

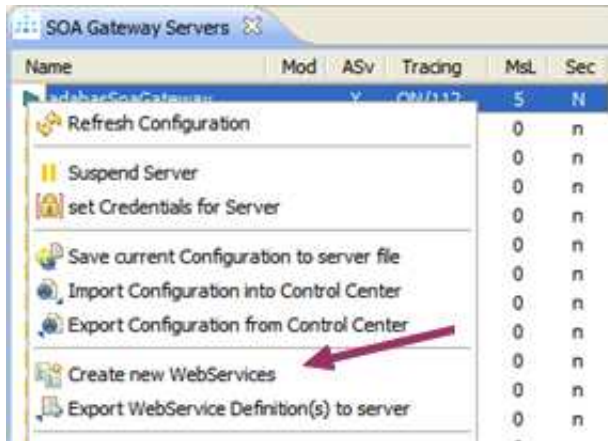
# SOA Gateway WebService Creation

WebService definitions, and related mapping files (DataViews, XSDs) are created, in a semi-automatic to automatic process (also known as WebService Discovery), from meta-data describing the characteristics of resources to be exposed as a WebService thru the SOA Gateway, for example from an Adabas FDT, SQL Table description, or Natural sources.

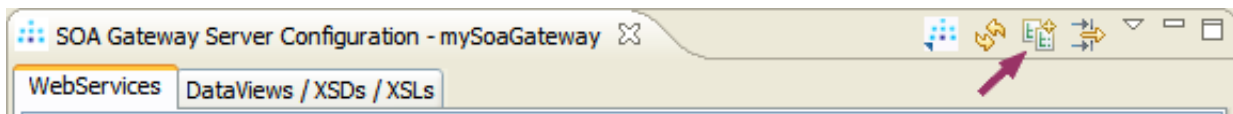
- Initiating WebService creation
- Creating WebService(s) from Adabas
- Creating WebServices from SQL databases
- Creating WebServices from Stored Procedures
- Creating WebServices from a SYSOBJH extract file
- Creating WebService(s) from Natural

## Initiating WebService creation

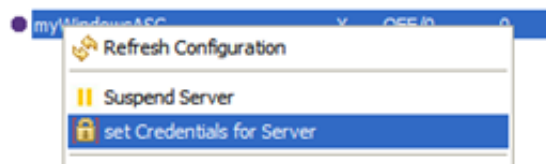
The WebService creation process can either be started from the server's context menu:



or by clicking the 'Create new WebServices' action button in the title area of the Configuration View:



It may be required to provide login credentials, for example when accessing a SQL database system, this can be done from the server's context menu as well:

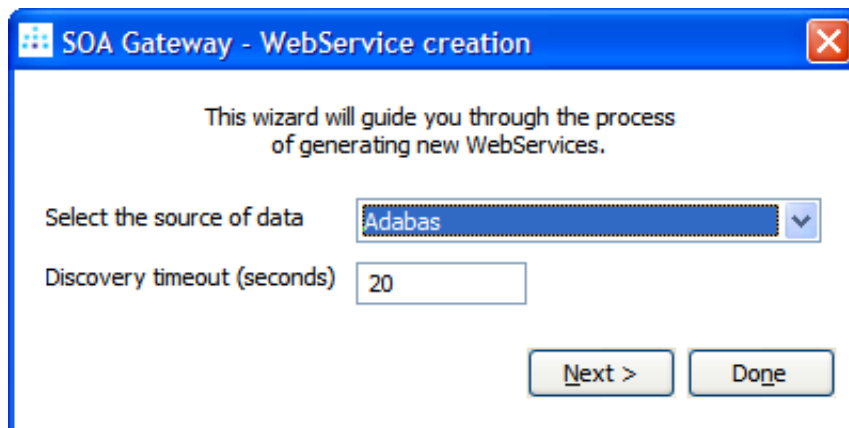


Specify the credentials - User ID and Password - in the *SOAP Security* section, click *Set*



The screenshot shows a dialog box titled "Credentials for SOA Gateway Server". It contains two sections: "HTTP Authentication" and "SOAP Security". In the "HTTP Authentication" section, the "User ID" and "Password" fields are empty. In the "SOAP Security" section, the "User ID" field contains the text "myUserId" and the "Password" field is filled with a series of dots. At the bottom of the dialog, there are two buttons: "Set" and "Cancel".

Once the WebService Creation Wizard has been activated, the first step is to select the driver to be used as the source:

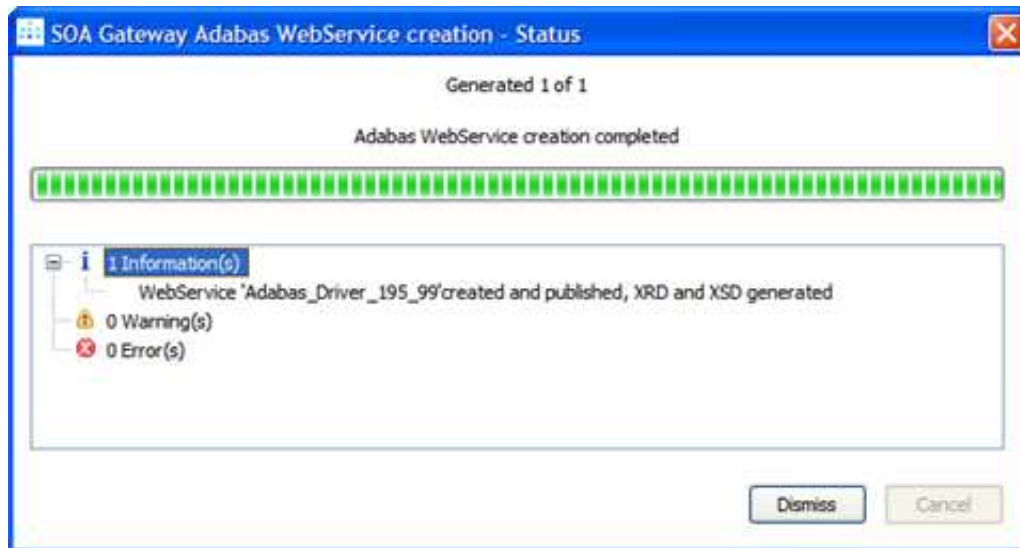


The screenshot shows a dialog box titled "SOA Gateway - WebService creation". It contains the text "This wizard will guide you through the process of generating new WebServices." Below this text, there is a dropdown menu labeled "Select the source of data" with "Adabas" selected. To the right of the dropdown is a small downward arrow. Below the dropdown is a text box labeled "Discovery timeout (seconds)" with the value "20" entered. At the bottom right of the dialog, there are two buttons: "Next >" and "Done".

