



	Version 2010-05-31 18:26:45 (MET)	Reference
--	--------------------------------------	-----------



This document applies to RisAris SOA Gateway 2010-05-31 18:26:45 (MET) and to all subsequent releases.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

© Copyright RisAris 2010.

All rights reserved.

The name RisAris Limited and/or all RisAris Limited product names are either trademarks or registered trademarks of RisAris Limited. Other company and product names mentioned herein may be trademarks of their respective owners.

Table of Contents

1 Reference	1
2 SOA Gateway SOAP Reference for Resource Access	3
Overview	4
3 SOA Gateway SOAP Reference for Server Configuration	17
Soap Operations for Server Configuration	18
4 Frequently Asked Questions	21
How do I active the Software AG sagenv file post SOA Gateway Installation?	22
How do I modify the machine identifier in the JESMSGGLG?	22
5 Performance Hints	23
6 Internationalization	25
Setting codepages	26
Which codepage do I use?	26
SOAP versus REST differences	26
Troubleshooting	27

1 Reference

The following reference materials are available for the SOA Gateway:

- Glossary
- **SOAP Reference for Resource Access**
- **SOAP Reference for Server Configuration**

2 SOA Gateway SOAP Reference for Resource Access

- Overview 4

Overview

This document describes the SOAP Operations and their inputs, exposed by the SOA Gateway to access data sources.

Prerequisites

Retrieve WSDL

Web Service Operations

Providing Key Information

SOAP Header Usage

Prerequisites

At this point your SOA Gateway Server should be installed, configured and started.

If you have not yet configured any web services, please refer to the SOA Gateway Control Center section and add at least one web service

Retrieving the WSDL for your resource.

All web services defined will have a WSDL associated with them. This is the starting point for using the operations provided by the SOA Gateway. The WSDL describes the operations that may be carried out, and how they are used. This includes a description of valid parameters, data and responses for each of the operations.

In order to get the WSDL for a particular resource you simply issue a standard http request, specifying:

- the server name or *IP-address* of the server where the SOA Gateway is installed [and running]
- the *TCP Port number* that the SOA Gateway is listening on (as provided in the installation)
- the *name* of the service. (this is the value of the "*Name*" field in the web service properties)

The following example shows the URL required to retrieve the WSDL for an Adabas "Employees" file.

Server Name	SOAGateway
Port Number	56000
Name	adabas_Employees
URL for WSDL	http://SOAGateway:56000/adabas_Employees?WSDL

The SOA Gateway Web Services Operations for Data Resources

The operations provided by the SOA Gateway for accessing data resources [files, databases, programs, etc.] are now explained.

Parameters are required unless otherwise stated.

Operation	<i>list</i>
Description	<p>The <i>list</i> operation returns a list of records or rows from your data source.</p> <p>The data returned can be limited or restricted by providing <i>key</i> information.</p> <p>This operation will only be available when the web service is a "database" type, for example, Adabas or MySQL</p>
Parameters	<ul style="list-style-type: none"> ■ <i>Key data</i> <p>Key data must be entered for at least one of the fields defined as a key.</p> <p>Refer to the section Specifying Key Data for more information.</p> ■ <i>Options</i> <p>None</p>
Result	<p>The result will be either :</p> <ul style="list-style-type: none"> ■ an XML document wrapped in a SOAP message and containing the requested data ■ a SOAP fault message

Operation	<i>select</i>
Description	<p>The <i>select</i> operation returns a list of records or rows from your data source. The maximum number of rows/records returned can be set via the SOA Gateway Control Center.</p> <p>This operation will only be available when the web service is a "database" type, for example, Adabas or MySQL</p> <p>The data returned can be limited or restricted by providing <i>key</i> information. The select operation extends the capability of the list operation by enabling searches on a larger set of criteria.</p> <p>The key information for a select is wrapped in a condition block and can be repeated several times within that block. Each key entry represents an 'AND' condition. Condition blocks can</p>

