




_bridge_excerpts

Group	Name	Excerpt																											
Action Language	deprecat ed_alias_ macros	<div>All macros that get information from aliases are deprecated since Builder 7.5.0. Please use the new Alias Reader instead.</div>																											
Action Language	writeTyp eDiscrimi nator	Runtime 2021.6 Use writeTypeDiscriminator to suppress the generation of xUML type properties ("e2e:type") to the generated JSON. If this option is true, the Runtime will write the original xUML type to the generated JSON in form of "e2e:type": "<name of the original xUML type>" if the type being serialized does not match the expected metadata.																											
Adapter	correlatio n_ID	The xUML Runtime assigns a correlation ID to each adapter call. This ID is stored in header field X-Bridge-CorrelationID . Adapter calls can be identified by this ID. Also, it is logged to the transaction log.																											
Adapter	hint_bac kslash_in _path	When using the Windows style with backward slashes "\" you have to be aware that you have escape this character. The escape character is also the "\". To avoid this, use forward slashes with Windows as well.																											
Adapter	request_ http_he ader_rol es	Builder 7.12.0 Runtime 2020.12 In the context of HTTP based adapters (URL , REST , SOAP), enable automatic header generation for the listed headers. These definitions overwrite the default behavior, and X-Transaction-Id , X-Request-Id , X-Sender-Host and/or X-Sender-Service will be substituted by this definition.																											
Adapter	request_ http_he ader_rol es_ _adapter	requestHttpHeaderRoles can hold a list of definitions in format <http header name>:<role>, that will automatically be generated for each adapter call on this alias. <role> can be one of the listed allowed values (one list entry per line). Refer to HTTP Header Support > Overwriting the Standard HTTP Headers for more details on header roles.																											
Education	documen tation_m ap	<p>The BRIDGE is delivered with comprehensive documentation. Apart from this self-study, the BRIDGE also comes with many examples you can have a look at. Each feature is subject of an example that is ready to run.</p> <p>This chapter provides an overview about all documentation available after installation. How to access the xUML examples is explained on Unzipping Examples. The complete documentation is available online.</p> <table><tr><th colspan="2">Document</th><th>Content</th></tr><tr><td colspan="2">BRIDGE Education Lesson 1-3</td><td>Comprehensive self-learning guide, which leads you through all steps of creating a Web service. Additionally, it provides conceptional information for each topic.</td></tr><tr><td colspan="2">Bridge User Guide</td><td>Comprehensive guide explaining concepts and usage of the BRIDGE, which comprises all management tasks to run the BRIDGE and deployed xUML, Node.js and Java services.</td></tr><tr><td rowspan="2">Reference Guides</td><td>Reference Guide</td><td>Comprehensive reference guide for xUML modelers containing information about concepts, modeling, xUML Action Language, add-ons, and import/export mechanisms.</td></tr><tr><td>Node.js Services Reference Guide</td><td>Guide to Scheer PAS Node.js modules that can be used by Node.js modelers in BRIDGE context.</td></tr><tr><td>Utilities</td><td><ul style="list-style-type: none">• Dashboards• Monitoring Service With UI• RESTifier for SAP• WSDL Generator for SAP RFC</td><td>Guides to BRIDGE utilities that may be helpful for modelers.</td></tr><tr><td colspan="2">Builder User Guide</td><td>Comprehensive guide explaining concepts and usage of BUILDER including all tools like xUML Model Compiler, xUML Model Debugger, and xUML Importers.</td></tr><tr><td rowspan="3">Installation Guides</td><td>BRIDGE Installation Guide</td><td>Description of the installation of BRIDGE including a troubleshooting chapter. This guide also contains a description of the following firmware installations:<ul style="list-style-type: none">• BRIDGE SQL Libraries to integrate SQL databases• BRIDGE SAP Libraries to integrate SAP systems• BRIDGE Java Libraries to use Java classes in your service</td></tr><tr><td>BUILDER Installation Guide</td><td>Description of the installation of the BUILDER including a troubleshooting chapter.</td></tr><tr><td>ANALYZER Installation Guide</td><td>Description of the installation of the ANALYZER.</td></tr></table>	Document		Content	BRIDGE Education Lesson 1-3		Comprehensive self-learning guide, which leads you through all steps of creating a Web service. Additionally, it provides conceptional information for each topic.	Bridge User Guide		Comprehensive guide explaining concepts and usage of the BRIDGE, which comprises all management tasks to run the BRIDGE and deployed xUML, Node.js and Java services.	Reference Guides	Reference Guide	Comprehensive reference guide for xUML modelers containing information about concepts, modeling, xUML Action Language, add-ons, and import/export mechanisms.	Node.js Services Reference Guide	Guide to Scheer PAS Node.js modules that can be used by Node.js modelers in BRIDGE context.	Utilities	<ul style="list-style-type: none">• Dashboards• Monitoring Service With UI• RESTifier for SAP• WSDL Generator for SAP RFC	Guides to BRIDGE utilities that may be helpful for modelers.	Builder User Guide		Comprehensive guide explaining concepts and usage of BUILDER including all tools like xUML Model Compiler, xUML Model Debugger, and xUML Importers.	Installation Guides	BRIDGE Installation Guide	Description of the installation of BRIDGE including a troubleshooting chapter. This guide also contains a description of the following firmware installations: <ul style="list-style-type: none">• BRIDGE SQL Libraries to integrate SQL databases• BRIDGE SAP Libraries to integrate SAP systems• BRIDGE Java Libraries to use Java classes in your service	BUILDER Installation Guide	Description of the installation of the BUILDER including a troubleshooting chapter.	ANALYZER Installation Guide	Description of the installation of the ANALYZER.
