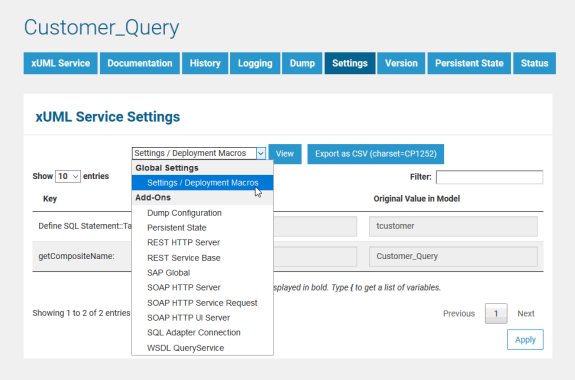
xUML Service Settings

The Bridge provides the flexibility to directly modify xUML service settings that have been defined with the Builder in component or activity diagrams without re-deploying the xUML service. After the first deployment of the xUML service, the settings can be overwritten on the **Settings** tab.

Switch to the **Settings** tab in the information/working area.
Users with administration rights are allowed to view and modify the xUML service settings of any xUML service. Users who are member of a group, to which the role **MODELER** has been assigned, are only allowed to view and modify them, if they themselves or a member of the same group deployed the xUML service.

> Users who do not have the permission to view the settings cannot see the **Settings** tab at all. This prevents unauthorized users accessing sensitive information like passwords, etc.



The settings are grouped. Initially, the Bridge displays the settings of the first settings group **Settings /Deployment Macros**.
Select the group of settings you want to change.

The settings are categorized into the following:

- **Global Settings**
  In this category, you can change setting values that are global to the xUML service. Deployment information retrieved with deployment macros can be overwritten in this category, too (see Global Settings below).
- **Add-ons**
  In this category, the settings comprise add-on related values like tagged values, interface URIs, and others. For instance, data of the SQL adapter, File System adapter, Timer, or SOAP service request may be overwritten. For more details, refer to the example in Add-on Settings.

Within a settings category, select a settings group you want to modify settings in.

Change settings and click **Apply** to apply your changes. Click **View** to refresh the working area. Click **Export as CSV** to export all settings to an Excel sheet.

> You can only modify settings of xUML services, which have been stopped.
>
> Once you have overwritten values of an xUML service on this page, they will be used permanently even when re-deploying the configuration with updated values. Setting values on this page will always overrule values that will be deployed with the xUML service afterwards. This rule also applies when updating or reinstalling the Bridge (by keeping the deployed xUML service). This does **not** apply if you remove the xUML service first and redeploy it again.

Global Settings

> If you want to define settings that can be used in multiple services, refer to Defining Overall Settings for Multiple E2E Services.

Settings Group	Setting	Description	Allowed Values

On this Page:

Related Pages:

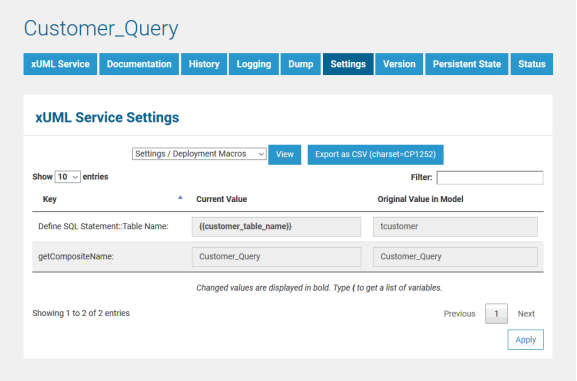
- Using Global Setting Variables
- Exporting Settings

