

Service Configuration

When a driver entry is expanded, the list of services associated with this driver is also displayed.

Left-clicking the web service name brings up the Service Properties in the Properties view.

Right-clicking the web service name brings up the Service Context Menu

This section also provides information about Lifecycle Governance (LG)

Service Properties

The screenshot shows the 'Properties for Service 'adabas_Employees_extended' (v1)' dialog box. It is organized into several panes:

- Service Definition:** Name (adabas_Employees_extended), Driver (Adabas_Driver), SBCS-Codepage, MBCS-Codepage, Status (Test), and WSDL URL (http://localhost:56005/adabas_Employees_extended?WSDL).
- Service Settings:** Write Byte Order Mark (BOM) checkbox.
- Service Identification and options:** DatabaseId (1), FileNumber (9).
- Service Meta Information:** FileName, MaxIsn, LobFileNumber, LobFileName.
- Usage Governance and Statistics:** Service Statistics checkbox, Use service overrides, Use global settings, Information, Input Data, Output Data checkboxes.

When a web service is selected, the following information is displayed.

- Name : The name of the web service. This will determine the WSDL URL and service endpoint.
- Driver: The driver in use for this web service.
- SBCS/MBCS : The SingleByte/MultiByte codepages used for data conversion with this service.
- Service Status: For more details see here
- Service Identification. These fields vary on the type of service. They define the resource-specific parameters for this service.
- Service Meta Information: These fields vary on the type of service. They define the resource-specific meta parameters for this service, often used when creating new resources. For example the "MaxIsn" value (setting the maximum amount of records which can be stored on the file) to be used when creating a new Adabas file.

- Usage Governance Information. For more details see [here](#)

Important:

The **Save** button must be clicked to publish the changes to the server. The button is located in the top-right corner of the properties view.

Service context menu



- *Edit DataView*: Edit the DataView associated with the selected Service. The dataview will open in the Editor View, and the details of each field will appear in the Properties View. Changes will not be reflected on the web service (WSDL) until the file is saved.
- *Rename Service*: Rename the service. This will have consequences on the URL of the WSDL, and the endpoint of the web service.
- *Duplicate Service*: Create a new web service that is a copy of the existing. This service needs to have a different name from the original.
- *Refresh Service*: clears the internal cache on the SOA Gateway server to pick up changes to the (Business)DataView, XSD or XSL
- *Copy Service definitions*:
- *UDDI Register Service via JAXR or SOAP*
- *Show WSDL* : Display the WSDL for this service. This is the starting point of for any web service clients that wish to consume this web service.
- *Remove Service*: Remove this service from the SOA Gateway Server

Important:

If the *Remove orphaned dataviews* box is checked, the internal configuration files (XRD, XSD and XSL) will also be deleted from the server.

- *Create Physical Resource*: Create a back-end resource (E.g Adabas file), physically, based on the SOA Gateway Definitions.
- *Delete Physical Resource*: Physically delete a back-end resource described by the web service.

Rename Service

- Select "Rename Service" from the context menu
- This will bring up a dialog and allow you to choose a new name for the service. By default the original service name will be appended with the string "_Rename".

Enter the new name of your choice and click *OK*.

- The service will appear in the list of services with its new name immediately

Duplicate Service

This function will create an initially exact copy of a service. You will be prompted to enter a name for the copy.

The copy will appear in the list of services immediately

Modify the Dataview and/or service parameters as needed.

Copy Service definitions to other server(s)

Copying service definitions from one server to another is possible in two ways

- using the Copy Wizard
- using drag-and-drop

Copy Service(s) using the Copy Wizard

- Right-click the service you wish to copy, select the *Copy Service definitions ...* function from the context menu.
- Assumptions:
 - Both SOA Gateway Servers are accessible from the Control Centre.
 - If migrating services from a previous version of SOA Gateway, the utility is invoked from the most recent Control Centre.
 - Both SOA Gateway servers have the same drivers enabled and configured. See here for more info.

Note:

"Same drivers" refers to the *type* of drivers, they need not necessarily be of the same name.

Process:

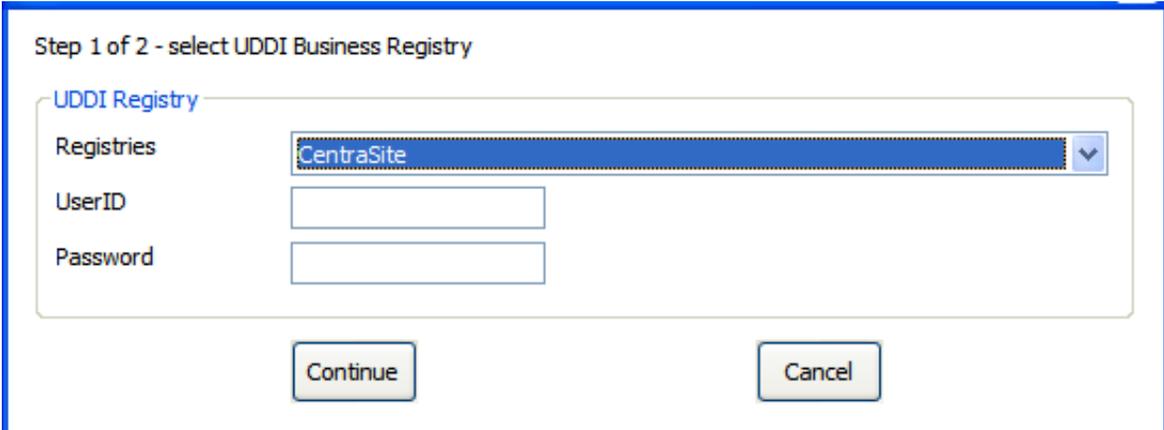
1. In the SOA Gateway you wish to *copy from*, choose the service you wish to copy by right-clicking and choosing **Copy Service Definitions**
2. Choose the server you wish to *copy the service to*.
3. If no driver with the same name exists on the target server you will be given the option to choose another driver of the same type as the source.
4. If you wish to overwrite the service on the target server, check the **Overwrite existing....** box.
5. The status of the copy will be written to the Action Log, and this you can double-click this to get more information.

