Lifecycle Governance

This section describes how the SOA Gateway web services can be governed in terms of their availability, their versions and their status. This concept is referred to as “Lifecycle Governance” – the governance of SOA Gateway web services throughout their life time.

The availability (or unavailability) of a web service is something that an organization may wish to set for a period of time. For example, if there is maintenance required on a specific database table, the web service which uses this table could be made unavailable without affecting other services. In this case, clients will receive a specific SOA Gateway message to indicating the service is unavailable.

The versioning of web services allows similar services to co-exist, transparent to existing clients, while allowing new clients to take advantage of the latest service modifications. For example, what happens when the type of a parameter needs to change from xs:string to xs:int? This change must be coordinated so that existing clients of the web service can continue unaffected, and ensuring that new clients receive the most up-to-date implementation. SOA Gateway can make this possible seamlessly.

The status of a service is a concept where the service is in the process of being migrated from different states as it evolves. For example, a service may be in a “test” or “frozen” state.

For the purposes of this example, we will use the “adabas_EmployeesMini” sample service that can be created when a new Adabas driver is defined (check “Create Sample Services” box). The process is the same for any other service defined in SOA Gateway.

Important:
The SOA Gateway legacy perspective should not be used for configuring services with lifecycle governance. All further references to the Control Centre assume the SOA Gateway Administration perspective.

Concept

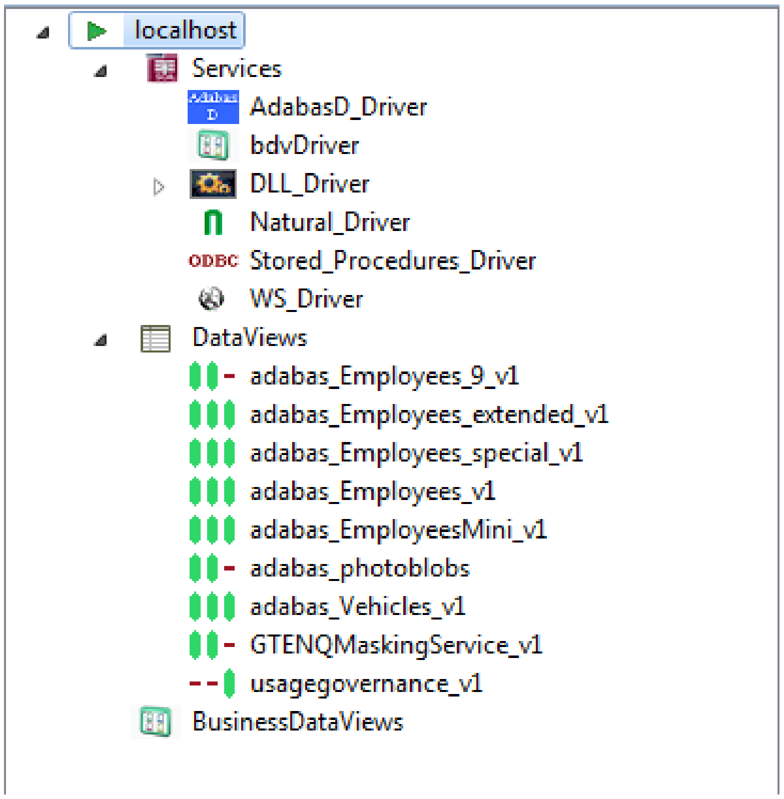
The basic concept of lifecycle governance is that it allows changes to be made to existing services without forcing existing clients to upgrade. Each client that wishes to consume a web service starts with retrieving the service WSDL. Thus, SOA Gateway will return different WSDLs based on the different version numbers of the service. The key point is that the endpoint will also have the version number as part of the URL. Thus a client may use “version 1” of a SOA Gateway web service. If the administrator decides that the web service needs to be changed, a new WSDL with a new endpoint is made available. New clients can use version 2, while existing clients can continue to use version 1 seamlessly.

From a clients perspective, this process is seamless, there is no requirement for them to have any understanding of web service versions. New clients just start using the WSDL, existing clients work with their services as before.

From an administrators perspective, SOA Gateway will control the allocation and incrementation of service version numbers.

Web Service Versioning

Every web service that is created in SOA Gateway has an associated version. By default, this version number starts a "1". The control centre will display the version number of each service after its service name.



When retrieving the WSDL for a service, SOA Gateway will always return the most recent version of the WSDL. The most recent version is determined by the incrementally highest version number. For example, version 3 is "more recent" than version 1. It is still possible to retrieve a WSDL other than the most recent, but you must pass an extra "__version=N" argument on the URL. For example:

```
http://host:port/adabas_EmployeesMini?WSDL&__version=1
```

Important:
You will notice that the WSDL endpoint now has a unique version number appended on the end of the URL in the <service> section.

Web Service Status

There are 4 possible states a SOA Gateway Web Service can be in.

1. Test

In this state, the service may be modified as often as is required to bring the service to a point where it is to be made available to others. The service can be edited and/or modified, and the version number will stay the same.

2. Frozen

When the service is to be deployed or made available to others for use, it must be frozen. At this point, the version assigned to the service will represent this version of the service forever. There can be multiple frozen versions of any service, but they must have different version numbers.

3. Deprecated

When a service has been marked as deprecated, clients will still be able to call the service, but they will receive a warning message in the response, and the administrator will see a warning message in the system log. In this scenario, it is expected that administrators should recommend a service upgrade to their clients.