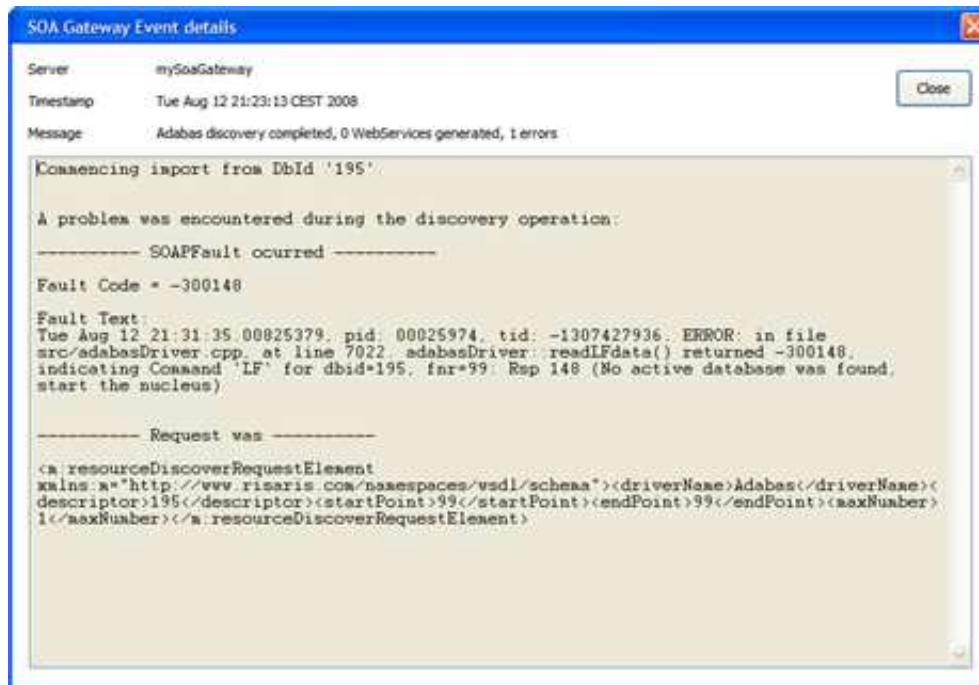
The dropdown-box will contain an entry for every defined driver, plus a number of special entries, for example *<SYSOBJH>*, which can be used to create WebService(s) from a SYSOBJH unload file containing DDMs, or *<3GL>* to create a DataView and XSD from a C- or COBOL program or copybook.

Additionally, if the process of gathering information about resources accessible by the selected driver takes an extended period of time, a higher timeout value can be specified here.

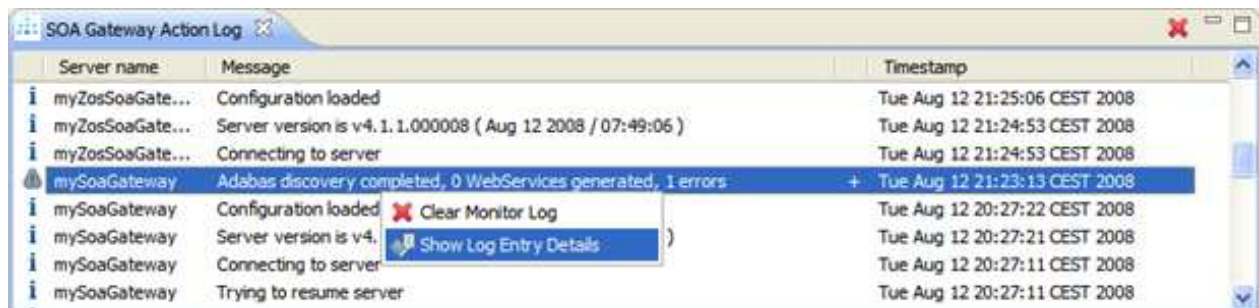
While the import process is running, its status is continually shown in the following window:



Error messages are shown in the status window



Status and error information can be re-retrieved later on from the SOA Gateway Action Log



## Creating WebService(s) from Adabas

- Select a defined driver of type 'adabas' from the dropdown box
- Click the 'Next >' button
- Specify the Database Id (range 1 - 255 on OpenSystems, 1 - 65535 on Mainframe platforms), plus 'From' and 'To' file numbers (1 - 5000) to limit the discovery to the selected range. Click the 'Discover' button:

The screenshot shows a dialog box titled "SOA Gateway Adabas WebService creation" with a close button (X) in the top right corner. The dialog is divided into two main sections. The first section, "Specify input & option(s)", contains four input fields: "Database Id" (a single character field), "File range from" (a numeric field), "to" (a text label), and another numeric field for the range end. Below these is a "Password" field and a checkbox labeled "Generate ISN as DataView field . . . . .". The second section, "Options for generated WebService name", contains a checkbox labeled "Advanced Options". At the bottom of the dialog are two buttons: "Discover" and "Cancel".

**Note:**

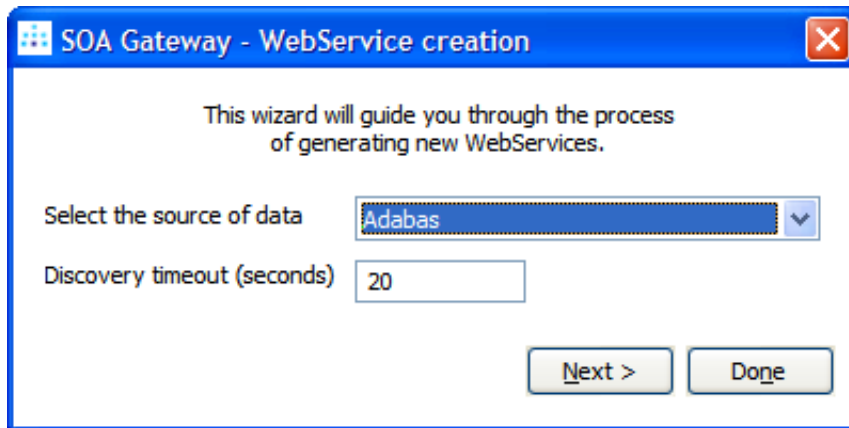
Discovering the full range of possible file numbers (1 - 5000) may take substantial time and adversely affect performance of both the SOA Gateway server as well as the Adabas target database. Break the discovery process down to smaller range(s) of files.

