

DML structure for optical components.

Schema fragment	Explanation
<pre> OXCEdgeRouter [_extends .schemas.Net.router failure %S srg %S] OXC [...] </pre>	<p>OXCEdgeRouter : defines a router that can connect optical networks to non-optical networks. The OXCEdgeRouter extends the component router.</p> <p>Optional attribute failure used to indicate if the OXC is failed (default value: false).</p> <p>Optional attribute srg : The Shared Risk Group. This attribute is an integer list that represents all SRG this node share. The integers are separated by comma (default value: null).</p> <p>The OXC is defined like the OXCEdgeRouter except that it cannot accept non-optical interfaces.</p>
<pre> interface [extend .schemas.Net.NIC (of SSFNet) inputMonitor debug %S DEBUG %S FRM [use %S]] </pre>	<p>GLASS framework provides an extended network interface card, which includes some features like, failure detection, input traffic monitoring.</p> <p>Optional attribute inputMonitor specifies the monitor to use for input traffic. This feature has been added to compute the number of data lost during a failure.</p> <p>Optional attribute debug allows additional information to be printed (default value: false) for this ONIC.</p> <p>Optional attribute DEBUG allows additional information to be printed for all ONICs (default value: false).</p> <p>Optional attribute FRM configure the failure module that will be notified a failure/recovery events. The use attribute represents the class name.</p>
<pre> onic [extend .schemas.Net.NIC (of GLASS) ber %I1 delay %I1 jitter %I1] </pre>	<p>Optical Interface (ONIC) extend the class _NIC which is the extended interface provided by GLASS</p> <p>The default queue is the BasicQueue from the package gov.nist.antd.optical</p> <p>Optional attribute ber represents the bit error rate (default value: 1E-9).</p> <p>Optional attribute delay is the delay for this ONIC (default value: 100 microseconds).</p> <p>Optional attribute jitter the distortion value for the lambdas used by this ONIC (default value: 0).</p>
<pre> link [_extend .schemas.Net.link (of SSFNet) id %I1! failure %S] </pre>	<p>The GLASS framework also provides an extended link that include failure capability and notification.</p> <p>Optional attribute id represents the id of the link to make easier the search of a link during the simulation (default value: -1).</p> <p>failure: indicates if the link is failed (default value: false)</p>

```

OpticalLink [
  extend .schemas.Net.link (of GLASS)
  id %I!
  ber %I
  distance %I!
  noAmplifiers %I
  noRegenerators %I
  protection %S
  protectionMode %S
  failure %S
  srlg %S
]

```

OpticalLink

An optical link extends the class link. This link must have an integer ID unique in the network.

Optional attributes:

ber: bit error rate. Default value=0

distance: length of the link in kilometers.

nbAmplifier: number of amplifiers on the optical link

nbRegenerator: number of regenerators on the optical link

protection: indicates if the link is used as a protection link. Default value: False

protectionMode: defines the mode (NEVER, ONLY, SHARED) associated to the attribute protection

ONLY: The algorithm is not able to change the protection attribute, the value is "true". The component is not allowed to be used in a non-backup scenario.

SHARED: The algorithm is able to change the protection attribute back and forth. The component can be used for everything.

NEVER: The algorithm is not able to change the protection attribute, the attribute is "false" for ever, the component is not allowed to be used in a backup setup.

The default values are "protection false" and "protectionMode SHARED".

srlg: The Shared Risk Link Group. This attribute is an integer list that represents all SRLG this link share. The srlg attribute is a list of integer separated by comma.

The user does not have to include fibers in the configuration of an optical link. If so, we automatically create 1 unidirectional fiber with 10 lambdas (included 1 control lambda) in both directions.