Document		Content																											
BRIDGE Education Lesson 1-3		Comprehensive self-learning guide, which leads you through all steps of creating a Web service. Additionally, it provides conceptional information for each topic.																											
Bridge User Guide		Comprehensive guide explaining concepts and usage of the BRIDGE, which comprises all management tasks to run the BRIDGE and deployed xUML, Node.js and Java services.																											
Reference Guides	Reference Guide	Comprehensive reference guide for xUML modelers containing information about concepts, modeling, xUML Action Language, add-ons, and import/export mechanisms.																											
	Node.js Services Reference Guide	Guide to Scheer PAS Node.js modules that can be used by Node.js modelers in BRIDGE context.																											
Utilities	<ul style="list-style-type: none">• Dashboards• Monitoring Service With UI• RESTifier for SAP• WSDL Generator for SAP RFC	Guides to BRIDGE utilities that may be helpful for modelers.																											
Builder User Guide		Comprehensive guide explaining concepts and usage of BUILDER including all tools like xUML Model Compiler, xUML Model Debugger, and xUML Importers.																											
Installation Guides	BRIDGE Installation Guide	Description of the installation of BRIDGE including a troubleshooting chapter. This guide also contains a description of the following firmware installations: <ul style="list-style-type: none">• BRIDGE SQL Libraries to integrate SQL databases• BRIDGE SAP Libraries to integrate SAP systems• BRIDGE Java Libraries to use Java classes in your service																											
	BUILDER Installation Guide	Description of the installation of the BUILDER including a troubleshooting chapter.																											
	ANALYZER Installation Guide	Description of the installation of the ANALYZER.																											

Installation	guiless_update	<p>Never change the guilessinstaller.properties after first installation, but use the same file for every update afterwards!</p> <p>Bridge 5.1.39.6 Bridge 6.0.40.2 Only exception is, when you are moving from a version below 6.0.40.2/5.1.39.6 to a higher version. Since then, the settings console.ip.address and console.hostname are mandatory. It is necessary that you edit your existing guilessinstaller.properties file before making an update installation. If you don't specify these settings, you will get additional node instances and proxy nodes and won't be able to deploy services anymore.</p> <ul style="list-style-type: none">• If console.hostname is empty use the value of Node Instance Name you find on the Preferences tab of the BRIDGE node instance.• If console.ip.address is empty use 0.0.0.0.																																																						
Logging	date_picker	<p>When filtering the log entries of a service by date and time, you can use a date picker to select a date from/to. Click the date picker icon  next to the input fields to open the a tiny calendar to pick the dates from.</p> <div><div>- < September 2018 > +</div><table><tr><td>Mo</td><td>Tu</td><td>We</td><td>Th</td><td>Fr</td><td>Sa</td><td>Su</td></tr><tr><td></td><td></td><td></td><td></td><td></td><td>1</td><td>2</td></tr><tr><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td></tr><tr><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td></tr><tr><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td></tr><tr><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td></tr></table><div><div>13</div><div>:</div><div>59</div><div>:</div><div>59</div></div><div><div>OK</div><div>Cancel</div></div></div> <p>The time part will be only visible if the related log entries contain a time part in format "HH:MM:SS". In all other cases, it is not possible to select log entries by time.</p> <p>Some dates within the calendar are colored to help you finding the appropriate date:</p> <table><tr><th>Color</th><th>Meaning</th></tr><tr><td>dark blue border</td><td>Today.</td></tr><tr><td>dark blue</td><td>The selected date.</td></tr><tr><td>bright blue</td><td>A weekday on which data has been logged.</td></tr><tr><td>light blue</td><td>A weekend day on which data has been logged.</td></tr><tr><td>grey</td><td>A day on which no data has been logged.</td></tr></table> <p>Select a day, enter a time (if necessary), and click OK to apply the selected date to the search field.</p>	Mo	Tu	We	Th	Fr	Sa	Su						1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	Color	Meaning	dark blue border	Today.	dark blue	The selected date.	bright blue	A weekday on which data has been logged.	light blue	A weekend day on which data has been logged.	grey	A day on which no data has been logged.