Settings / Deployment Macros	**Service Composite**			
	getCompositeCategory	Value returned by xUML Action Language macro getCompositeCategory(). Initial value of this setting is the composite category from the xUML model (see Frontend Components). You can change this value here.		any string
	getCompositeName	Value returned by xUML Action Language macro getCompositeName(). Initial value of this setting is the composite name from the xUML model (see Frontend Components). You can change this value here.		any string
	getCompositeVersion	Value returned by xUML Action Language macro getCompositeVersion(). Initial value of this setting is the composite version from the xUML model (see Frontend Components). You can change this value here.		a version string
	PAS Platform			
	AuthenticatorEnabled	Specify whether users are allowed to identify themselves using an **x-pas-user** header instead of a **bearer token**. For compatibility reasons **true** (x-pas-header allowed) is the default option but this is deprecated and may lead to security issues. If you do not rely on the **x-pas-header**, we recommend setting this option to **false**.	**true**	Allow using an **x-pas-header** or a bearer token for authentication (default).
			false	Authentication only by **bearer token**.
	AuthService::minimalAccessTokenLifeSpan	Define when a refresh token should be triggered. Default is **30** (if the token last less than 30 seconds it will be refreshed before it is used). In general there is no need to change the default of 30 seconds.		any integer
	KeycloakInstance::clientId	Specify the clientId to use when authenticating the service. Default is **keycloak-clientId**. The clientId is defined in Keycloak, it should be changed in production to define specific authorization for this service.		any string
	KeycloakInstance::clientSecret	Specify the clientSecret associated to the clientId. Default is **keycloak-clientSecret**. The clientSecret is defined in Keycloak, it should be changed in production to define specific authorization for this service.		any string
	KeycloakAlias:Location:host	Specify the hostname (domain) of the Keycloak SSO system. Default is **keycloak-host**.		any string
	KeycloakAlias:Location:basePath	Specify the subpath of the Keycloak system. Default is **keycloak-basePath**.		any string
	KeycloakAlias:Location:port	Specify the port of the Keycloak system. Default cannot be changed.		8080
	KeycloakAlias:Location:protocol	Specify the protocol of the Keycloak system. Default cannot be changed.		http
	PasSecurityService::allowAnonymous	Specify whether anonymous users (no PAS users) should be allowed to send requests to the PAS BPMN service. In contrast to anonymous users, PAS users are identified by a bearer token or x-pas-user header). If this setting is set to **false**, requests of unauthenticated users will return HTTP error 401.	**true**	Allow anonymous access (default).

		This setting does not disable any role configuration within the BPMN. If there are lanes configured in the BPMN, the service can only be accessed by authenticated users.	false	Access for identified users only.
	Service			
	<service settings>	In your xUML model, you can define name-value pairs with the setting macro in an activity diagram, or initial values of class attributes having the tagged value **setting** set to **true**. Having defined settings in the xUML model like that, you can modify these values on the Bridge here. Refer to xUML Service Settings below for more information.		

xUML Service Settings

You can define name-value pairs with the setting macro in an activity diagram, or initial values of class attributes having the tagged value **setting** set to **true**. Having defined settings in the xUML model like that, you can modify these values on the Bridge here.



Click **Apply** after you have changed the values.
Changed values are displayed in bold. The original values coming from the UML model are displayed in a separate column for you to compare the original value and the changed value.

Refer to the example mentioned below for more details.

Example File (Builder project E2E Action Language/Operating):

<your example path>\E2E Action Language\Operating\uml\settings.xml

You can only modify settings of xUML services that have been stopped.

Add-on Settings

Each xUML service adapter has its own settings that are initialized in the xUML model and can be changed here. The general context of the add-on settings is explained with the SQL ODBC example.

