logger

This page explains the **Logger adapter** in Bridge context. If you were looking for the same information regarding the PAS Designer, refer to Logger Adapter in the Designer guide.

Tagged Values of the Logger Configuration

<<LogChannel>>

Runtime 2020.8 Builder 7.10.1

Tagged Value	Description	Allowed Values
name	Name of the channel.	

<<LogSink>>

Runtime 2020.8 Builder 7.10.1

Tagged Value	Description	Allowed Values / Example	
name	Name of the sink.		
fileNamePattern	Specify a filename/path pattern for the log file to be generated. You can use the variables listed at Log Filename. This tagged value will be only taken into account if you specify a **sink Type** that logs to a file.	`logs/{channel}.{sink}_{date_time}.log`	
level	Specify a log level. This log level serves as a filter and is the lowest log level this sink will accept. Refer to Bridge Server Log Levels of an xUML Service for a list of log levels.	One of **Fatal**, **Error**, **Warning**, **Info** (default), and **Debug**.	
sinkType	Specify where to log to. If logging to a file, the filename will be generated as specified with **fileNamePattern**.	**daily_rotated_file**	Log to a daily rotated file (default).
		hourly_rotated_file	Log to an hourly rotated file.
		std_out	Log to std_out.
		std_err	Log to std_err.

<<LogFormatter>>

Runtime 2020.8 Builder 7.10.1

Tagged Value	Description	Allowed Values / Example	
format	Specify the log format.	**json**	Log contents will be formatted as JSON (default).
		pattern	Log contents will be formatted by a pattern given with **pattern**.
pattern	Specify a pattern for the log content format. This is only necessary if "pattern" is selected as **format**. You can use the variables and formatting options listed at Log Content.	`[{timestamp:%d.%m.%Y %H:%M:%S}][{level}] {message}`	

On this Page:

- Tagged Values of the Logger Configuration
  - <<LogChannel>>
  - <<LogSink>>
  - <<LogFormatter>>
  - <<LogKeys>>
- Log Adapter Parameters
  - Action "log"
  - Action "logTo"
  - Action "report"
- Log Filename
  - Pattern Variables
  - Pattern Formatting Options
- Log Content
  - Text Pattern Variables
  - JSON Attributes
  - Content Formatting Options

Related Pages:

- log() Function
- report() Function

Related Documentation:

- Bridge Server Log Levels of an xUML Service

timestamp_resolution	Runtime 2023.4 Specify the resolution of the timestamp to be logged. • Custom timestamps are limited to a resolution of **microseconds** due to the precision of the DateTime type. If you specify a higher resolution, "000" will be appended for nanoseconds. • The same will be done if the platform the service is running on does not provide an accurate enough timestamp, e.g. Windows is limited to a resolution of 100 nanoseconds.	**seconds**	Log timestamps with a resolution of seconds (default).
		milliseconds	Log timestamps with a resolution of milliseconds.
		microseconds	Log timestamps with a resolution of microseconds.
		nanoseconds	Log timestamps with a resolution of nanoseconds.

<<LogKeys>>

Runtime 2020.8 Builder 7.10.1

Tagged Value	Description	Allowed Values	
name	Specify the name of the log key you want to overwrite in the JSON file.	channel	Overwrites the key of the channel name.
		at	Overwrites the key of the timestamp.
		message	Overwrites the key of the error description.
		details	Overwrites the key of the error details.
externalName	Specify the name that should be written to the JSON file as a key instead of **name**.	Any string that matches requirements of a JSON key.	
externalType	Specify the type of the changed JSON attribute. Can only be used with **DateTime** values for now, resp. attribute **at**.	integer	Convert content to **Integer**.
		string	Convert content to **String**.
formatString	Specify a format string for the logged value. Can only be used with **DateTime** values for now, resp. attribute **at**.	A valid format string. Refer to Content Formatting Options further below for more information on formatting options.	

Log Adapter Parameters

Action "log"

Name	Type	Direction	Description	Allowed Values / Example
level	String	in	Specify one of the error log levels described in the log level guidelines on Bridge Server Log Levels of an xUML Service.	One of **Fatal**, **Error**, **Warning**, **Info**, and **Debug**.
type	String	in	Specify a custom error domain.	`Stock_Synchronization`
code	String	in	Specify a custom error code.	`SYNC001`
description	String	in	Specify a description for the log entry.	`Item ID could not be found.`

details	Any	in	Runtime 2020.6 Builder 7.8.0 Specify an object of complex type (class or array) that contains additional details. If provided, the contents of this object will be flattened and appended to the **description**.	an error detail object

Action "logTo"