Register a Service with an UDDI Business Registry (UBR)

SOA Gateway Services, which are essentially "Services", can be registered with an UDDI Business Registry, for example Software AG's Centrasite, so that any UDDI enabled client application can find an exposed service, retrieve its signature, issue requests against it etc.

These are the steps required to register a Service with an UBR:

1. Define a UDDI Business Registry to the SOA Gateway Control Centre (from the SOA Gateway Eclipse Preferences dialog)
2. Select the **UDDI register Service** function from the context menu
3. Select one of the predefined UDDI Business Registries, enter the user credentials required to be able to publish



Step 1 of 2 - select UDDI Business Registry

UDDI Registry

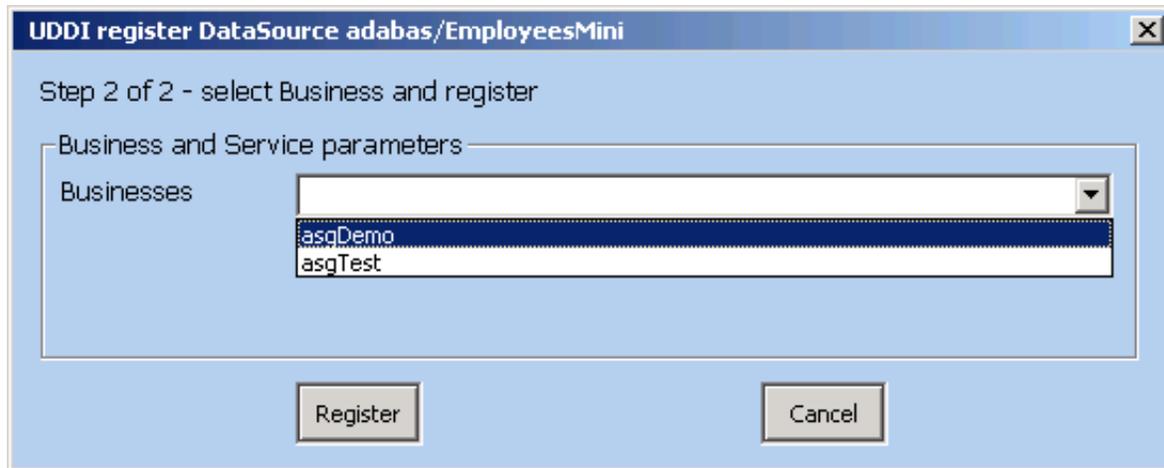
Registries: CentraSite

UserID:

Password:

Continue Cancel

4. Select a Business registered with the UDDI Registry for the selected account, click **Register** to continue



5. After successful completion of the registration process the service and tModel registry keys will be shown



6. Click **Done**

Lifecycle Governance

The term *Lifecycle Governance* (LG) is used to define the SOA Gateway handling of distinct versions and states of web services. States allow the administrator to define services in Test mode, move them to frozen production environments, and, in time, deprecate them from use. Versioning allows the administrator to define multiple versions of a service.

Service Version

The WSDL is the main starting point for web service clients who wish to consume SOA Gateway web services. As part of the SOA Gateway LG support, it is possible to create multiple versions of the WSDL (and so the web service), thus allowing different clients to consume the different versions. It also allows modifications to be made to existing services, safe in the knowledge that existing clients will not be affected.

- When requesting the WSDL, SOA Gateway will always return the most recent version. To retrieve a specific version, the `__version` argument must be provided on the WSDL URL. E.g.

```
http://host:port/myService?WSDL&__version=5
```

The WSDL will contain a service endpoint which reflects the service version.

- When working with REST requests, the same `__version` argument can be used.

```
http://host:port/myService?LIST&__version=5&ID=4*
```

- Internally, a new DataView is created for every service version. The DataView name will have the version number appended on the end of the file name.

Service Status

There are 4 statuses a SOA Gateway service can be in:

1. *Test*: 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
2. *Frozen*: The DataView (metadata) for this service cannot be modified. If the service is being migrated from a Test status, it will be assigned a version number incremented by 1.

If a newer version of this service exists, clients will receive an informational message in the SOAP Header response.

When the Administrator edits the DataView of a frozen service, a new version of this service will be created, it's status set to test, and it's version number incremented by 1.

3. *Deprecated*: Clients will still be able to call a service with this status, but they will receive a warning message in the SOAP headers. A warning message will also be written to the SOA Gateway log, and the client should be urged to upgrade.
4. *Historical*: The service is no longer active, and clients who attempt to call this service will receive a SOAP fault.

When a service is in a specific status, the following transitions are possible

<i>Current Status</i>	<i>Next Status</i>
Test	Frozen
Frozen	Test, Deprecated
Deprecated	Test, Historical
Historical	Deprecated