Mo	Tu	We	Th	Fr	Sa	Su																																																		
					1	2																																																		
3	4	5	6	7	8	9																																																		
10	11	12	13	14	15	16																																																		
17	18	19	20	21	22	23																																																		
24	25	26	27	28	29	30																																																		
Color	Meaning																																																							
dark blue border	Today.																																																							
dark blue	The selected date.																																																							
bright blue	A weekday on which data has been logged.																																																							
light blue	A weekend day on which data has been logged.																																																							
grey	A day on which no data has been logged.																																																							
Logging	date_search	<p>Select the date/time range you want to inspect.</p> <ul style="list-style-type: none">• Upon opening the logging tab, from is set to (now - 10 minutes) if there are log entries existing within this range of time. If not, from will be set to the point of time the first entries are found.• An empty to field displays all log entries from the from date/time until the most recent entries.• An empty from field triggers a search backwards until the first entries are found.• You can remove the time part of the to field to search until the end of the day, and of the from field to search from the beginning of the day.																																																						
Logging	ls_date_filter_settings	<p>Click View to update the displayed logging information.</p> <p>As per default, for logs with a time stamp the log entries are displayed latest first in the search results. Click the tiny arrow in the table header to change the order to oldest first.</p> <p>The date filter settings will be kept as long as your Browser tab is open. They will be reset to default as soon as you open the Logging tab in a new Browser tab.</p> <div><p>If you close your Browser with the Logging tab open, and start your Browser again with restoring all recent tabs (session restore), your date filter settings will be reloaded from your previous search.</p></div>																																																						
Logging	ls_search_for	<p>Insert a string or a valid regular expression to search the log entries for. Only log entries that match the expression will be displayed. Refer to Java Regular Expressions for more information on which regular expressions you can use here.</p> <p>Pressing Enter in this field triggers the search.</p>																																																						

pstate	pstate_control_adapter	<p>The Persistent State Control Adapter gives access to persistent state metadata directly from within a service (self context). The same data can be retrieved using the xUML Runtime API.</p> <div><p>If you want to retrieve metadata of persistent state of the very same service, always use the Persistent State Control Adapter. If you want to retrieve data from other services, use the xUML Runtime API.</p></div>						
PState	pstate_owner	<p>In Load Balancing context, when e.g. running multiple Bridges, you can setup persistent state services to share persistent state objects. The persistent state objects are distinguished by an owner and owner id reflecting the actual service that owns these objects. Prerequisite is that these services share the same persistent state database, see Load Balanced Persistent State for more details.</p>						
PState	pstate_service_transfer	<p>Transfer the service to another Bridge (e.g. by service export and re-deployment).</p> <p>In this case, you also need to transfer file PersistentState.tab from the service directory (<your Bridge data directory>/<service directory>) to the new location. This file contains the actual owner ID of the service. If not present, a new owner ID will be assigned and all objects with the old owner ID will still be stalled.</p>						
PState	pstate_warning_external_attributes	<div><p>Do not switch an attribute between internal and external unless you are sure that no objects of the related class are still pending in the persistent state db. The attribute contents will get lost in this case. This behavior will be improved in future versions.</p></div>						