Runtime 2020.8 Builder 7.10.1

Name	Type	Direction	Description	Allowed Values / Example
channel	String	in	Specify the channel you want to log to.	
level	String	in	Specify a log level. Allowed error log levels are described in the log level guidelines on Bridge Server Log Levels of an xUML Service.	One of **Fatal**, **Error**, **Warning**, **Info**, and **Debug**.
message	String	in	Specify a description for the log entry.	`Item ID could not be found.`
details	Any	in	Specify an object of complex type (class or array) that contains additional details. If provided, the contents of this object will be flattened and appended to the **description** for text files. In JSON files, **details** have their own key.	any detail object

Action "report"

Object Name	Type	Direction	Description	Field in Transaction Log	Allowed Values / Example	
domain	String	in	Specify a self-defined name of the domain of the logged entry.	Domain	`Stock_Synchronization`	
state	Boolean	in	Specify how to set the status field in the log entry. This will result in setting the status field in the log entry.	State	true	Set the status field to **OK**.
					false	Set the status field to **ERROR**.
parameter	String	in	Specify a self-defined parameter.	Parameter 2	`Item: YY-2388`	
	Any		Runtime 2020.6 Builder 7.8.0 Specify an object of complex type (class or array) that contains additional details. The contents of this object will be flattened and logged instead of **parameter**.		`e2e: type='ErrorDetail', element='YY-2388', value='Connector'`	
description	String	in	Specify a description for the log entry.	Parameter 1	`Item ID could not be found.`	
timestamp	DateTime	in	Bridge 7.0.0-beta2 Specify a custom timestamp. This parameter is optional. If it is not given, the timestamp in the transaction log will be generated from the current timestamp. **timestamp** and **parameter** (Any) are mutual exclusive: You cannot	Date, Time	NULL	Generated timestamp in the transaction log.

			provide a custom timestamp when providing a details object. In this case, the custom timestamp will be overwritten by the actual date and time.		a v al ue	Dedicated timestamp in the transaction log.

Log Filename

Pattern Variables

Runtime 2020.8 Builder 7.10.1

Variable	Description
`{channel}`	Channel name from the <<LogChannel>> class.
`{sink}`	Sink name from **name**.
`{date_time}`	Timestamp of format `%Y-%m-%d` or `%Y-%m-%d-%H`, depending on the **sinkType** .
`{extension}`	Value depends on the formatting specified in the <<LogFormatter>> class: • **json**: If **json** is selected as log format. • **log**: If **pattern** is selected as log format.

Pattern Formatting Options

Pattern	Description
`%Y`	year, 4 digits
`%m`	month, 2 digits
`%d`	day of the month, 2 digits
`%H`	hour, 24-hour format, 2 digits
`%M`	minute, 2 digits
`%S`	second, 2 digits

Log Content

Text Pattern Variables

Runtime 2020.8 Builder 7.10.1

Variable	Description
`{channel}`	Log channel.
`{timestamp}`	Timestamp.
`{level}`	Log level.
`{message}`	Log message.
`{compositeName}`	Name of the service composite (see Frontend Components).
`{trxId}`	Transaction id (see Contents of the Transaction Log).
`{correlationId}`	Correlation id (see Contents of the Transaction Log).

JSON Attributes

JSON Attribute	Description
`{channel}`	Log channel.
`{at}`	Timestamp.
`{message}`	Log message.
`{details}`	Serialized content of the details object.
`{compositeName}`	Name of the service composite (see Frontend Components).
`{trxId}`	Transaction id (see Contents of the Transaction Log).
`{correlationId}`	Correlation id (see Contents of the Transaction Log).

Content Formatting Options

You can format the output using pattern for text files and formatString for JSON files. When used in a **patt ern**, add the formatting to the variable like `{timestamp:<formatting options>}`.

Reasonable format strings are:

Pattern	Description	Example Output
Format Patterns For Date/Time Variables		
`%Y`	**Year** as a 4-digit decimal number.	`2020`
`%m`	**Month** as a 2-digit decimal number (range [01,12]).	`07`
`%d`	**Day** of the month as a decimal number (range [01,31]).	`15`
`%F`	Equivalent to `%Y-%m-%d`.	`2020-07-15`
`%H`	**Hour** as a 2-digit decimal number, 24 hour clock (range [00-23])	`16`
`%M`	**Minute** as a 2-digit decimal number (range [00,59])	`34`
`%S`	**Second** as a 2-digit decimal number (range [00,60])	`25`
`%T`	Equivalent to "%H:%M:%S" (the ISO 8601 time format).	`16:34:25`
`%z`	**Offset from UTC** in the ISO 8601 format, or no characters if the time zone information is not available.	`+0200`
Format Patterns for Numeric Variables		
`099`	**Sign-aware zero-padding** for numeric types, where 0 is the indicator to apply the padding and 99 the maximum padding width.	`010` 10-digit number with leading zeros.

If you want to apply more sophisticated formatting, find more options on the {fmt} documentation pages for string formats and cpp documentation pages for date/time formats.