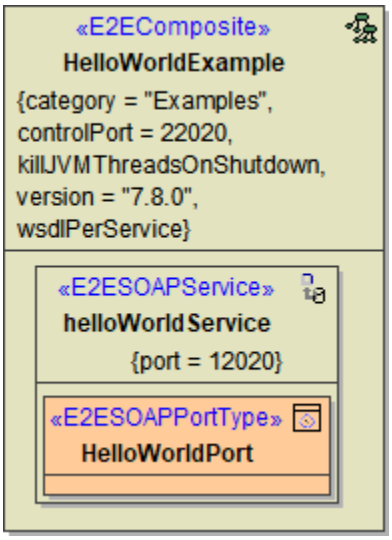
Frontend Components

In front of xUML services we have the service client followed by the service itself. Thus, components serving these clients are called frontend components or frontend services. These frontend services can be further grouped into a logical deployment unit called composite. This is done in the component diagram.

The most simple component diagram is depicted below. It contains a <<E2EComposite>> component. This component is the deployment unit of a set of service components. Each <<E2EComposite>> component may contain several services. The different service types are described in the Service Implementations section, e.g. SOAP, HTTP, JMS, Java, Timer, Scheduler, SAPRFC, etc. Finally, each service contains at least one class realizing the service. For example, SOAP services contain SOAP port types.

This is the component diagram of the **HelloWorldExample**.



Each composite manifests itself as repository file after compilation. This means, after compilation the **<projectPath>/repository** folder contains the repository file **HelloWorldExample.rep**. Additionally, each of the logical components can be configured by the use of tagged values. This repository belongs to category **Examples** and uses a control port **22020**. For further composite attributes see the table below.

> After deployment, each composite is started as operating system process. More details about this architecture can be found in Bridge Architecture.

<<E2EComposite>> Attributes

The composite holds the following tagged values:

Attribute	Type	Description	Allowed Values / Example
Name	String	Name of the composite.	`HelloWorldExample`
Version (version)	String	A service version number. This service version is visible in service context on the Bridge.	`7.8.0`
ControlPort (controlPort)	Integer	Port used by the Bridge to control this xUML service.	`22020`
Category (category)	String	Optional category to group similar xUML services.	`Examples`
Node Type (nodeType)	String	Not used, leave empty.	

On this Page:

- <<E2EComposite>> Attributes

Related Pages:

- Bridge Architecture
- Service Implementations
- Modeling the Java Components
- Persistent State Components
- Using Testable Classes

External Test Proxy Host (externalTestProxyHost)	String	Specifies the host name as seen by the client. • If a server certificate is used, the certificate must be issued for this name. • If this tagged value is not set, the name of the node hosting the proxy is being used.	`scheer-acme.com`

Class To XML Default Root Name (classToXMLDefaultRootName)	String	Bridge 7 Specify which name to assign to the XML root element upon serializing. This setting can be overridden by using XML composer options as described on classToXML() Operation. Refer to XML - UML Class Mapping for more information on the topic of XML serialization.	**Default**	Try to use name and namespace defined on the class by the <<XML>> stereotype. Fallback to **VariableName** if not provided (default).

Type Name	Use static name and namespace of the class as name of XML root element.

			Variable Name	Use the name of the reference (object/variable) as name of XML root element.
HTTP				

Http Header Roles (httpHeaderRoles)	Array of String	Builder 7.12.0 Runtime 2020.12 In the context of HTTP based services (HTTP, REST, SOAP), assign roles to dedicated incoming headers that define how the related header will be treated by the xUML Runtime. 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. Refer to HTTP Header Support > Overwriting the Standard HTTP Headers for more details. **Http Header Roles** can hold a list of definitions in format `<http header name>:<role>`, where `<role>` can be one of the listed allowed values (one list entry per line).	client_host	Take the sender host from header <http header name> instead of **X-Sender-Host**.

client_service	Take the sender service from header <http header name> instead of **X-Sender-Service**.

correlation_id	Take the correlationID from header <http header name> instead of **X-Request-Id**.

			transaction_id	Take the transaction ID from header <http header name> instead of **X-Transaction-Id**.