- Select the file(s) to be turned into WebServices, click the 'Import' button:



Mod	Driver	WebService	DataSource Id	DataView
	Adabas Driver	Adabas_Driver_195_99	DatabaseId=195, FileNumber=99	Adabas_Driver_195_99

- The Wizard is still active and ready for further discovery, click 'Next >' to start the process again, or 'Done' to dismiss the wizard.



## Creating WebServices from SQL databases

- Select a defined driver pointing to the SQL database system of your choice from the dropdown box, this step-by-step guide will use MySQL to demonstrate the process.
- It is assumed that the 'world database' demo has been loaded as per the guide published at the MySQL Community website, and a ODBC Data Source been set up as 'world\_dsn', pointing to the 'world' database.
- Click the 'Next >' button to start the process
- Specify the ODBC Dsn to be browsed, optionally enter a 'From Table' limiting the operation, for example 'C%' will return a list of tables starting with an uppercase 'C'. Optionally specify 'Max.Tables' to further limit the number of tables to be listed. Click the 'Discover' button:

SOA Gateway WebService Creation (mysql)

Step 1 - select input

Specify ODBC descriptors & option(s)

ODBC Dsn: world\_dsn

UserId / Password: [ ] [ ]

Pattern: C%

Max.entries to list: 99

Options for generated WebService name

Advanced Options:

Discover Cancel

- zOS and DB2

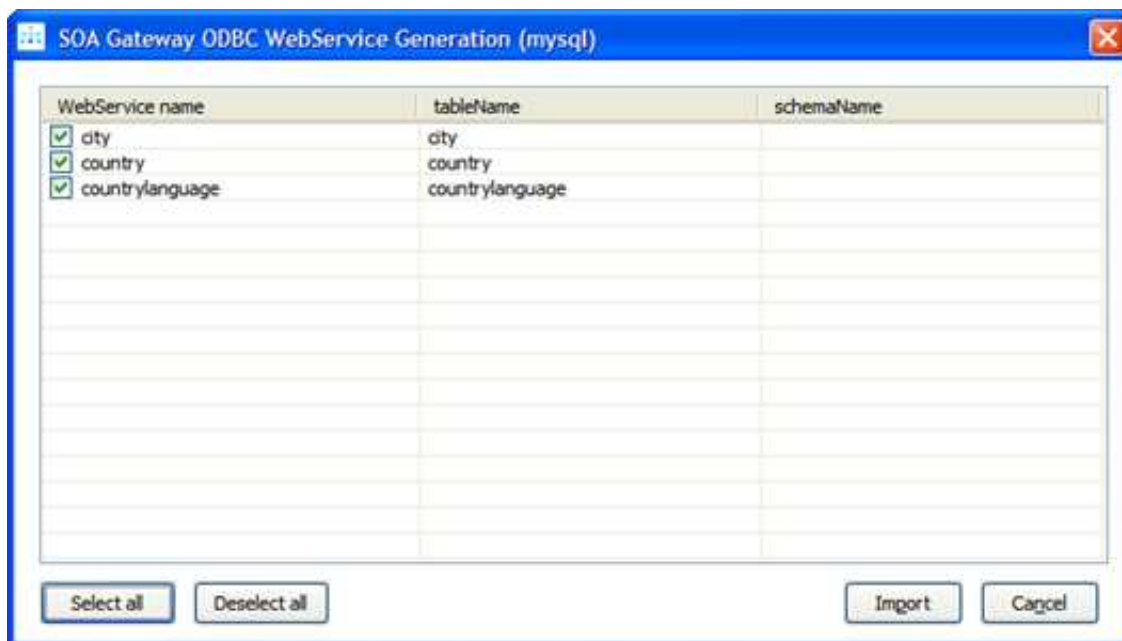
In the ODBC Dsn field above - enter the Location name returned from the DB2 command Display DDF

- in the example below the location name is S1D831

```
COMMAND INPUT ==> /-D831 DIS DDF
RESPONSE=POC1
DSNL080I -D831 DSNLTDDF DISPLAY DDF REPORT FOLLOWS:
DSNL081I STATUS=STARTD
DSNL082I LOCATION          LUNAME          GENERICCLU
DSNL083I S1D831           DEIBMIPA.IPAA77D8 -NONE
.....
```

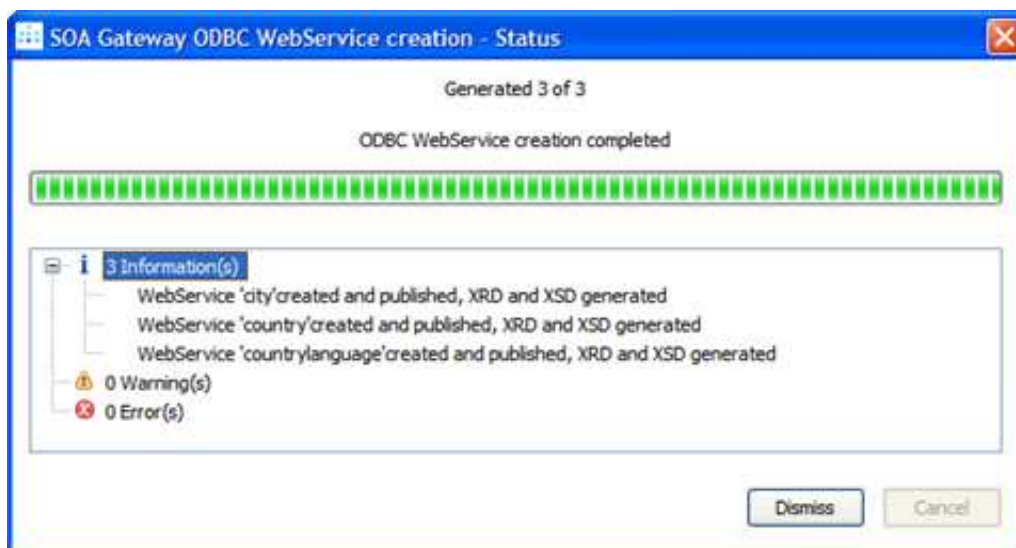
Also ensure that the member CLIINI in the JCL dataset has been modified as necessary. Refer to your DBA on this.

- Select the table(s) to be turned into WebServices, click the 'Import' button:



Optionally, the WebService name(s) can be modified.