Example File (Builder projectAdd-ons/SQL):

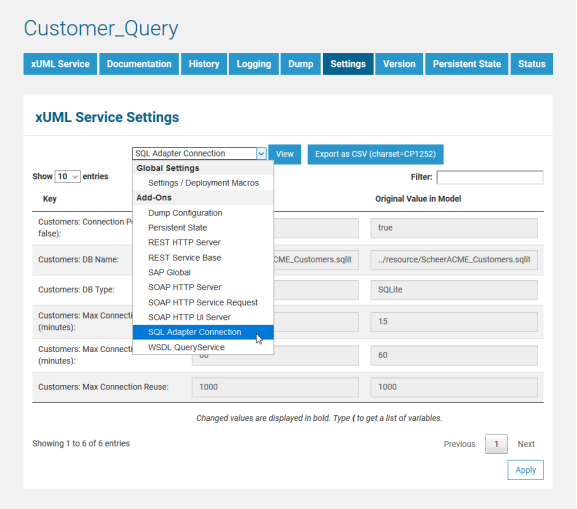
<your example path>\Add-ons\SQL\uml\sqlOdbc.xml

If you are running an xUML service that is connecting to a database backend, you can modify the tagged values of the database interface respectively dependency (see picture below). In this example, the key values displayed on this page correspond to the tagged values that have been defined in the UML model.

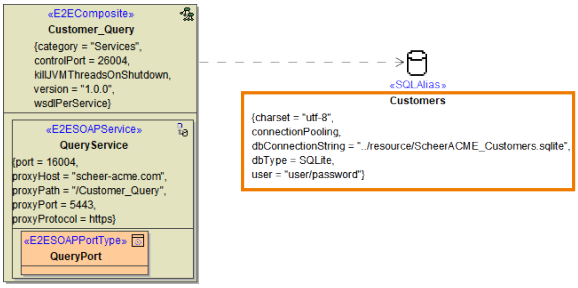
Before starting a deployed xUML service that connects to a database backend, you may want to redefine required database parameters on the Bridge.

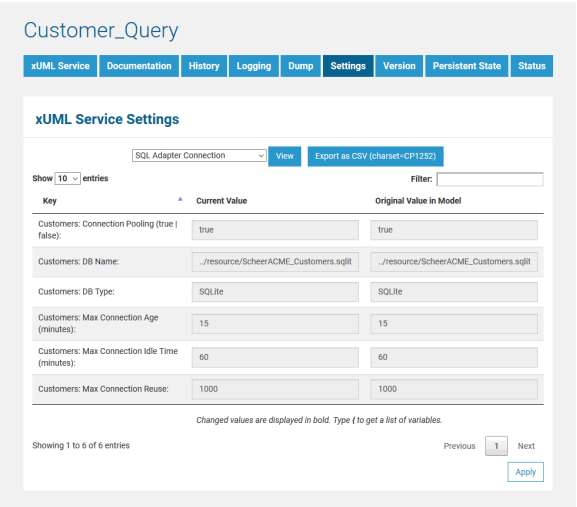
SQL Adapter Connection Settings

Select the xUML service in the navigation on the left (in this example **CustomerQuery**). Switch to tab **Set tings** and select the option **SQL Adapter Connection** in the **Add-ons** category from the drop-down list.



The default parameters are defined in the component diagram of the xUML service (see example **ODBCExample** below). The connection is defined by the xUML named **ODBCExample** and the SQL Alias named **customers**. The database user and password are defined in the tagged value **user** on the alias.