Request Http Header Roles (requestHttpHeaderRoles)	Array of String	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. **requestHttpHeaderRoles** can hold a list of definitions in format `<http header name>:<role>`, that will automatically be generated for each adapter call inside this service composite. `<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.	client_host	Provide the client host in a header <http header name> instead of **X-Sender-Host**.

				client_service	Provide the client service in a header <http header name> instead of **X-Sender-Service**.

correlation_id	Provide the correlationID in a header <http header name> instead of **X-Request-Id**.

			transaction_id	Provide the transaction ID in a header <http header name> instead of **X-Transaction-Id**.

			passthrough	Pass a present header <http header name> to the called service.
			passthrough= <request header name>	Pass an incoming header <request header name> to the called ser

vice under the name of <http-header-name>. This is equivalent to renaming a header.

JVM				
Read Modeling the Java Components for more information on these tagged values.				
Persistent State				
Read Persistent State Components for more information on these tagged values.				
SAP				
SAP Default Connection Pool Size (sapDefaultConnectionPoolSize)	Integer	Default capacity of a single SAP connection pool (Bridge acting as a SAP client). If undefined, a default of 10 connections will be applied. Each distinct connection to a SAP system has its own pool. Connections are distinguished by the set of connection parameters (connection string). You can override the connection pool size for a specific connection on the corresponding SAP alias. On using dynamic SAP access, the default connection pool size is used.	a valid integer, default is **10**	
SAP Padding (sapPadding)	String	Service-wide setting for SAP values padding. This setting will be applied to all IDoc and SAP adapters within the service. It is not recommended to use **Mixed** padding. This option is only available for reasons of backwards compatibility. Mixed padding is default for older services that have been compiled before the implementation of this tagged value, whereas **Never** is default, if no SAP padding is specified.	**Never**	No padding, removes existing padding (default).
			Always	Always pads to specified length.

			Mixed	Padding only fields that are not within deep structures.
SAP Server Worker Threads (sapServerWorkerThreads)	Integer	Number of parallel request (workers) the Bridge (acting as an RFC server) can process. If this value is undefined, the Bridge will only process one request at a time (equivalent to **sapServerWorkerThreads**=1). Each active worker requires one license slot (concurrent connection). For more information on licensing and concurrent connections, refer to License for Running xUML Services.	a valid integer, default is **1**	
Startup/Shutdown				
Startup Shutdown Trace Port (startupShutdownTracePort)	Integer	Default port for tracing startup or shutdown activities is **30000**. You can change this default here, if necessary.		
Startup Activity (startupActivity)	Reference to Activity	The referenced activity is called while starting up *before* any other component gets invoked - including timers and schedulers.		
Shutdown Activity (shutdownActivity)	Reference to Activity	The referenced activity is called when the xUML Runtime is being shutdown.		

Startup Must Succeed (startupMustSucceed)	Boolean	Runtime 2021.9 The service does not start if the implemented startup activity fails. Older Runtimes than 2021.9 will not start if this flag is set.	true	The service does not start if the startup activity fails.
			false	The service starts although the startup activity fails (default).
Test				
Read Using Testable Classes for more information on these tagged values.				
WSDL				
WSDL Per Service (wsdlPerService)	Boolean	If true (default=false), each xUML service gets its own WSDL file. Additionally, all XML Schema elements and types having the same namespace are put into one schema file. These schema files are imported into the WSDLs to be shared among them. In this case it is also possible to mix RPC/soap-encoded services with Document/literal services.		
WSDL Namespace (wsdlNamespace)	String	Target WSDL namespace of the generated WSDL file. Relevant only, if **wsdlPerService** is false (this is the default).		

Soap Version (soapVersion)	String	Specify the version of the SOAP protocol you want to use with the service.	**1.1**	SOAP version 1.1 (default)
			1.2	SOAP version 1.2
Resolve Inheritance (resolveInheritance)	Boolean	If true, the inheritance hierarchy is resolved into flat messages. As of Bridge 7, setting **resolveInheritance** to true is deprecated, because this will generate a different output structure than modeled. It also has hidden requirements to the element uniqueness.	false	Keep inheritance hierarchy. (Bridge 7 default)

			true	Resolve inheritance hierarchy into flat messages. (before Bridge 7 default)

The component diagrams, the composites, and services artifacts are always found in the same place in all UML models:

Figure: Component View in the Containment Tree