	also be repeated several times. Each condition block represents an 'OR' condition. The condition block accepts the following specifiers:
Conditions	<ul style="list-style-type: none"> ■ <i>Less than a specific value (LT).</i> ■ <i>Less than or equal to a specific value (LE).</i> ■ <i>Equal to a specific value (EQ).</i> ■ <i>Greater than a specific value (GT).</i> ■ <i>Greater than or equal to a specific value (GE).</i> ■ <i>Not equal to a specific value (NE).</i> ■ <i>Starting with a specific value (START). Character based fields only.</i> ■ <i>Ending with a specific value (ENDS). Character based fields only.</i> ■ <i>Containing a specific value (CONTAINS). Character based fields only.</i>
Example	<pre><soapenv:Body> <cus:customerinformationGroupSelectElement> <condition> <CustomerNumber Condition="GE">40</CustomerNumber> <CustomerNumber Condition="LT">50</CustomerNumber> </condition> <condition> <CustomerNumber Condition="EQ">59</CustomerNumber> </condition> <condition> <CustomerNumber Condition="EQ">61</CustomerNumber> </condition> </cus:customerinformationGroupSelectElement> </soapenv:Body></pre> <p>The example above specifies 3 condition blocks. This will return data where the (CustomerNumber >= 40 and CustomerNumber <= 50) or CustomerNumber = 59 or CustomerNumber = 61</p>
Parameters	<ul style="list-style-type: none"> ■ <i>Key data</i> <p>Key data must be entered for at least one of the fields defined as a key.</p> <p>Refer to the section Specifying Key Data for more information.</p> <ul style="list-style-type: none"> ■ <i>Options</i> <p>None</p>
Result	<p>The result will be either :</p> <ul style="list-style-type: none"> ■ an XML document wrapped in a SOAP message and containing the requested data ■ a SOAP fault message

Operation	selectNext
Description	<p>The <i>selectNext</i> operation returns a list of records or rows from your data source. A selectNext operation may only be called following a select and subsequently other selectNext calls. For this functionality the initial select operation has to initiate a new Conversation. See Conversational Processing. The resultant conversation id must be passed in any associated selectNext calls. selectNext calls may be issued until end of file is reached or may be terminated by a selectEnd call. The maximum number of rows/records will be that set for set via the Soa Gateway Control Center.</p> <p>This operation will only be available when the web service is a "database" type, for example, Adabas or MySQL</p> <p>There is no key information for a selectNext operation as this will have been passed in by the initiating select operation.</p>
Parameters	<ul style="list-style-type: none"> ■ <i>Key data</i> <li style="padding-left: 20px;">None ■ <i>Options</i> <li style="padding-left: 20px;">None
Result	<p>The result will be either :</p> <ul style="list-style-type: none"> ■ an XML document wrapped in a SOAP message and containing the requested data ■ a SOAP fault message

Operation	selectEnd
Description	<p>The <i>selectEnd</i> operation terminates a sequence of select and/or selectNext calls with a conversation. A selectEnd operation may only be called following select or selectNext operations. For this functionality the select operation has to initiate a new Conversation. The resultant conversation id must be passed in the selectEnd call.</p> <p>This operation will only be available when the web service is a "database" type, for example, Adabas or MySQL</p> <p>There is no key information for a selectEnd operation as this will have been passed by the initiating select operation.</p>
Parameters	<ul style="list-style-type: none"> ■ <i>Key data</i> <li style="padding-left: 20px;">None ■ <i>Options</i> <li style="padding-left: 20px;">None
Result	<p>The result will be either :</p>