<pre> Fiber [#general attributes id %II! failure %S protection %S1! protectionMode %S1! bandwidth %II! noLambdas %S1! controlLambda %S1! startWavelength %II! deltaWavelength %II!] Fiber [#attributes for a bidirectional fiber bidirectional true host2Id %II! host1PortId %II! host2PortId %II!] Fiber [#attributes for a unidirectional fiber bidirectional false receiverHostId %II! senderPortId %II! receiverPortId %II!] </pre>	<p>Fiber: defines the structure of a fiber.</p> <p>The ID of a fiber must be unique in an ONIC.</p> <p>Optional attribute failure indicates if the fiber is failed (default value: false). The optional attribute protection shows if the fiber is used to protect another fiber.</p> <p>protectionMode : defines the mode (NEVER, ONLY, SHARED) associated to the attribute protection. See the information of these attributes in the OpticalLink.</p> <p>Optional attributes bandwidth, noLambdas and controlLambda are used when the user does not care about the specification of each lambda. The attribute bandwidth defines the bandwidth of the fiber. The attribute controlLambda specifies if control lambdas must be generated (if true, 1 for unidirectional fiber and 2 for bidirectional fiber). If no lambda is defined by the user and these attributes are not used, a configuration of 5 lambdas included control lambdas will be created. If the attribute bandwidth is used, noLambdas is required but not the attribute controlLambda (default value : true).</p> <p>Optional attributes startWavelength and deltaWavelength are used to define the first wavelength to use and the space between 2 wavelengths when lambdas are automatically generated and to check the user-defined wavelength of the lambdas. (Default value 1550 nm for startWavelength and 0.8nm for deltaWavelength).</p> <p>If Optional attribute bidirectional is false, use the attribute receiverHostId to specify the direction of the fiber. If it is true, the user has to specify the attribute host2Id to be able to assign the good port Id of each interface (default value: false).</p> <p>The attribute host1PortId can be used in lieu of senderPortId and host2PortId in lieu of receiverPortId. The portIds must be unique in a node.</p>
---	--

<pre> Lambda [id %II! wavelength %II! bandwidth %II! control %S1! snr %I failure %S protection %S protectionMode %S1! pair %S1] </pre>	<p>Lambda : defines the lambda's characteristics</p> <p>The ID of a lambda is unique inside a fiber.</p> <p>Optional attribute wavelength defines the wavelength of the lambda. If not specified, the system will set the attribute automatically according to the value of startWavelength and deltaWavelength of the fiber.</p> <p>Optional attribute bandwidth defines the bandwidth of a lambda (default value: 2.5 Gbps).</p> <p>Optional attribute control indicates if this lambda is used for control messages (default value: false).</p> <p>Optional attribute snr is the Signal to Noise Ratio (default value: 0).</p> <p>Optional attribute failure indicates if the lambda is failed (default value: false).</p> <p>If the optional attribute protection is true, it means that this lambda can be used for protection purposes.</p> <p>protectionMode defines the mode (NEVER, ONLY, SHARED) associated to the attribute protection. See the information of these attributes in the OpticalLink.</p> <p>The optional attribute pair defines the pair lambda. The lambda is specified by the couple fiberId (lambdaId). It means the lambda is in the same link (default null).</p>
<pre> AlgorithmContainer [algorithm [name %S1! use %S1!]] </pre>	<p>Algorithm Container: defines the list of centralized algorithms to use in the network.</p> <p>Attribute centralized defines an algorithm to use into the OXC. An algorithm container can contain zero or more centralized algorithms. Each centralized algorithm implements a method config to include parameters.</p>
<pre> connection [id %II static false destination %II routing %S1 wavelength %S1 rwa %S1 bandwidth %II delay %II srlg %S1] </pre>	<p>Dynamic connection request.</p> <p>id: the id of the connection. Must be systemwide unique.</p> <p>destination: the id of the destination.</p> <p>routing: the name of the centralized routing algorithm. The connection will be store in the path container that has this name.</p> <p>wavelength: the name of the centralized wavelength assignment algorithm.</p> <p>rwa : the name of the routing and wavelength algorithm (done in one call).</p> <p>bandwidth: defines the QoS for the connection.</p> <p>delay: optional attribute for the connection request (depending on the algorithms used).</p> <p>srlg : the list of SRLG that must not be included in the path.</p>

<pre> connection [id %I static true node %I#source node %I node %I #destination channel [segment [port %I lambda %I]]] </pre>	<p>Static connection request.</p> <p>id: the id of the connection. Must be systemwide unique. node: the list of the node crossed by the connection channel: the list of lambda for one channel. The segments are identified by the port number and lambda id. We can have multiple channels for one connection.</p>
<pre> connection [id %I static true node %I#source node %I node %I #destination wavelength %S bandwidth %I] </pre>	<p>Static connection request</p> <p>The route is user-defined but an algorithm according to the bandwidth does the wavelength assignment.</p>
<pre> event [type %S time %I use %S] </pre>	<p>The scripted events are defined as follow: Attribute type defines the object modified by the event (node, fiber, link...). Attribute time is the time in second when the event will be fired.</p> <p>Optional attribute use indicates which event class to use. If it is not indicated then it will use the default events provided by nist.</p> <p>Other attributes are required depending on the type of the event and the event class used. For example, if the attribute use is not used and the type is node, then the DML must contain the attribute id that indicates the node's id. And also, the kind of modification (like failure true).</p>
<pre> ProtocolSession [name oxswitch use gov.nist.antd.optical.OXCswitch converter %S noConverters %I concatenation %S noAddDrop %I] </pre>	<p>OxcSwitch: this class defines the switching capabilities of the OXC.</p> <p>Optional attribute converter indicates if the OXCEdgeRouter has lambda conversion capability (default value: true). Optional attribute noConverters indicates the number of converters included in switch (default value = infinite). Optional attribute concatenation defines the type of concatenation available for waveband switching. Three types are defined: none, standard (lambdas must be contiguous), virtual (lambdas don't necessary be contiguous). Optional attribute noAddDrop specifies the number of AddDropPort in the switch. These ports are used to connect other protocols on top of the OXCSwitch (default value: max between the inLambdas and</p>

