

# Common Gateway

# **R G I** **Rexx** **Interface Function** **Library**

Version 0.1  
May 1996

By Albert A. Modderkolk  
mdd01@ibmnet





## Preface

This function package is intended to be used with the IBM Internet Connection Server for OS/2.

None of the functions are dependent on IBM ICS for OS/2 and should be usable with any other OS/2 HTTP server.

The code has been compiled and linked with the IBM Visual Age C++ V3 compiler. It should therefore be relatively easy to migrate it for use with the IBM ICS product for AIX and MVS.

**Please note!** This is my first attempt at writing a .dll for OS/2. I have tried to ensure that all code is multi-threading but I have not been able to test that so don't rely too much on it till I can delete this disclaimer... If you have any problems, please let me know and I'll try to correct it.

Good Luck! ... and remember, **the bugs are all mine!**

# Table of Contents

## User's Guide

This section describes the available functions, their syntax and their output.

All functions, unless explicitly specified, return a severity level through the Rexx RESULT variable and assign diagnostic information to the following reserved Rexx variables:

RGI_MSGI	contains the message identification, such as RGILDF003I. The messages are listed and explained in the Messages section.
RGI_MTXT	contains the message text.
RGI_RETC	contains the severity level of the message. These are: <ul style="list-style-type: none"><li>0 Information. The function performed as requested.</li><li>4 Warning. The function performed as requested. RGI_MTXT shows the reason for the warning.</li><li>8 Error. Probably a syntax error. The function did not perform as requested. See RGI_MTXT for the error description.</li><li>12 System error. The operating system did not behave as expected. See RGI_MTXT for the error description.</li></ul>

Whenever a function assigns variables with the name XXX.*n*, *n* represents a progressive number starting with 1. The number of variables is contained in XXX.0. allowing you to process the variables with a simple loop:

```
Do i=1 to XXX.0
  Say XXX.i
End i
```

## The HTML Subcommand Environment

HTML text is normally sent from the Rexx CGI application to the browser through the »Say« statement and the LineOut function. RGI implements the »HTML« subcommand which allows you to send HTML text to the browser by simply inserting the text lines in the Rexx procedure.

For example:

```
Address 'HTML'  
'The default environment has now been set to that of the »HTML«'  
'procedure which is implemented in the RGIServ.dll.'  
'All lines that are not recognized by Rexx are sent to that subcommand'  
'processor and printed to STDOUT. Normal Rexx statements and functions'  
'are interpreted as before.'  
timenow=left(time(),5)  
'It is' date() 'and the time is' timenow.'
```

## RGI Functions

### RGIDropFuncs

De-registers the functions to the Rexx programs in this process, thus making them unavailable. RGIDropFuncs also attempts to release the resources that were allocated by the RGILoadFuncs function.

Syntax:

```
RGIDropFuncs ( )
```

where:

count        contains the number of functions that were successfully de-registered.

### RGILoadFuncs

Registers the functions to the Rexx programs in this process, thus making them available to the CGI application.

Syntax:

```
RGILoadFuncs ( )
```

where:

count        contains the number of functions that were successfully registered.

Example:

```
Call RxFuncAdd 'RGILOADFUNCS', 'RGIServ', 'RGILoadFuncs'  
count=RGIFuncAdd ( )
```

Note:

CGI applications are stateless. It would therefore be necessary to call the above functions in each Rexx CGI application to ensure availability. To avoid this, place the above statements in the OS/2 »startup.cmd« file.

### RGIOut

replaces variables in an HTML text string or file with the corresponding values in the caller's Rexx variable pool. The argument is assumed to be a file name if the first argument has at least one period and contains the string »HTM« after the last period.

## Syntax:

```
retcd=RGIOut (arg1 [, arg2])
```

## where:

- retcd** contains the return code.
- arg1** contains the HTML text string or the name of the file that contains the text. There is no limit to the number of records in the file.
- arg2** suppresses direct output to the browser, through »STDOUT:« and, instead, creates variables with the name RGI\_OUT.n.