	<ul style="list-style-type: none"> ■ an XML document wrapped in a SOAP message containing an acknowledgement that the operation was successful ■ a SOAP fault message
Operation	selectCount
Description	<p>The <i>selectCount</i> operation returns a count of the records or rows that match the criteria set in the condition block(s).</p> <p>This operation will only be available when the web service is a "database" type, for example, Adabas or MySQL</p> <p>The selectCount operation is identical to that of the select operation in terms of its search capabilities.</p> <p>The key information for a selectCount is wrapped in a condition block and can be repeated several times within that block. Each key entry represents an 'AND' condition. Condition blocks can also be repeated several times. Each condition block represents an 'OR' condition. The condition block accepts the following specifiers:</p>
Conditions	<ul style="list-style-type: none"> ■ <i>Less than a specific value (LT).</i> ■ <i>Less than or equal to a specific value (LE).</i> ■ <i>Equal to a specific value (EQ).</i> ■ <i>Greater than a specific value (GT).</i> ■ <i>Greater than or equal to a specific value (GE).</i> ■ <i>Not equal to a specific value (NE).</i> ■ <i>Starting with a specific value (START). Character based fields only.</i> ■ <i>Ending with a specific value (ENDS). Character based fields only.</i> ■ <i>Containing a specific value (CONTAINS). Character based fields only.</i>
Parameters	<ul style="list-style-type: none"> ■ <i>Key data</i> <p>Key data must be entered for at least one of the fields defined as a key.</p> <p>Refer to the section Specifying Key Data for more information.</p> <ul style="list-style-type: none"> ■ <i>Options</i> <p>None</p>
Result	<p>The result will be either :</p> <ul style="list-style-type: none"> ■ an XML document wrapped in a SOAP message and containing the number of records/rows satisfying the search criteria ■ a SOAP fault message

Operation	<i>get</i>
Description	<p>The <i>get</i> operation returns a single record or row from your data source.</p> <p>The data returned is specified by providing <i>unique key</i> information identifying a single record / row.</p> <p>This operation will only be available when the web service is a "database" type, for example, Adabas or MySQL</p>
Parameters	<ul style="list-style-type: none"> ■ <i>Key Data</i> <p>Key data must be entered for at least one of the fields defined as a primary key field, so that a single record can be identified.</p> <p>Refer to the section Specifying Key Data for more information.</p> <ul style="list-style-type: none"> ■ <i>Options</i> <p>None</p>
Result	<p>The result will be either :</p> <ul style="list-style-type: none"> ■ an XML document wrapped in a SOAP message and containing the requested data ■ a SOAP fault message

Operation	<i>add</i>
Description	<p>The <i>add</i> operation adds a single record or row of data to your data source.</p> <p>This operation will only be available when the web service is a "database" type, for example, Adabas or MySQL</p>
Parameters	<ul style="list-style-type: none"> ■ <i>Add Data</i> <p>Provide values for each of the fields defined. These are the fields / columns of the data on your data source.</p> <p>You must add data for at least one of the fields specified as being <i>primary key</i> fields.</p> <p>Refer to the section Specifying Key Data for more information.</p> <ul style="list-style-type: none"> ■ <i>Options</i> <p>None</p>
Result	<p>The result will be either :</p> <ul style="list-style-type: none"> ■ a SOAP message indicating the operation succeeded ■ a SOAP fault message

Operation	<i>update</i>
Description	<p>The <i>update</i> operation updates a single record or row of data.</p> <p>This operation will only be available when the web service is a "database" type, for example, Adabas or MySQL</p>
Parameters	<ul style="list-style-type: none"> ■ <i>Update Data</i> <p>Provide values for each of the fields defined. These are the fields / columns of the data on your data source.</p> <p>Key data must be entered for at least one of the fields defined as a <i>primary key</i> field.</p> <p>Refer to the section Specifying Key Data for more information.</p> <p>Others fields may be left empty if you do not wish to change their value.</p> <ul style="list-style-type: none"> ■ <i>Options</i> <p>None</p>
Result	<p>The result will be either :</p> <ul style="list-style-type: none"> ■ a SOAP message indicating the operation succeeded ■ a SOAP fault message

Operation	<i>delete</i>
Description	<p>The <i>delete</i> operation deletes a single record or row of data from the data source.</p> <p>This operation will only be available when the web service is a "database" type, for example, Adabas or MySQL</p>
Parameters	<ul style="list-style-type: none"> ■ <i>Key Data</i> <p>Key data must be entered for at least one of the fields defined as a <i>primary key</i> field, so that a single record can be identified.</p> <p>Refer to the section Specifying Key Data for more information.</p> <ul style="list-style-type: none"> ■ <i>Options</i> <p>None</p>
Result	<p>The result will be either :</p> <ul style="list-style-type: none"> ■ a SOAP message indicating the operation succeeded ■ a SOAP fault message

Operation	<i>invoke</i>
Description	The <i>invoke</i> operation makes a call to a function or program. This operation will only be available when the web service is a "program" type, for example, NATURAL, COBOL or when the web service describes a database stored procedure.
Parameters	<ul style="list-style-type: none"> ■ <i>Parameters</i> <p>Each of the fields defined can be input, output, or input+output.</p> <p>Provide values for each of the fields defined as input or input+output. These match the function/program parameters that are required on the call.</p>
Result	<p>The result will be either :</p> <ul style="list-style-type: none"> ■ a SOAP message return the results of the program/function invoked ■ a SOAP fault message

Specifying Key Data

Keys or *Key Fields* in SOA Gateway terms are simply fields or columns on your data source (file, table, etc) that may be used to narrow a search for a record or row of data.