4. Historical

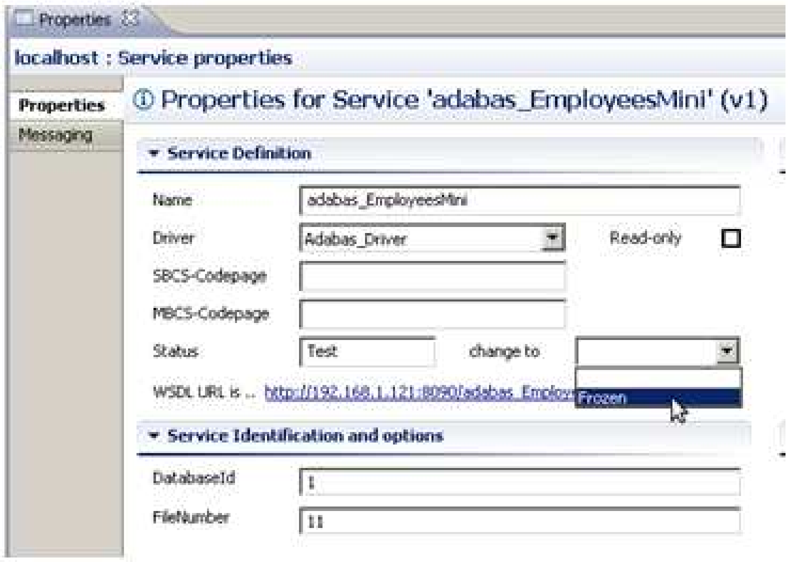
This status will be assigned to a service which is no longer active and has essentially been deleted. It is maintained in the configuration for historical purposes. If this service is invoked, the client will be returned a soap fault and a record logged that an attempt was made to use a historical version of the service.

Practical Example

In this example, we will use the "adabas_EmployeesMini" service that is created when adding a new Adabas driver. The principal is the same for all other SOA Gateway web services. Initially, the version number of the web service is "1". This service can be modified has often as required, and neither the version number nor the service state will change.

Freezing a service

When the administrator is happy that the service is ready to be put into general use, they can mark the service as "Frozen" using the drop down box in the web service properties.


Properties
localhost : Service properties
Properties
Messaging
Properties for Service 'adabas_EmployeesMini' (v1)
Service Definition
Read-only
change to
Frozen
Service Identification and options

Name	adabas_EmployeesMini
Driver	Adabas_Driver
SBCS-Codepage	
MBCS-Codepage	
Status	Test
WSDL URL is ...	http://192.168.1.121:8090/adabas_Employ
DatabaseId	1
FileNumber	11

Once the service is frozen, it is not possible to change the DataView of this service. It is still possible however to modify the properties of this web service. For example, it is still possible to change the database id/name of the frozen web service.

Modifying a service

If the administrator now wishes to make changes to the DataView of the frozen service, it is still possible to right-click the service, and choose “Edit DataView”. This will bring up the data view editor as usual. When this view is saved, the control centre will create a new version of the modified service, bye incrementing the version number.

Therefore, no modifications have been made to the frozen service (version 1), but a new service has been created (version 2) which contains these modifications. The services names will be the same (adabas_EmployeesMini). The version 2 service will be created with its status set to Test.

Accessing WSDL and calling the service

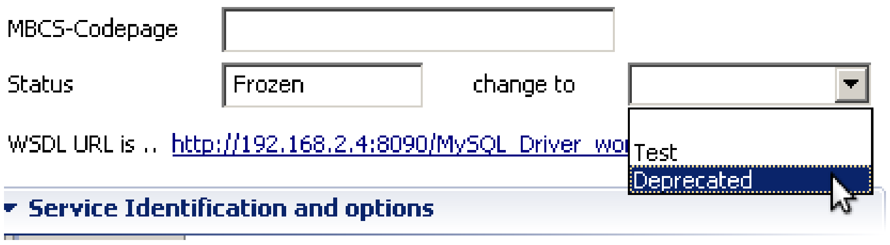
When a client requests the WSDL for this service, by default they will receive the version 2 WSDL. Thus new clients will get the most up to date version of the service. It is assumed that existing clients will only import the WSDL once (when version 1 was the most recent). To access the version 1 WSDL, provide the __version=1 argument.

```
http://host:port/adabas_EmployeesMini?WSDL&__version=1
```

Any client that uses the existing version 1 service will now receive a informational message in the SOAP response Header indicating that a newer version of this service exists. Clients could be configured to recognised this information, and take appropriate action, such as synchronizing with the SOA Gateway to get the latest version of this service.

Deprecating a service

When the administrator wishes to move clients off an old service, they should mark it as deprecated.


MBCS-Codepage
Status Frozen change to
WSDL URL is .. http://192.168.2.4:8090/MySQL_Driver_wo

Clients will not be able to retrieve the WSDL for this service version, so new clients cannot use this service. Existing clients will receive a warning message in the SOAP headers when they try to use this service. A warning message will also be written to the server log.

```
Fri May 13 21:12:46.00558327, pid: 00003267, tid: -1237358848. WARNING:
in file uriCache.cpp, at line 1358. indicating The service
MySQL_Driver_world_City version 1 has been deprecated.
```

Historical Service

When the lifecycle of a service has come to an end, it can be marked as Historical. This means the WSDL is no longer available, and any client trying to call this service version will receive an error message.

```
Fri May 13 21:12:46.00558327, pid: 00003267, tid: -1237358848. ERROR:
in file uriCache.cpp, at line 1358. uriCache::checkForCacheEntry()
returned -33044, indicating The service MySQL_Driver_world_City version
1 is no longer available
```