	outLambdas).
<pre>ProtocolSession [name trafficManager use gov.nist.antd.merlin.generator.traffic.TrafficManager generator [use ...]]</pre>	<p>DML fragment for the traffic manager.</p> <p>Attributes name and use are the same as any ProtocolSession.</p> <p>Attribute generator defines which module will be used to generate connection requests.</p>
<pre>ProtocolSession [name topologyManipulator use gov.nist.antd.merlin.generator.event.TopologyManipulator generator [use ...]]</pre>	<p>DML fragment for the topology manipulator.</p> <p>Attributes name and use are the same as any ProtocolSession.</p> <p>Attribute generator defines which module will be used to generate events.</p>
<pre>ProtocolSession [name simpleProtocol use gov.nist.antd.merlin.protocol.sample.SimpleProtocol destination %I1 routing %S1 wavelength %S1 bandwidth %S1]</pre>	<p>DML fragment for the SimpleProtocol</p> <p>Attributes name and use are the same as any ProtocolSession.</p> <p>Optional attribute destination specifies the destination node of the messages (default value: -1). The default value will allow the node to receive messages but not to send.</p> <p>Optional attribute routing specifies the name of the routing protocol (default value: shortestPathDistance).</p> <p>Optional attribute wavelength specifies the name of the wavelength algorithm (default value: bestFist).</p> <p>Optional attribute bandwidth specifies the bandwidth request for the connection (default value: 2.5 Gbps).</p>
<pre>ProtocolSession [name simpleOpticalNeighbourDiscovery use gov.nist.antd.merlin.protocol.discovery.OptNeighbour addLambda [id %I1 onicId %I1 fiberId %I1 lambdaId %I1] dropLambda [id %I1 onicId %I1 fiberId %I1 lambdaId %I1]]</pre>	<p>DML fragment for the simple optical neighbour discovery.</p> <p>Attributes name and use are the same as any ProtocolSession.</p> <p>This protocol extend the classes AutoConfigCtrl who connects automatically the control lambdas of the fibers if the user does not precise them.</p> <p>The user can specify its own connections by using the optional attributes addLambda and dropLambda.</p>
<pre>ProtocolSession [name backup use gov.nist.antd.merlin.protocol.protectionlink.BackupManager BACKUPALGORITHM %S1]</pre>	<p>DML fragment for the protocol session BackupManager.</p> <p>Attributes name and use are the same as any ProtocolSession.</p> <p>Attribute BACKUPALGORITHM specifies the name of the algorithm to be called for the restoration. The same algorithm will be used for all the BackupManager so the attribute should be specified only once.</p>

<pre> ProtocolSession [name DynRecoveryUDPIP Use gov.nist.antd.merlin.protocol.signaling.DynRecoveryUDPIP debug %S1 messages %S1 BCKUPALGO %S1] </pre>	<p>DML fragment for the protocol session BackupManager.</p> <p>Attributes name and use are the same as any ProtocolSession. Optional attribute debug is used to switch to debug mode (default value: false). Optional attribute messages indicates if message content should be displayed (default value: false).</p> <p>Attribute BCKUPALGO specifies the name of the algorithm to be called for the restoration. The same algorithm will be used for all the BackupManager so the attribute should be specified only once.</p>
--	--

NOTE about the nodes:

GLASS-TSC is also using extra attributes from the nodes dml configuration that represent their position in the gui. When creating a topology with the GLASS-TSC, 4 attributes are included in each node:

- x and y, use for the position of the node in the topology panel
- xCon and yCon, use for the position of the node in the connection panel.

For more information about the GLASS-TSC, look at the document (GMPLS Agile Switching Simulator- Topology and Simulation Creator).