Note: *key fields* may or may not be *indexed* at your data source. Indexing provides better response times from your data source. If you are unsure what, if any, fields are key fields (in Adabas terms: Descriptors) or allowed to be used for searching, contact your data source administrator (DBA, etc.).

Key fields are specified to the SOA Gateway by setting the appropriate value for the attribute "*key type*" in the Resource Description.

Key Types

There are two types of *key fields* for data sources:

Primary Key Fields

Primary Key Fields are fields in a data source that must always contain unique values.

They are required on all data source operations except the *list* operation, where they are optional. Secondary keys may be used instead or in addition on a list request.

Their values will not be altered in the *update* operation.

Primary Keys cannot contain *wild card* symbols, except on the *list* operation.

Secondary Key Fields

Secondary Key Fields are fields that may be used in narrowing a search.

They are not required in any operation, and their values can be updated.

They may contain wild card symbols.

Using Wild card Symbols and other Generic Search criteria

Wild cards are used where you do not wish to specify an exact value in a key field, but use a generic specification that will match for a range of different values.

These search modifiers may only be used on the *list* operation.

The SOA Gateway currently supports the following generic search criteria :

Wild card for one or more characters

The character "*" may be used as a wild card for all characters.

It may only be used on fields defined as "string" fields.

Where it appears, it will match any character or group of characters.



Note: for Adabas resources, this wildcard may only appear at the *end* of the string data supplied.

Example

"Ga*" would match "Gat", "Gate", "Gateway", etc.

SOAP Header Usage

In the SOA Gateway, the SOAP Headers are used for versioning, the support of conversational SOAP processing, support of transactions, and specific settings on the datasource you are accessing. By default all elements are "empty". To get the default behaviour, all header elements should be left blank, or removed altogether. Example:

```
<soap:Envelope xmlns:rapdv="http://www.risaris.com/namespaces/xmiddle"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <soap:Header>
    <rapdv:AdabasEmployeeHeader>
      <Version/>
      <ConversationState/>
      <ConversationId/>
      <TransactionState/>
      <TransactionId/>
    </rapdv:AdabasEmployeeHeader>
  </soap:Header>
  <soap:Body>
    ...
  </soap:Body>
</soap:Envelope>
```

```
</soap:Body>
</soap:Envelope>
```

Conversational Processing

Every time a SOAP request is made to the SOA Gateway, this request must be associated with a specific context. By default, a new context is created and destroyed for every SOAP Request.

The user may also use the SOAP Headers to re-use and re-connect to a specific context.

When a user starts, re-uses, and ultimately finishes with their context, the process is known as a "conversation". In the SOAP Header, the "ConversationState" and "ConversationId" are used for conversational processing, see below for more information.

The Version Element

The 'Version' element is currently unused. It will be brought into use in future versions of the SOA Gateway.

The ConversationState Element

The 'ConversationState' element is used to control conversation processing. It should be one of the following:

State	Description
New	A new conversation is being started. In this case, the ConversationID (see below) value must be NULL or an error will occur.
Old	An existing conversation is active with which the current SOAP message should be associated. When the SOAP request is processed, the conversation must remain active as there are further SOAP requests to be sent. The ConversationID found must have been returned as a result of a previous 'new' request. An error will occur if the ConversationId (see below) provided cannot be found.
End	An existing conversation is active with which the current SOAP message should be associated. When the SOAP request is processed, the conversation must be terminated. The ConversationID (see below) found must have been returned as a result of a previous 'new' request. An error will occur if the ConversationId provided cannot be found.