## Example:

```
timenow=time ()
today=date ()
Call RGIOut 'Today is <&today> and the time here in Rome is <&timenow>.'
```

## Could yield:

```
Today is 5 May 1996 and the time here in Rome is 14:46.
```

## RGIParse

Makes the FORM input available to the CGI application. HTTP special characters (+ and %xx) are translated to ASCII. Both the GET and POST request methods are transparently supported.

## Syntax:

```
Call RGIParse '[-Form] [-Key] [-Init]'
```

## where:

- Form** parses the elements with the template »varnm=value« and generates Rexx variables with the names RGI.varnm and assigns value to them. Multiple values are concatenated and delimited by the »n« character. Elements without a value are assigned the Rexx null value.
- Keyword** parses the elements sequentially and assigns them to the variables RGI\_OUT.n.
- Init** transfers the contents of STDIN to the environment variable QUERY\_CONTENTS if the POST method was used.

More than one option may be specified in the call. RGIParse will return an error message when it is called more than once.

## RGIQEnv

transfers the contents of the CGI environment variables to the caller's Rexx variable pool, using the names RGI.envnm, where envnm is the name of the environment variable.

## Syntax:

```
retcd=RGIQEnv ()
```

The argument »P[rint]« may be passed to have the environment variables sent to STDOUT.

The following environment variables are transferred:

AUTH_TYPE	CONTENT_LENGTH	CONTENT_TYPE
GATEWAY_INTERFACE	HTTP_ACCEPT	HTTP_USER_AGENT



PATH_INFO	PATH_TRANSLATED	QUERY_STRING
REMOTE_ADDRESS	REMOTE_HOST	REMOTE_IDENT
REMOTE_USER	REQUEST_METHOD	SCRIPT_NAME
SERVER_NAME	SERVER_PORT	SERVER_PROTOCOL
SERVER_SOFTWARE		

## RGIQMethod

returns or confirms the request method that is being used to provide the CGI application with the form input.

Syntax:

```
value=RGIQMethod([arg1])
```

Example, assuming that the POST method is used:

```
value=RGIQMethod() /* yields 'POST' */
value=RGIQMethod('POST') /* yields '1'
value=RGIQMethod('GET') /* yields '0'
```

## RGIUtils

Generates the HTTP header and sends it either directly to the browser through »STDOUT:« or generates variables with the name RGI\_HDR.n.

Syntax:

```
retcd=RGIUtils([arg1][,arg2])
```

where:

- arg1** contains the options.
- arg2** suppresses direct output if 'No'.

Options are:

<i>Option</i>	<i>Description</i>	<i>Value</i>	<i>Default when omitted</i>
<b>-ce</b>	Content-encoding	Yes	None
<b>-cl</b>	Content-language.	Yes	None
<b>-ct</b>	Content-type.	Yes	Content-Type: plain/text
<b>-expires</b>	Expiration date of this document. The value must be enclosed in quotes and may contain the time to expiration in terms of <b>years</b> , <b>months</b> , <b>days</b> , <b>hours</b> , <b>minutes</b> and <b>seconds</b> (Minimum abbreviations are shown in bold characters). A value »NOW« generates the time and date of generation of the document.	Yes	None
<b>-extra</b>	This parameter allows to introduce any other non-standard option in the HTTP header.	Yes	None
<b>-length</b>	Content-length	Yes	None
<b>-nocontent</b>	Do not generate a content-type.	No	None
<b>-nodate</b>	Do not generate a document date	No	None
<b>-noemptyline</b>	Do not generate an empty line after the HTTP header	No	None

	line. Allows the caller to add additional header lines.		
<b>-notype</b>	Do not generate a content-type	No	None
<b>-reason</b>	Explains the STATUS code. Can be used only in combination with -Status.	Yes	None
<b>-status</b>	Status code.	Yes	None
<b>-uri</b>	Redirect the browser to another server to retrieve the document there. Requires the -Status option.	Yes	None
<b>-version</b>	Generate the HTTP protocol version.	No	None