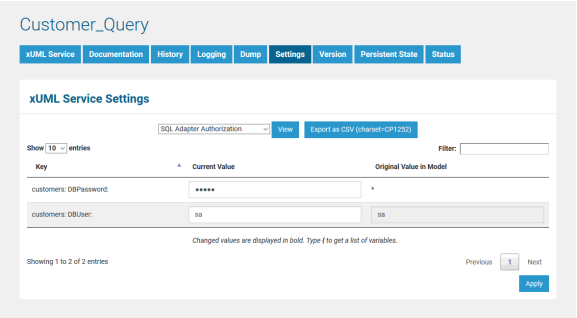
Key	Value
customers: DBType	Name of the SQL Service component, e.g. **SQLite**
customers: DBName	Name of the database, e.g. **ScheerACME_Customers**
customers: Connection Pooling	Added in Builder 5.1.8.58 Runtime 5.1.82.0 This tagged value controls the connection pooling. If true, each connection is put into a pool after use. If an SQL adapter requires a connection, it is taken from the pool. If no connection is available, a new connection is being created and put into the pool after use. The time the connection is kept in the pool depends on the other pooling parameters.
customers: Max Connection Reuse	This tagged value controls how often a connection can be re-used. After the connection has been re-used for **maxConnectionReuse**, it will be closed and not put back into the pool. This feature has been introduced because some databases had problems if the connection was re-used too often. Value **-1** means the connection will be re-used forever. In this case you should define reasonable values for **maxConnectionAge** or **maxConnectionIdleTime** (see above). Note that the pooling is implicitly switched off, if **maxConnectionReuse** is set to 0.
customers: Max Connection Age	After a given connection age (in minutes) the connection will be closed and removed from the pool.
customers: Max Connection Idle Time	Connections not used for the time specified (in minutes) will be closed and removed from the pool. This is useful for connections going through firewalls because such connections might be cut off after some time.

For more information on the SQL adapter settings (other tagged values, default values, ...) refer to SQL Adapter Reference.

For each SQL adapter alias found in the activity diagrams of a UML model, you will find the SQL adapter connection settings as described above.

SQL Adapter Authorization Settings

Now, select **SQL Adapter Authorization** from the list.



These settings allow you to adapt the SQL database user and password of the xUML service.

Key	Value
customers: DBUser	Database user
customers: DBPassword	Database password

Once you have overwritten values of an xUML service's deployment on this page, they will be used permanently even when re-deploying the service with updated values. Setting values on this page will always overrule values that are deployed with xUML services later. This does not apply, if you remove the service before deploying it again.

Overview on other Add-on Settings

Generally, most of the add-on settings that are related to an xUML Service Adapter have a corresponding tagged value in the component diagram (as described above with help of the SQL ODBC example). Have a look at the documentation pages of the corresponding adapter for these settings.

Find below a list of other add-on settings and their description. For the sake of completeness, we also mentioned settings coming from the model for some setting groups (see column **Specified in Model on**).

Settings Group	Setting	Specified in Model on	Description
Dump Configuration	Caught Error Code		Only dump errors wi This setting takes
	Caught Error Domain		Only dump errors of This setting takes
	Dump Caught Errors: enabled		Enable/Disable writi This setting takes
	Dump not Caught Errors: enabled		Enable/Disable writi Service).
Persistent State	Owner		Change the name of subsequently create All existing persis
	Worker Limit	Composite	Specify the worker li **Workers** defines the configured. The impl Each active connection
REST Service HTTP Server	Descriptor: Cache Control: value		Specify the Browser For more informatio
	Test Tool: Cache Control: value		Specify the Browser For more informatio
REST Service Base	<your service name>: enabled		Switch the service p This setting corresp

	<your service name>:JSONComposerOptions: compact	REST Service	Specify the JSON co When **jsonCompac** compile of an older r
	<your service name>: JSONComposerOptions: keepNulls	REST Service	Specify the JSON co When **jsonKeepNul** completely (see also
	<your service name>: MaximumConnections:		Change the maximu
	<your service name>: Port		Change the port the
	<your service name>: ResolveHostnames		Define whether the h
SOAP HTTP Server	Allow Tracing		Switching tracing on All xUML services ar service is called.
	Maximum Connections		Change the maximu
	ResolveHostnames		Define whether the h
	<your service name>Port		Change the port the
SOAP HTTP Server Request	<your service name>: <your port type>: enabled		Switch the service p This setting corresp
	<your service name>: <your port type>: URI	Component diagram	Change the service
SOAP HTTP UI Server	Library Cache Control: value		Specify the Browser For more information
WSDL: <your service name>	WSDL: <your service name> host		Specify the host nam