The ConversationId Element

The ConversationId uniquely identifies the conversation, and it returned only after a 'New' request is successfully processed. The user should never modify or create this ID. The ConversationId must be present on an 'Old' or 'End' request.

Transaction Processing

In the SOA Gateway, the platform's Transaction Manager (TM) will be engaged to handle transactions. A default TM is provided as part of the ASG installation. It can be found in <asg install dir>/libraries/transactionManagerDummyDll.so. The environment variable TMSTUB is used to point to the transaction manager shared object. This shared object provides interfaces to handle the transaction.



Note: The framework for engaging transaction managers is subject to change. Currently the transactionManagerDummyDll.so does not provide any "real" transactionality. Rísaris are current looking for early adopters to work with us to fully implement and test this technology.

In the SOAP Header, the "TransactionState" and "TransactionId" are used for transactional processing; see the table below for more information.

The TransactionState Element

If a transaction involves modifications that will occur over multiple SOAP messages, the conversational processing logic must be used to associate the SOAP messages and thus the transactional information.

If a transaction involves modifications that will all be completed as part of the one SOAP message, the conversational processing logic is not required.

The Transaction State may be one of the following:

State	Description
New	<p>A new transaction is being started.</p> <p>If no conversation will exist after this request, an error will result.</p> <p>If the request is part of an existing conversation, and this conversation already has a transaction associated with it, this will also be rejected as an error.</p>
Old	<p>An existing transaction is active with which the current SOAP message should be associated.</p> <p>If no conversation exists for the SOAP request, or no conversation will exist after the SOAP request, an error will result.</p> <p>If no transaction is active and associated with the current conversation, an error will result.</p>
Commit	This will cause a commit to be issued when the current SOAP message has been processed.

State	Description
	<p>If no conversation existed previously for this SOAP request, a new transaction will be started.</p> <p>If a conversation does exist for this SOAP request, any updates made as a result of the previous SOAP requests will be committed.</p> <p>When returned in the SOAP response, it indicates that the transaction has been committed.</p>
Backout	<p>This will cause a backout to be issued when the current SOAP message has been processed.</p> <p>If a conversation does exist for this SOAP request, any updates made as a result of the previous SOAP requests will be backed out.</p> <p>If no conversation existed previously for this SOAP request, an error will result.</p> <p>When returned in the SOAP response, it indicates that the transaction has been backed out.</p>

The Transaction Id Element

The Transaction ID is purely informational, and has no functional bearing on the transaction process. It is returned on any SOAP message and is intended to be used for tracking purposes.

Adabas specific headers

When working with Adabas services, there are a number of specific SOAP headers that will be available

SOAGateway_Internal_Adabas_ISN	On a GET operation an ISN (Adabas Internal Sequence Number) can be used in lieu of a primary key to retrieve a record.
SOAGateway_Internal_Adabas_StartAtISN	In addition to the specification of key data this header option can be used to further limit the result set by setting a starting ISN for LIST and SELECT operations.
SOAGateway_Internal_Adabas_Read_Direction	Specify an "A" (the default) for ascending retrieval, a "D" for descending.
SOAGateway_Internal_Adabas_Sort_Order	The result set can be sorted by up to three fields. Specify their XML names, comma separated, with this header option
SOAGateway_Internal_Adabas_Password	The password required to access an Adabas file secured with ADASCR (Adabas Security). This password is different from that in the global "soap security" section.
SOAGateway_Internal_Adabas_Multifetch_Limit	Limit the number of records to be retrieved by a "multifetched" operation (LIST or SELECT / SELECTNEXT) with a single call to Adabas.

Relational database specific headers

When working with relational database services, i.e. MySQL, MS SQL Server, DB2, Oracle, etc, the specific SOAP Header are

SOAGateway_Internal_AutoCommit	Turn off the AutoCommit flag on the database
--------------------------------	--

3 SOA Gateway SOAP Reference for Server Configuration

- Soap Operations for Server Configuration 18

Soap Operations for Server Configuration

The SOA Gateway exposes a number of SOAP operations / methods to retrieve and alter the SOA Gateway Server Configuration from any SOAP enabled client.