Service	get_pid	<p>If the service is up and running, you can see the system process id (PID) of the service. To match the PID with the Bridge service, you can also use system commands:</p> <table><tr><th>System</th><th>Command / Output</th></tr><tr><td>Linux</td><td><div><pre>ps -efa grep -- --instance</pre></div><div><pre>... 15228 ... /opt/e2e_bridge_prog/nodejs-8.11.4/linux-64/node index.js --instance /opt/e2e_bridge_data/nodejs_NodeService ... 15265 ... /opt/e2e_bridge_prog/nodejs-8.11.4/linux-64/node app.js --instance /opt/e2e_bridge_data/nodejs_api-test-helloworld_43 ... 15853 ... /opt/e2e_bridge_prog/j2re-11.0.2/linux-64/bin/java -jar repository.jar --instance /opt/e2e_bridge_data/java_helloworld [...]</pre></div></td></tr><tr><td>Windows</td><td><p>Run with administration privileges:</p><div><pre>wmic process where "caption='bridgeserver.exe' or caption='node.exe' or caption='java.exe'" get processid,caption,commandLine /format:csv findstr /C:--instance</pre></div><div><pre>...,bridgeserver.exe,"C:\E2E_BRIDGE_PROG\bridgeserver-2018.12\win32-64\bridgeserver.exe" --config "C:\var\E2E_BRIDGE_DATA\server.cfg" --instance "C:\var\E2E_BRIDGE_DATA\bridge_SoapWait",17600 ...,node.exe,C:\E2E_BRIDGE_PROG\nodejs-8.11.4\win32-64\node app.js "--instance C:\var\E2E_BRIDGE_DATA\nodejs_api-test-helloworld",20144</pre></div></td></tr></table>	System	Command / Output	Linux	<div><pre>ps -efa grep -- --instance</pre></div> <div><pre>... 15228 ... /opt/e2e_bridge_prog/nodejs-8.11.4/linux-64/node index.js --instance /opt/e2e_bridge_data/nodejs_NodeService ... 15265 ... /opt/e2e_bridge_prog/nodejs-8.11.4/linux-64/node app.js --instance /opt/e2e_bridge_data/nodejs_api-test-helloworld_43 ... 15853 ... /opt/e2e_bridge_prog/j2re-11.0.2/linux-64/bin/java -jar repository.jar --instance /opt/e2e_bridge_data/java_helloworld [...]</pre></div>	Windows	<p>Run with administration privileges:</p> <div><pre>wmic process where "caption='bridgeserver.exe' or caption='node.exe' or caption='java.exe'" get processid,caption,commandLine /format:csv findstr /C:--instance</pre></div> <div><pre>...,bridgeserver.exe,"C:\E2E_BRIDGE_PROG\bridgeserver-2018.12\win32-64\bridgeserver.exe" --config "C:\var\E2E_BRIDGE_DATA\server.cfg" --instance "C:\var\E2E_BRIDGE_DATA\bridge_SoapWait",17600 ...,node.exe,C:\E2E_BRIDGE_PROG\nodejs-8.11.4\win32-64\node app.js "--instance C:\var\E2E_BRIDGE_DATA\nodejs_api-test-helloworld",20144</pre></div>
System	Command / Output							
Linux	<div><pre>ps -efa grep -- --instance</pre></div> <div><pre>... 15228 ... /opt/e2e_bridge_prog/nodejs-8.11.4/linux-64/node index.js --instance /opt/e2e_bridge_data/nodejs_NodeService ... 15265 ... /opt/e2e_bridge_prog/nodejs-8.11.4/linux-64/node app.js --instance /opt/e2e_bridge_data/nodejs_api-test-helloworld_43 ... 15853 ... /opt/e2e_bridge_prog/j2re-11.0.2/linux-64/bin/java -jar repository.jar --instance /opt/e2e_bridge_data/java_helloworld [...]</pre></div>							
Windows	<p>Run with administration privileges:</p> <div><pre>wmic process where "caption='bridgeserver.exe' or caption='node.exe' or caption='java.exe'" get processid,caption,commandLine /format:csv findstr /C:--instance</pre></div> <div><pre>...,bridgeserver.exe,"C:\E2E_BRIDGE_PROG\bridgeserver-2018.12\win32-64\bridgeserver.exe" --config "C:\var\E2E_BRIDGE_DATA\server.cfg" --instance "C:\var\E2E_BRIDGE_DATA\bridge_SoapWait",17600 ...,node.exe,C:\E2E_BRIDGE_PROG\nodejs-8.11.4\win32-64\node app.js "--instance C:\var\E2E_BRIDGE_DATA\nodejs_api-test-helloworld",20144</pre></div>							
Service	http_header_roles	<p>Runtime 2020.12 If the standard HTTP header handling does not meet your needs, you can take control of the header handling by defining your own header roles.</p> <p>Refer to HTTP Header Support > Overwriting the Standard HTTP Headers for a detailed explanation of how to use this feature.</p>						
Service	http_headers	<p>Runtime 2019.9 Bridge xUML services read the following incoming HTTP headers containing correlation information:</p> <ul style="list-style-type: none">• X-Transaction-Id or xTransactionId (in JMS context) This header identifies the transaction the call belongs to. You can set the transaction id manually with setTransactionID. If not set, the Runtime will generate one. This header will be passed through the callstack to identify all service calls that belong to a transaction.• X-Request-Id This header should identify the unique request.• X-Sender-Host and X-Sender-Service These headers should contain the sender host resp. the sender service. <p>These headers will be all logged to the transaction log. Having this information, you can use this for error analysis or usage metrics.</p>						