#### An example without arguments:

```
retcd=RGIUtils(, 'No')
```

#### could yield:

```
Server: Internet Connection Server/1.0
Content-Type: text/plain
MIME-Version: 1.0
```

#### An example with an expiration date:

```
retcd=RGIUtils(-version -nodate -status 200 -reason 'Virtual document follows'
-expires "3 days 4 hours", "No")
```

#### could yield:

```
HTTP/1.0 200 Virtual document follows
Expires: Thursday, 08-May-96 23:48:11 GMT
Date: Sunday, 05-May-96 19:48:11 GMT
Server: Internet Connection Server/1.0
Content-Type: text/plain
MIME-Version: 1.0
```

#### An example with the »extra« field:

```
retcd=RGIUtils(-CT text/html -S 200 -extra to-user:' Thanks for browsing this
site!', 'No')
```

#### could yield:

```
Server: Internet Connection Server/1.0
Content-Type: text/html
MIME-Version: 1.0
to-user: Thanks for browsing this site!
```



# Messages

<i>Msg Id</i>	<i>Message Text</i>	<i>Sev</i>	<i>Description</i>
RGIRGI000I	OK!	0	Function performed as requested.
RGIRGI001E	Error	8	An error occurred, no description available.
RGIDRF002I	&1 functions were dropped.	0	The RGI functions were de-registered. Those which are registered in another process are not counted.
RGILDF003I	&1 functions were loaded.	0	The RGI functions were registered. Those which were already registered are not counted.
RGIOUT004E	No arguments passed. At least one is required.	8	RGIOut requires a text string or HTML file name.
RGIOUT005E	Unable to open &1.	8	RGIOut deduced that the string is a file name but the file could not be opened.
RGIPAR006W	RGIParse has already been called from this CGI application.	4	RGIParse has already been called in this CGI application.
RGIPAR007S	Request is 'GET' but the QUERY_CONTENTS environment variable does not exist.	12	Probably an HTTP server error. The QUERY_CONTENTS environment variable should have been set by the HTTP server.
RGIPAR008S	The REQUEST_METHOD environment variable does not exist.	12	Probably an HTTP server error. The REQUEST_METHOD environment should have been set by the HTTP server.
RGIPAR009S	Short on memory.	12	Unable to allocate the requested memory. None of the RGI functions require an excessive amount of memory. It is probably best to reboot or restart the HTTP server.
RGIPAR010W	There are no input elements.	4	There is no input for the CGI application.
RGISAV011E	RGISwapSave requires two arguments: a unique id and at least one name or stem of the variables to save.	8	Not yet available
RGISWP012S	Unable to create the semaphore for the swapping functions, return code is &1.	12	Not yet available
RGISWP013S	Unable to open/request use of the swap semaphore.	12	Not yet available
RGISWP014S	Unable to release the swap semaphore.	12	Not yet available
RGISWP015E	RGISwapOpen has not been issued.	8	Not yet available
RGISWP016I	Swap functions closed.	0	Not yet available
RGISWP017E	&1 requires &2 argument(s).	8	Not yet available
RGISWP018E	Id &1 not found.	8	Not yet available
RGIPAR019E	Option -&1 not supported.	8	The shown option is not supported by the RGIParse function. The function is not executed.
RGIQME020E	REQUEST_METHOD environment variable not available.	8	The RGIQMethod function could not determine the input method because the environment variable has not been set by the HTTP server.
	RGIUtills requires at least one argument.	8	At least one option must be specified in argument 1.
RGIUUTI022E	&1 is not a valid argument.	8	The shown argument is not a valid RGIUtil argument.
RGIUUTI023E	The last argument should have had a value.	8	The last RGIUtil argument requires a value.
RGIUUTI024S	Environment variable &1 is not available.	12	The shown environment has not been set.
	The Reason argument can only be used when also the Status argument is supplied.	8	A reason is supplied but not the status code.
RGIUUTI026E	Expires argument &1 should be numeric.	8	The number of years, months, days, etc. Is not numeric.