Note: these interfaces are likely to change in the future, they are provided on a trial basis, be aware that you might have to change any "applications" built on top of these interfaces ! As soon as "stable" interfaces are available, this fact will be announced and documented.

adaptorList

This operation is used to load up a SOA Gateway driver library, and return the associated internal information.

This operation takes 1 input, `libraryName` should be set to the name of the library to load and query.

configList

This operation is used to list the current configuration in use by the SOA Gateway server. The current configuration is useful where you wish to make changes to existing configuration items, or just to make sure you are not trying to add, for example, a resource *URI* that is already in use.

This operation currently has no options.

The full configuration document, minus the XML header, is returned in the *soap response*.

configRemove

This operation is used to reset the value of an item in the configuration, or to remove an item from the configuration.

Only certain items or levels of items may be altered using this operation. The definition in the WSDL shows what elements may be used. Refer to it for further details on what type of items may be removed and what items may only have their value reset.

A configuration item which is reset will assume it's default value if it is an item that cannot be removed from the configuration.

Configuration items changed by this operation are effective as soon as the engine can make them so.

This operation currently has no options.

configReplace

This operation is used to replace the current configuration file, or to write a new configuration to file on the server.

If the element '*configFileName*' is omitted or left empty, then the configuration file currently in use will be overwritten - if it is in the configuration directory.

If the element '*configFileName*' is specified, then the configuration will be written to a file of that name in the configuration directory.

The newly written configuration will not come into effect until the SOA Gateway server is next started.

If the element '*configFileName*' was specified, and you wish to use this configuration, then before re-starting the SOA Gateway server you must alter the system environment variable *XMIDDLE_CONFIGURATION_FILE* so that it refers to your new configuration file.

This operation currently has no options.

configSet

This operation is used to set the value of an item in the configuration, or to add an item to the configuration.

Only certain items or levels of items may be altered using this operation. The definition in the WSDL shows what elements may be used. Refer to it for further details on what type of items may be added and what items may only have their value altered.

Configuration items changed by this operation are effective as soon as the engine can make them so.

This operation currently has no options.

4

Frequently Asked Questions

- How do I activate the Software AG sagenv file post SOA Gateway Installation? 22
- How do I modify the machine identifier in the JESMSGLG? 22

How do I active the Software AG sagenv file post SOA Gateway Installation?

This issue only applies to the SOA Gateway running on Linux, AIX or Solaris. If your license does not have any Adabas or Natural drivers enabled, this issue does not apply.

If the sagenv file is not available during the SOA Gateway install, Adabas or Natural drivers may not work correctly until this file is made available. For example, you have installed the SOA Gateway before installing Adabas.

Follow these steps to enable a new sagenv file in an existing SOA Gateway installation.

1. Edit the `[INSTALL_HOME]/xmiddleEnv.sh`
2. Append the following to the end of this file (assuming sagenv file is `/opt/softwareAg/sagenv.new`):

```
if test -e "/opt/softwareAg/sagenv.new"; then
source /opt/softwareAg/sagenv.new &> /dev/null
fi
```

3. Save and close this file.
4. Stop and then Start the SOA Gateway server. See here for more information.
5. Modify the `[INSTALL_HOME]/apache2/conf/adabas_soa_gw.conf` and change the User directive to use the sag user.
6. Change the file permissions of the SOA Gateway files and directories

```
chown -R sag [INSTALL_HOME]
```

7. Review the `error_log` for errors/warnings.
8. Adabas or Natural drivers should now be added successfully. See here for more info.

How do I modify the machine identifier in the JESMSG LG?

During the FTP of the SOA Gateway to z/OS, the machine identifier will be set to the hostnam or IP address of the FTP server. This text will be displayed on messages appearing in the JESMSG LG. To change this, modify the SYSPARM member of the CONF dataset and set this as required. It is recommended that this is set to the hostname or IP address of the z/OS machine.

5 Performance Hints

This section outlines some suggestions to improve the performance of the SOA Gateway.

■ Remove XSD

This only applies for web services which use the invoke operation, e.g services built from the Natural, Cobol or DLL drivers.

The SOA Gateway will validate the incoming XML against an XSD. This ensures that the contents and structure of the payload are correct, and will catch potential errors early on in processing. But XML validation is a relatively expensive operation, so it is possible to turn this off if required.