- The generation report indicates what has been generated:



- The requested WebServices and WebService definition files (DataView(s) and XSD(s)) have been added to your server configuration

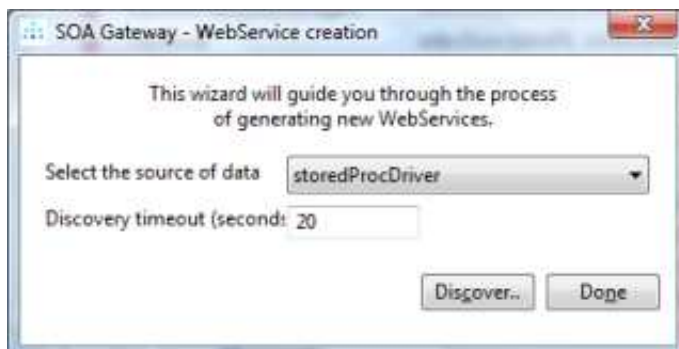
Mod	Driver	WebService	DataSource Id	DataView
MySQL	MySQL	city	odbcDsn=world_dsn, tableName=city	city
MySQL	MySQL	country	odbcDsn=world_dsn, tableName=country	country
MySQL	MySQL	countrylanguage	odbcDsn=world_dsn, tableName=countrylanguage	countrylanguage

- The Wizard is still active and ready for further discovery, click 'Discover' to start the process again, or 'Done' to dismiss the wizard.



## Creating WebServices from Stored Procedures

- Details on how to create/modify/delete a stored procedure driver can be found [here](#).
- Select the stored procedure driver from the dropdown box. This step-by-step guide will use MS SQL Server database AdventureWorks to demonstrate the discovery process.
- Click the 'Discover' button next to the driver selection box



- Specify the ODBC Dsn to be browsed. Optionally enter a 'Pattern' limiting the operation, for example 'uspGet%' will return a list of stored procedures starting with 'uspGet' as in the example below. Optionally specify 'Max. entries to list' to further limit the number of stored procedures to be listed. Click the 'Discover' button:

SOA Gateway WebService Creation (storedProcs)

Step 1 - select input

Specify ODBC descriptors & option(s)

ODBC Dsn: RisarisDS

Userid / Password: [ ] [ ]

Pattern: uspGet%

Max entries to list: [ ]

Options for generated WebService name

replace whitespaces: [ ]

prefix with:  storedProcDriver\_

Discover Cancel

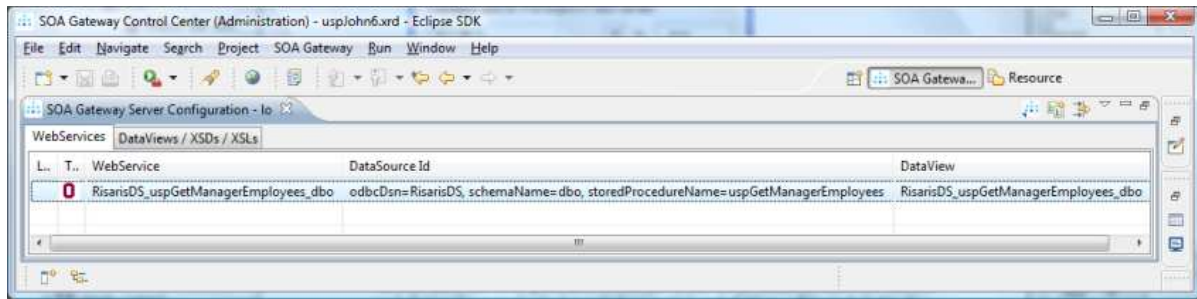
- Select the stored procedure(s) to be turned into WebServices, click the 'Import' button:

SOA Gateway ODBC WebService Generation (storedProcs)

WebService name	storedProcedureName	schemaName
<input type="checkbox"/> RisarisDS_uspGetBillOfMaterials_dbo	uspGetBillOfMaterials	dbo
<input type="checkbox"/> RisarisDS_uspGetEmployeeManagers_dbo	uspGetEmployeeManagers	dbo
<input checked="" type="checkbox"/> RisarisDS_uspGetManagerEmployees_dbo	uspGetManagerEmployees	dbo
<input type="checkbox"/> RisarisDS_uspGetWhereUsedProductID_dbo	uspGetWhereUsedProductID	dbo

Select all Deselect all Import Cancel

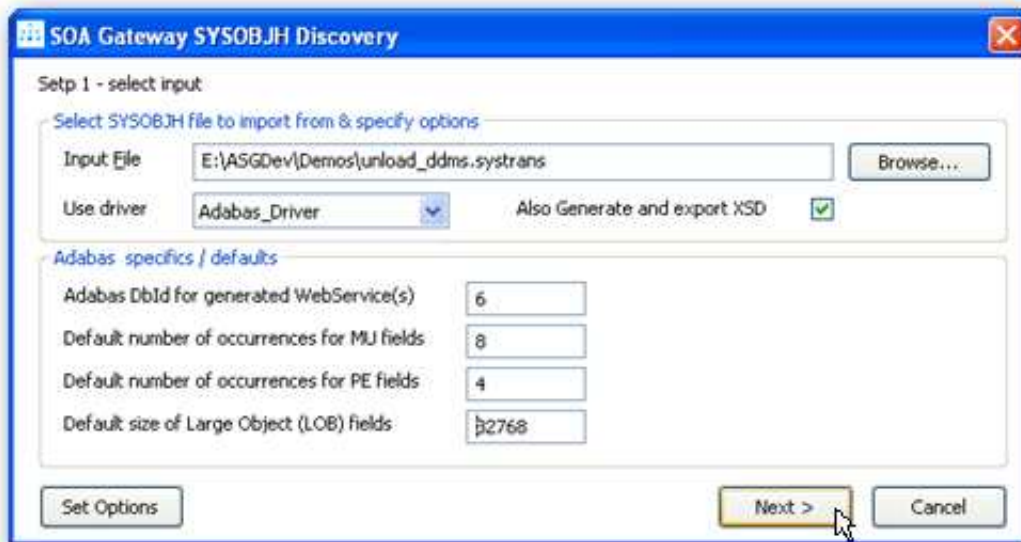
- The generation report indicates what has been generated.
- The requested WebServices and WebService definition files (DataView(s) and XSD(s)) have been added to your server configuration



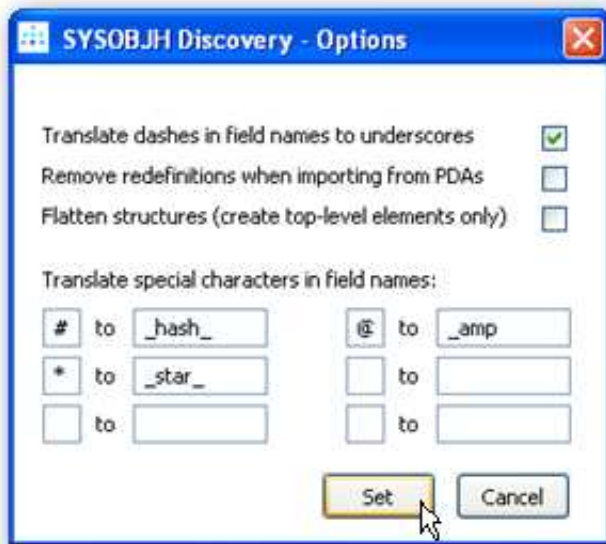
- The Wizard is still active and ready for further discovery, click 'Discover' to start the process again, or 'Done' to dismiss the wizard.