The &n variables in the message are dynamically resolved with the appropriate value.

## Installation

Unzip the RGIServ.zip package in a temporary directory.

The package consists of the following files:

<i>File Name</i>	<i>Description</i>
RGIServ.dll	Executable file containing the RGI functions.
RGIServ.sdw	This document in native StarWriter 3.0 format.
RGIServ.rtf	This document in Rich Text Format.
RGI_Test.html	Installation Test Document
RGI_Tn.cmd	Installation Test Scripts ( <i>n</i> is in the range 1-5)

### **RGIServ.dll**

Copy this file to a directory listed in your LIBPATH statement in your config.sys file.

### **Installation Test Scenarios**

Copy RGI\_Test.html to the HTML library (usually \WWW\HTML) and the RGI\_Tn.cmd files to the CGI applications library (usually \WWW\CGI-BIN).

### **RGIServ Initialization**

Create a .cmd file and add the following statements to it:

```
Call RxFuncAdd 'RGILOADFUNCS', 'RGIserve', 'RGILOADFUNCS'  
Call RGIloadFuncs  
Say RGI_RETC RGI_MSGI RGI_MTXT
```

The latter statement is optional and will show, if completed successfully, the message »*n* functions were loaded« and a return code '0'. Purpose of these statements is to load the RGIServ library and to keep it loaded in memory. Execute this procedure before you invoke the IBM Internet Connection Server and keep this session active by, for example, minimizing the window. (See Known Bugs).

### **Config.sys**

Add a SET statement to indicate where RGIOut should retrieve the HTML files to resolve. Default is the IICS server's document root, specified in the IICS DOCUMENT\_ROOT environment variable. If you are using another server, add a statement like:

```
SET RGI_DOCUMENT_ROOT=E:\WWW\HTML\
```

**Reboot before trying to use the RGI functions!**

### **Installation Test**

After rebooting the system, you may want to run the installation test. Load the document »RGI\_TEST« which will guide you through several CGI scripts that show the RGI functions and how to use the HTML subcommand. (See *known bugs*)

## **Known Bugs**

### ***Expiration Time***

The expiration time that is generated by the »EXP« parameter of the RGIUtils function is off by two hours. Although my config.sys has a line »SET TZ CET-1SET« (i.e., I am one hour east of UTC) the gmtime() function returns a time that is 3 hours later than it should. Please let me know if it works well on your system so that I can determine whether I have to look more carefully at the SET statement.

### ***Installation Scripts***

The intent was to have a document »RGI\_TEST.HTML« which triggers the execution of 5 Rexx CGI procedures through »action« tags. For reasons that I do not understand, this does not work.

RGI\_Test/html will invoke RGI\_T1.cmd but the output of RGI\_T1.cmd will not invoke RGI\_T2.cmd, etc...

Till somebody can tell me what I am doing wrong, you will have to execute each script by loading the »RGI\_Tn.html« (where *n* is in the range 2-5) document. RGI\_T3.cmd will therefore not resolve the <&RGI.FNAME> and <&RGI.CNAME> variables.

### ***Table of Contents***

The table of Contents of this document is empty and page numbering does not work. I have contacted Star Division but they are not fast repliers...

### ***RGIServ Initialization***

This is more a usability problem than a real bug... The IBM Internet Connection Server has no exit for initialization when it starts-up. Until it has such an exit (if ever) I would like to call the IICS from a Rexx procedure that initializes the RGIServ library and then calls IICS. However, the IICS settings notebook does not contain a »program« page and therefore, till I know to invoke IICS, you will have to live with running that small procedure before starting IICS.