a Fledge "operation" (from North to South). In that case the filter converts an operation with asset_name = "**opcua_operation**" into a "**PIVOTTC**" object with an asset_name = "**PivotCommand**".
- While receiving an acknowledgement for a previous "operation" (from South to North). In that case, the filter converts a "**PIVOT.GTIC**" into a "**opcua_reply**" object.

The conversion rules are details below.

Conversion from opcua_operation to PIVOT.GTIC

opcua_operation.<...>	PIVOT field ("PIVOT.GTIC".<...>)	Type	Default value	Note
co_id	Identifier	String		PIVOT identifier.
co_se	Select.stVal	Boolean	false	- 0 is mapped with false, for Execute - 1 is mapped with true, for Select before Execute
N.A.	ComingFrom	String	"opcua"	This plugin always uses the value "opcua" when converting to PIVOT
co_type	<type>	String		"opcua_spc" "opcua_dpc".
co_test	<type>.q.test	Boolean	false	
co_ts	<type>.t.SecondSinceEpoch	Integer		
co_value	ApcTypctlVal	Float		Float 32
co_value	DpcTypctlVal	String		intermediate-state off on bad-state
co_value	IncTypctlVal	Integer		Int 32
co_value	SpcTypctlVal	Boolean		0 or 1
co_value	BscTypctlVal	String		stop lower higher reserved

Conversion from PIVOT.GTIC to "opcua_reply"

PIVOT field ("PIVOT.GTIC".<...>)	opcua_reply field	Type	Default value	Note
Identifier	ro_id	String		PIVOT identifier.
Confirmation.stVal	ro_reply	Boolean		Converts the values as follow: ▪ confirmation.stval = false then ro_reply = 0 (positive confirmation) ▪ confirmation.stval = true then ro_reply = 1 (negative confirmation)
<All other fields are ignored>				