## Creating WebServices from a SYSOBJH extract file

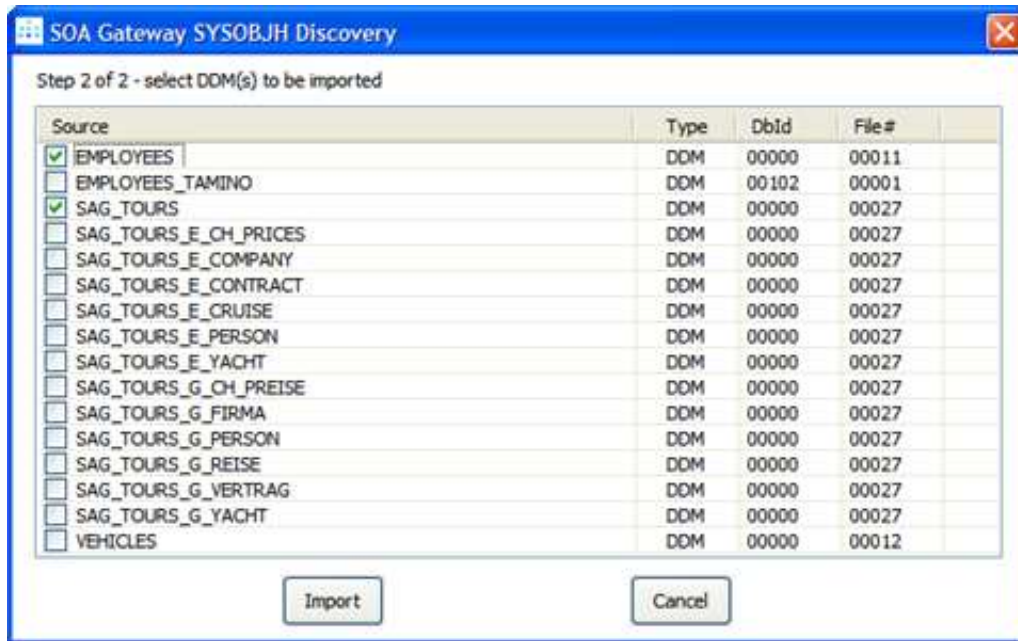
- Select the 'special' driver <SYSOBJH> from the dropdown box.
- A step-by-step guide on how to create a SYSOBJH extract file can be found here.
- Click the 'Next >' button to start the process
- Select the SYSOBJH file to be used as the input to the creation process, specify if XSD(s) are to be created for the generated WebService(s) as well, in addition to the DataView(s). Optionally specify defaults to be used when generating WebService(s) and DataView(s) based on Adabas files - the Database Id used when the DDM is defined with a DbId of 0 (zero), and the default number of occurrences for MU(ultiple value) fields and PE(riodic) groups:



- Overriding options can be specified from the Options dialog, click the "Set Options" to activate it



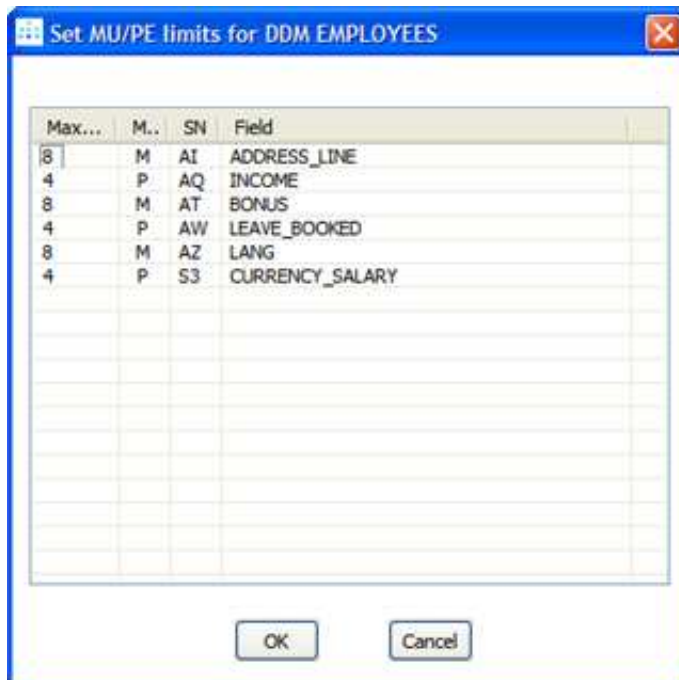
- if you wish dashes ('-') in file and element names to be translated to underscores ('\_'). This may be desirable where the target programming language used to access SOA Gateway Resources do not accept dashes in field names, an example being PHP.
  - When PDAs include redefinitions you will need to choose the layout that suits you for the DataView to be generated. If you are unsure at the moment, leave the "remove redefinitions" box unchecked, in which case ALL redefinitions will be included in the generated DataView, with a generated fgroup name of "\*redefinition\_of\_<PDA\_group\_name>". Before the DataView is uploaded to the server, remove all definitions/redefinitions not required. Multiple DataViews (and thus resources) are required when requiring different layouts/set of parameters for the same subprogram.
  - "Flatten structures" when the PDA includes nested groups but you want all fields to be generated at the "top level" without any group/field structure.
  - You may need to translate special characters, like the # (hash) sign if they are used as part of field name(s), because they are not valid in a SOAP context.
  - Click the "Set" button to save your changes. They will be saved permanently, you will not have to set your preferences every time.
- Click the *Continue* button
  - Select the DDM(s) to be turned into WebServices, click the 'Import' button:



- For each DDM selected, which contains MU and or PE fields, a dialog will request you to specify the maximum number of occurrences expected.