Service	http_headers_adapter	<p>Runtime 2019.9 With xUML service adapter calls, the xUML Runtime adds the following outgoing HTTP headers containing correlation information to the request:</p> <ul style="list-style-type: none"> • X-Transaction-Id or xTransactionId (in JMS context) This header identifies the transaction the call belongs to. You can set the transaction id manually with setTransactionID. If not set, the Runtime will generate one. This header will be passed through the callstack to identify all service calls that belong to a transaction. • X-Request-Id This header identifies the unique request. The Runtime generates a unique number for each adapter call. • X-Sender-Host and X-Sender-Service These headers contain the sender host resp. the sender service. They are set by the Runtime automatically. <p>Transaction id and request id will be logged to the transaction log on the adapter call. Having this information, you can use this for error analysis or usage metrics.</p>
Service	httpHeaderMap	<p>Runtime 2020.11 Header information as a map. The map contains arrays of header value strings whereas the header name is the key of the map.</p> <ul style="list-style-type: none"> • Header names are lowercase and treated case insensitive. • Multiple headers with the same name are treated as arrays. <p>Refer to HTTP Header Support for more information on the standard xUML HTTP headers.</p>
Service	httpHeaders_depr	<p>DeprecatedThis attribute is deprecated as of Runtime 2020.11. Please use httpHeaderMap (see below) for new implementations as its implementation complies to the HTTP specification.</p>
Service	kill_service	<p>Bridge 7.2.0 The Kill functionality will first try to regularly stop the service. When the service is still running after 10 seconds, it will forcibly terminate the service with <code>Java Process.destroyForcibly()</code>.</p>
Service	max_request_body_size	<p>Runtime 2021.2 Specifies the maximum size of the request in KB (1 KB = 1024 Bytes). This can be used to prevent DoS or similar attacks. When the payload of the service exceeds the given maximum, incoming request are rejected.</p>
Service	max_request_header_size	<p>Runtime 2022.6 Specifies the maximum size of the request header in KB (1 KB = 1024 Bytes). This can be used to prevent DoS or similar attacks. When the header payload of the service exceeds the given maximum, incoming request are rejected.</p> <div>  Compatibility Hint For older Runtimes, a limit of 8 KB applies. </div>
Service	stop_service	<p>Bridge 7.2.0 Clicking Stop will send <code>SIGTERM</code> to the service. Older Bridges called <code>Java Process.destroy()</code>.</p>
Service	transaction_id	<p>The Transaction ID identifies a transaction. It is a unique number used to trace service calls through the call stack of multiple service calls.</p> <ul style="list-style-type: none"> • Runtime 2019.9 Clients calling a service running on the Bridge can provide a transaction ID in HTTP header X-Transaction-Id or xTransactionId (in JMS context). • SOAP clients can also use the SOAP headers to provide a transaction ID. • If an xUML service is called without providing a transaction ID, the xUML Runtime will generate such an ID. <p>This ID will be passed on through the call stack of the xUML service, so that the whole transaction can be traced. This can be useful, when analyzing the log file in case of error.</p>
Service Preferences	automatic_restart	<p>Whenever the service crashes, it will restart immediately. Nevertheless, in the navigation the icon  next to the xUML service name will indicate the abnormal termination.</p>
Service Preferences	automatic_startup	<p>Select this option, if you want the service to startup automatically, whenever the Bridge is started. Only users who are member of a group, to which the role ADMIN has been assigned, are allowed to change this option.</p> <div> <p>This option can be globally disabled by the Disable Automatic Service Startup option on the node instance preferences.</p> <p>When updating the BRIDGE, all deployed services (xUML, Node.js, and Java services) will be kept. However, the automatic startup option will be ignored on the very first start-up after the update.</p> </div>

Service Preferences	minimum_uptime	<p>To allow the Bridge to distinct whether the service has crashed during start-up or not, specify the minimum uptime of the Node.js service in seconds.</p> <p>Implications:</p> <ul style="list-style-type: none"> • If the service crashes during the uptime period, the Bridge will assume that the service could not be started. No Automatic Restart will be applied. • If the service crashes after the uptime period, the Bridge will assume a crash. If option Automatic Restart is set, the Bridge will try to restart the service.
Service Preferences	owner	The group id of the user who has deployed the service. Only users who are member of a group, to which the role ADMIN has been assigned, are allowed to modify the owner of the service.
Update	service_startup_after_update	When updating the BRIDGE, all deployed services (xUML, Node.js, and Java services) will be kept. However, the automatic startup option will be ignored on the very first start-up after the update.