You may want to back up your existing XSD files before deletion. Use the "Import Web Service Definition" and ensure the XSD box is checked. See [here](#) for more information.

To delete the XSD, use the "Delete Web Service Definition", and ensure the XSD box is checked. The XSD has now been deleted from the server and validation of the payload will not take place.

■ Turning off Access Logging

Each time the SOA Gateway handles a request, it writes some logging information to the access log via Apache. By default, this file is access_log / access.log / DD:ACCESS based on the platform, *nix, Windows, z/OS respectively.

To restrict this logging, see the following Apache directive [here](#)

To remove this logging, remove the CustomLog directive from your httpd.conf / HTCONF. This can be accomplished by adding a # in front of the directive.

■ Use PFS caching

This only applies on z/OS or z/VSE.

Edit your SYSPARM and ensure caching of the SOA Gateway filesystem has been turned on. The option is CACHESIZE=N option on the CDI_DRIVER directive

E.g

```
CDI_DRIVER=('pfs,PAANPFS,CONTAINER=CIO://DD:PFS,CHARSET=ASCII,LRECL=4096,CACHESIZE=4096')
```

■ Change MPM settings

The SOA Gateway uses the [Apache worker MPM](#) to handle requests. This can be modified to increase server threads, therefore allowing the server to serve more requests. See the Apache documentation for more information.



Important: Ensure that the ServerLimit of 1 is maintained at all times. The SOA Gateway will not function correctly if more than server process is started.

6 Internationalization

- Setting codepages 26
- Which codepage do I use? 26
- SOAP versus REST differences 26
- Troubleshooting 27

The SOA Gateway uses IBM's [International Components for Unicode](#) to support internationalization (i18n). This supports text data conversion between almost any codepage.

Setting codepages

The Single Byte Character Set (SBCS) or Multi Byte Character Set (MBCS) codepage can be set on the driver, for more information, see [here](#). The codepages can also be set on the on the web service itself, by simply left-clicking the web service and entering the codepage in the appropriate section of the web service Properties.



Important: The codepage set on the web service overrides the one set on the driver.

Which codepage do I use?

This depends on what sort of information your service is going to return. Generally the ASCII codepage is sufficient for the English language. The ISO-8859-1 (often called latin1) codepage should suffice for most languages of Western Europe. The windows-1251 codepage supports Cyrillic languages such as Russian and Bulgarian. The ISO-8859-8 codepage can be used for Hebrew script.

The ICU home page has provided a [useful web page](#) which displays the ICU internal name, and a list of the aliases that the SOA Gateway will recognise. This page will also display the codepage map, which will allow you to choose the codepage best suited to your service.

SOAP versus REST differences

Generally when using WSDL and SOAP, once the correct codepage has been set, the payload should be recognised or returned correctly.

When using REST requests, things are slightly different. Non-ASCII characters entered on a URL bar of a browser will be escaped into their native hex value, of the form %XX. This native hex value differs depending on what codepage the browser recognises the character as. For example, a browser running in the latin1 codepage will recognise Á as %C1, but a browser running in the Cyrillic codepage will recognise Б as %C1.

For this reason the SOA Gateway allows users to provide an extra field on the REST request. This field is called `__encoding`. Thus users can indicate what codepage their browser is running in.



Important: By default, the SOA Gateway assumes the escaped values are in the ISO-8859-1. The `__encoding` field is not required in this case.

Example 1

The browser escapes the Russian Б into %C1. You need to tell the SOA Gateway that this is the Cyrillic encoding.

The URL should be `http://host:port/WebService?LIST&key=%C1&__encoding=windows-1251`.

Example 2

The browser escapes the Hebrew Shin (ש) into %F9. You need to tell the SOA Gateway that this is the Hebrew encoding.

The URL should be `http://host:port/WebService?LIST&key=%F9&__encoding=iso-8859-8`.

Troubleshooting

When the SOA Gateway cannot display a character in the requested codepage, it writes a message to the error log, and continues to attempt to process the rest of the payload. If you find your responses are missing some characters, check the `error_log / error.log / XMIDCARD` on *nix, Windows and z/OS respectively.

The error message to check should be something like this :

Unicode char 0xF1 is not representable in encoding ASCII.