**Important:**

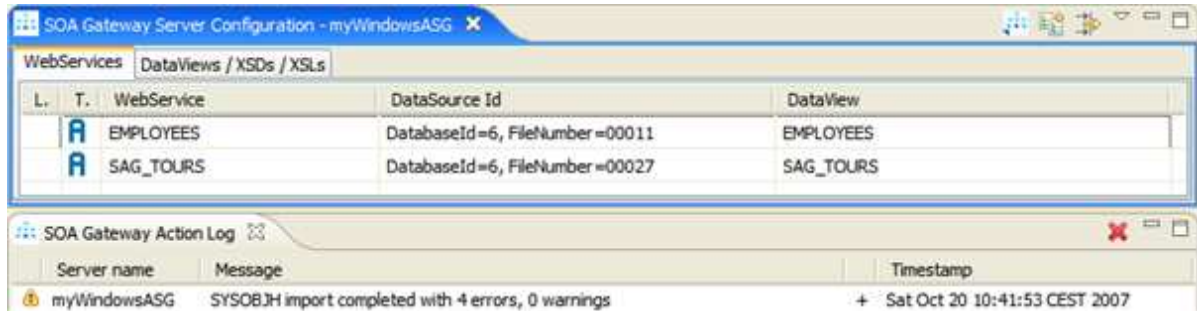
The "MaxOcc" accepted by this dialog is 32767, depending on the ADABAS version the actual limit accepted by the database system may be lower (191).



- The generation report indicates what has been generated:



- The requested WebServices and WebService definition files (DataView(s) and XSD(s)) have been added to your server configuration



- The Wizard is still active and ready for further discovery, click 'Next >' to start the process again, or 'Done' to dismiss the wizard.



## Creating WebService(s) from Natural

- Select a defined driver of type 'natural' from the dropdown box
- Click the 'Next >' button
- Specify the name of the Library to be scanned, select if it is a Natural "User" or a "System" library. Click the 'Discover' button:

SOA Gateway Natural WebService creation

No libraries matching criteria, please re-specify

Specify input

System Library from SYSSOAE\X to  Select ...

Next > Cancel

Natural Import Options

Translate dashes in field names to underscores

Translate special characters in field names:

#	to	<input type="text"/>	@	to	<input type="text" value="_amp"/>
*	to	<input type="text" value="_star_"/>		to	<input type="text"/>
	to	<input type="text"/>		to	<input type="text"/>

Additional Options

Defaults for WebService name generation

Alternatively, if the library name is unknown, specify a generic library name, or a from-to range and click the **Select** button. For example, the following input

SOA Gateway Natural WebService creation

No libraries matching criteria, please re-specify

Specify input

System Library from SYSSOA\* to  Select...

Next > Cancel

Natural Import Options

Translate dashes in field names to underscores

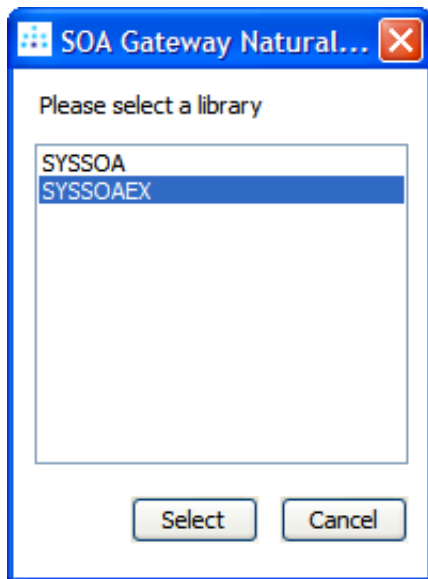
Translate special characters in field names:

#	to	<input type="text"/>	@	to	<input type="text" value="_amp"/>
*	to	<input type="text" value="_star_"/>		to	<input type="text"/>
	to	<input type="text"/>		to	<input type="text"/>

Additional Options

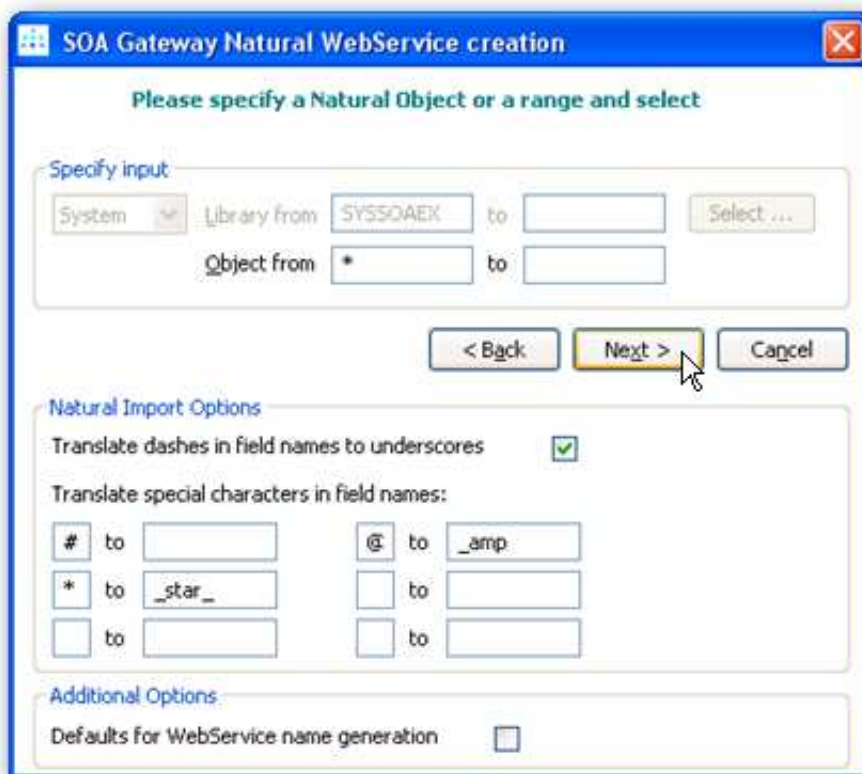
Defaults for WebService name generation

will result in this additional selection dialog to pop up



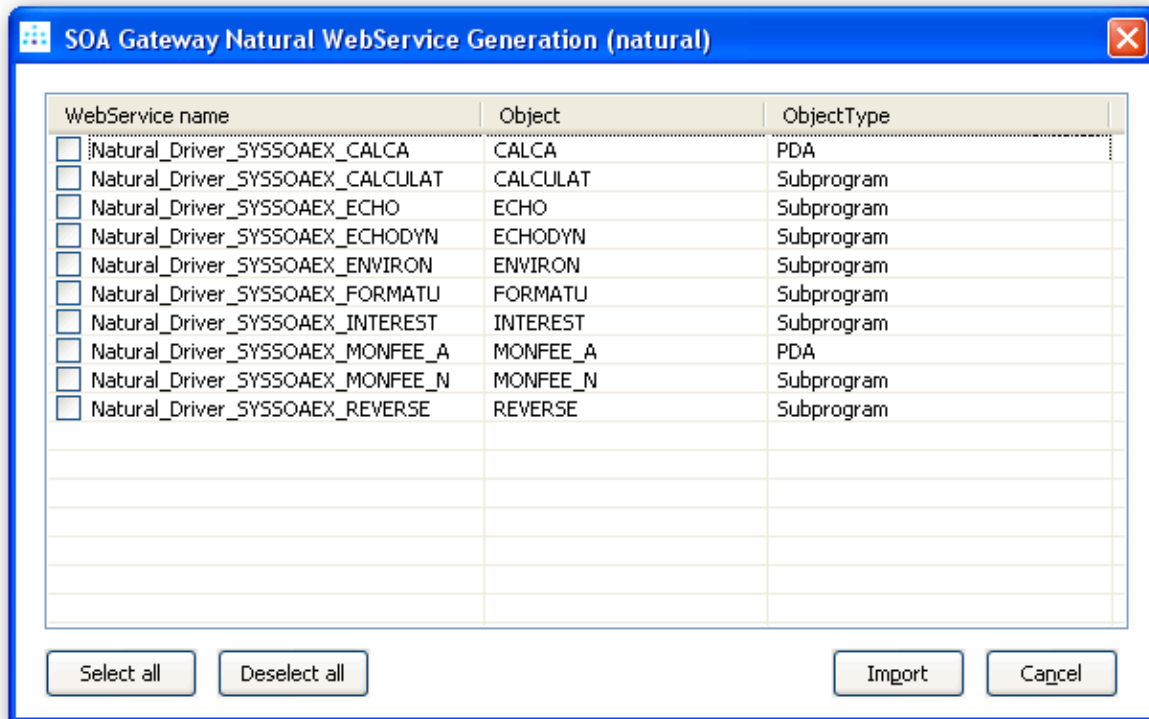
Select a library with a double-click, or select a library name and click **Select**, or stop the wizard by clicking the **Cancel** button.

- Select the name(s) of the object(s) to be turned into WebServices, click the **Next** button. Again, this can be a specific object name, a generic object name, or a range of objects. When a specific object name is entered in the Object field, but the "to" Object name field is left empty, this will result in all objects *starting from the specified name* will be listed



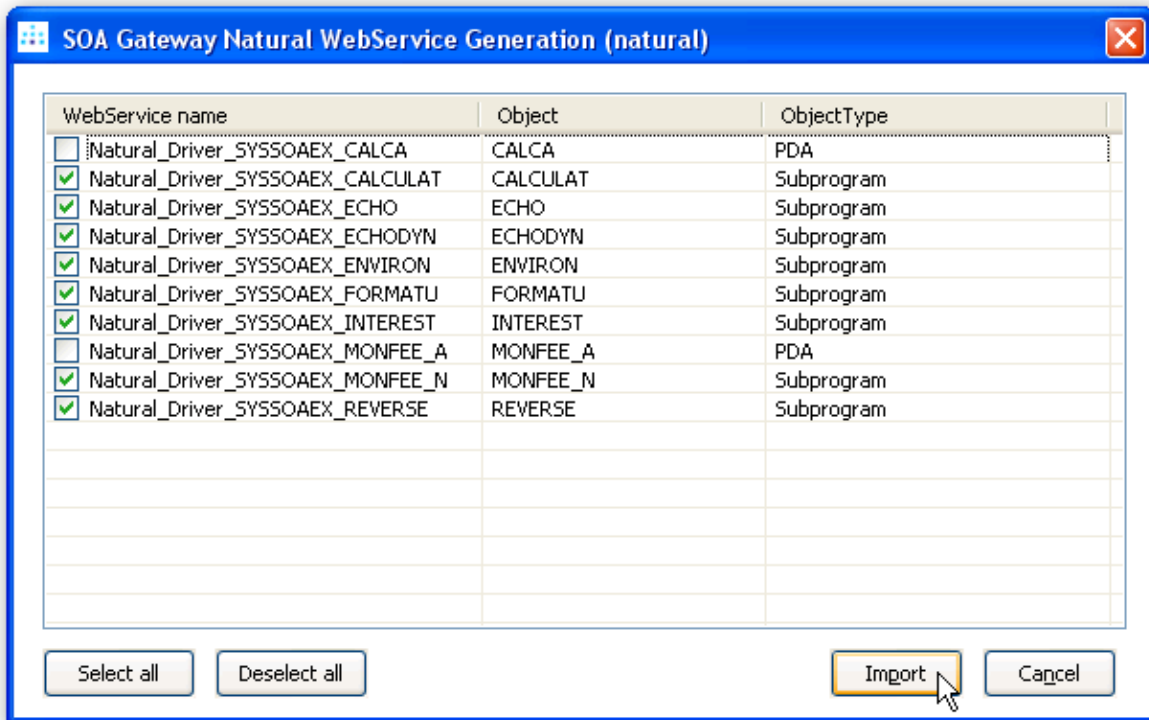
You may return to Library selection mode by clicking the **Back** button.

- A list of objects matching the selection criteria is displayed

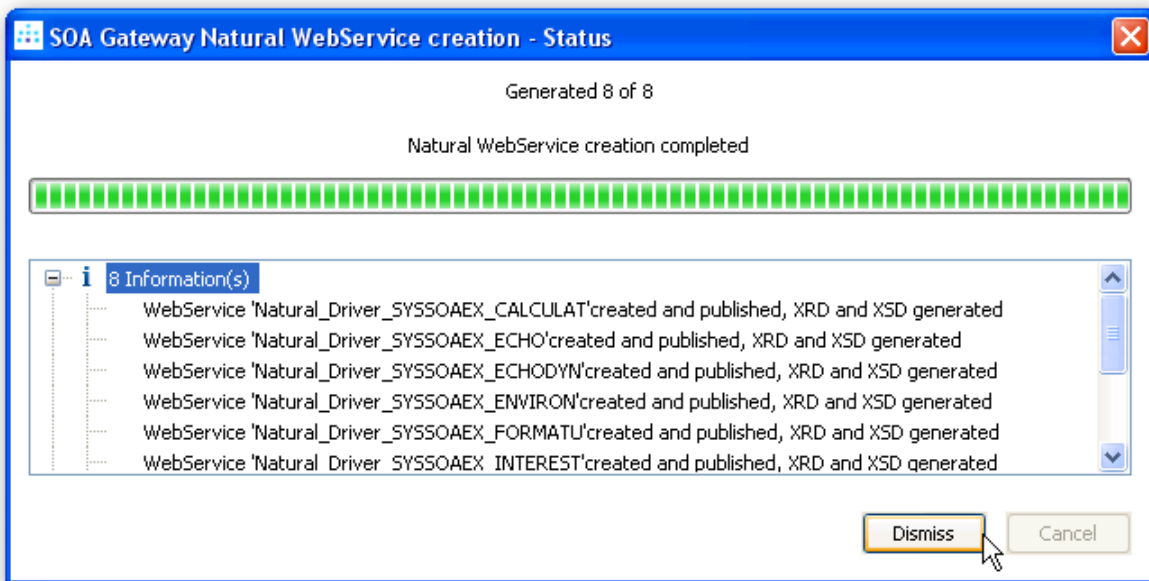


Either select all, or only the one(s) you are specifically interested in, click the **Import** button.

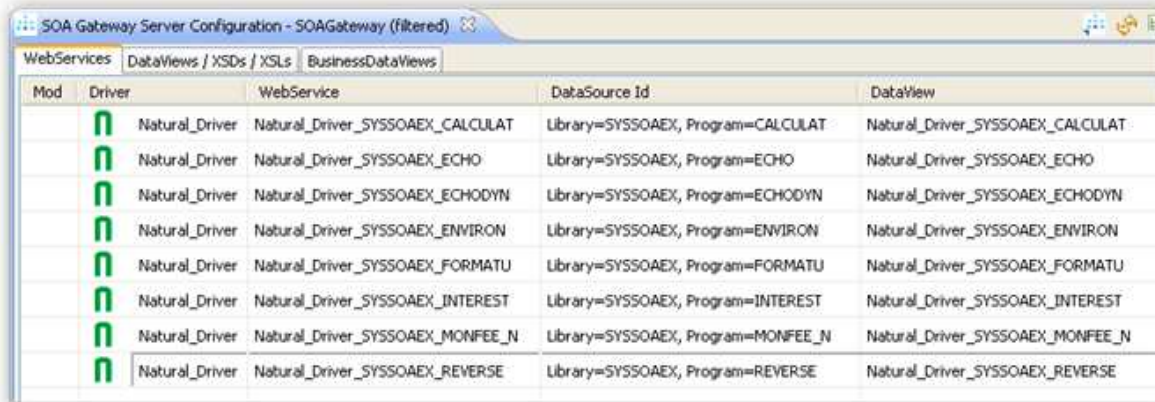
- In this case we select (check the box left to the WebService name) all objects of type subprogram.



The generation report indicates what has been created:



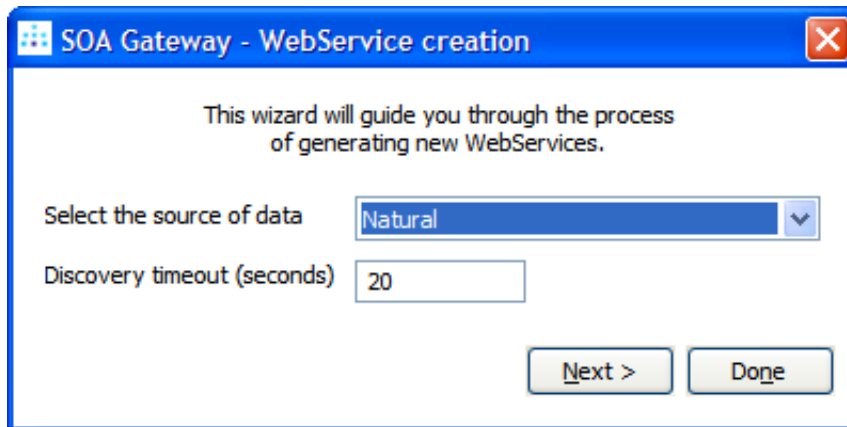
- The requested WebServices and WebService definition files (DataView(s) and XSD(s)) have been added to your server configuration



The screenshot shows a window titled "SOA Gateway Server Configuration - SOAGateway (filtered)". The window has tabs for "WebServices", "DataViews / XSDs / XSLs", and "BusinessDataViews". The "WebServices" tab is active, displaying a table with the following columns: Mod, Driver, Webservice, DataSource Id, and DataView. The table contains 8 rows, all with a green 'N' icon in the Mod column.

Mod	Driver	Webservice	DataSource Id	DataView
N	Natural_Driver	Natural_Driver_SYSSOAEX_CALCULAT	Library=SYSSOAEX, Program=CALCULAT	Natural_Driver_SYSSOAEX_CALCULAT
N	Natural_Driver	Natural_Driver_SYSSOAEX_ECHO	Library=SYSSOAEX, Program=ECHO	Natural_Driver_SYSSOAEX_ECHO
N	Natural_Driver	Natural_Driver_SYSSOAEX_ECHODYN	Library=SYSSOAEX, Program=ECHODYN	Natural_Driver_SYSSOAEX_ECHODYN
N	Natural_Driver	Natural_Driver_SYSSOAEX_ENVIRON	Library=SYSSOAEX, Program=ENVIRON	Natural_Driver_SYSSOAEX_ENVIRON
N	Natural_Driver	Natural_Driver_SYSSOAEX_FORMATU	Library=SYSSOAEX, Program=FORMATU	Natural_Driver_SYSSOAEX_FORMATU
N	Natural_Driver	Natural_Driver_SYSSOAEX_INTEREST	Library=SYSSOAEX, Program=INTEREST	Natural_Driver_SYSSOAEX_INTEREST
N	Natural_Driver	Natural_Driver_SYSSOAEX_MONFEE_N	Library=SYSSOAEX, Program=MONFEE_N	Natural_Driver_SYSSOAEX_MONFEE_N
N	Natural_Driver	Natural_Driver_SYSSOAEX_REVERSE	Library=SYSSOAEX, Program=REVERSE	Natural_Driver_SYSSOAEX_REVERSE

- The Wizard is still active and ready for further discovery, click 'Next >' to start the process again, or 'Done' to dismiss the wizard.



The screenshot shows a dialog box titled "SOA Gateway - WebService creation". The dialog contains the following text and controls:

This wizard will guide you through the process of generating new WebServices.

Select the source of data:

Discovery timeout (seconds):

Buttons: