

G2 Web

User's Guide

Version 2.3 Rev. 0



G2 Web User's Guide, Version 2.3 Rev. 0

November 2013

The information in this publication is subject to change without notice and does not represent a commitment by Gensym Corporation.

Although this software has been extensively tested, Gensym cannot guarantee error-free performance in all applications. Accordingly, use of the software is at the customer's sole risk.

Copyright (c) 1985-2013 Gensym Corporation

All rights reserved. No part of this document may be reproduced, stored in a retrieval system, translated, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of Gensym Corporation.

Gensym®, G2®, Optegrity®, and ReThink® are registered trademarks of Gensym Corporation.

NeurOn-Line™, Dynamic Scheduling™, G2 Real-Time Expert System™, G2 ActiveXLink™, G2 BeanBuilder™, G2 CORBALink™, G2 Diagnostic Assistant™, G2 Gateway™, G2 GUIDE™, G2GL™, G2 JavaLink™, G2 ProTools™, GDA™, GFI™, GSI™, ICP™, Integrity™, and SymCure™ are trademarks of Gensym Corporation.

Telewindows is a trademark or registered trademark of Microsoft Corporation in the United States and/or other countries. Telewindows is used by Gensym Corporation under license from owner.

This software is based in part on the work of the Independent JPEG Group.

Copyright (c) 1998-2002 Daniel Veillard. All Rights Reserved.

SCOR® is a registered trademark of PRTM.

License for Scintilla and SciTE, Copyright 1998-2003 by Neil Hodgson, All Rights Reserved.

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (<http://www.openssl.org/>).

All other products or services mentioned in this document are identified by the trademarks or service marks of their respective companies or organizations, and Gensym Corporation disclaims any responsibility for specifying which marks are owned by which companies or organizations.

Gensym Corporation
52 Second Avenue
Burlington, MA 01803 USA
Telephone: (781) 265-7100
Fax: (781) 265-7101

Part Number: DOC013-230

Contents

| | | |
|------------------|-------------------------------|-----------|
| | Preface | v |
| | About this Guide | v |
| | Audience | v |
| | Conventions | vi |
| | Related Documentation | vii |
| | Customer Support Services | x |
| Chapter 1 | Introduction to G2 Web | 1 |
| | Introduction | 1 |
| | Loading GWEB | 2 |
| Chapter 2 | Module Settings | 3 |
| | Introduction | 3 |
| | gweb-module-settings | 4 |
| Chapter 3 | Built-in Web Support | 7 |
| | Introduction | 7 |
| | Web Pages | 8 |
| | Built-in Web Pages | 9 |
| | SOAP Operations | 12 |
| | Basic SOAP Demo | 13 |
| | Operator Message Demo | 15 |
| | WSDL Demo | 17 |
| Chapter 4 | Web Servers | 21 |
| | Introduction | 21 |
| | G2-Built HTTP Server | 23 |
| | Class | 23 |
| | APIs | 25 |
| | Example | 27 |

| | |
|--|----|
| G2 WebLink HTTP Server | 28 |
| Class | 28 |
| APIs | 30 |
| URL Requests | 33 |
| Base Response Object | 34 |
| URL Response Object from Files | 38 |
| URL Response Object with Tag Replacement | 39 |
| URL Response Object from Workspace | 40 |
| Dynamic URL Response Object | 41 |
| SOAP Operation | 42 |
| WSDL | 43 |
| G2GL Operation | 44 |
| Custom Response Object | 46 |
| User Sessions | 48 |
| Class | 48 |
| APIs | 49 |

Chapter 5 HTML Support 51

| | |
|---------------------------------|----|
| Introduction | 51 |
| HTML Pages | 52 |
| Layout and Formatting | 53 |
| Hyperlinks | 54 |
| Images | 54 |
| Text Formatting | 55 |
| Lists | 56 |
| Tables | 56 |
| Frames | 58 |
| Forms | 59 |
| Creating HTML Content | 61 |
| Extracting Form Response Values | 61 |

Index 63

Preface

Describes this document and the conventions that it uses.

| | |
|---------------------------|-----|
| About this Guide | v |
| Audience | v |
| Conventions | vi |
| Related Documentation | vii |
| Customer Support Services | x |



About this Guide

This guide describes the G2 Web (GWEB) module, which provides out-of-the-box support for displaying Web pages from within G2 and provides an API for creating Web servers, making SOAP requests, and creating HTML pages.

Audience

This guide is for G2 developers who want to customize applications, using a set of standard application programmers' interface (API) procedures and methods, and built-in classes. It assumes familiarity with the G2 procedure language.

Conventions

This guide uses the following typographic conventions and conventions for defining system procedures.

Typographic

| Convention Examples | Description |
|--|--|
| g2-window, g2-window-1, ws-top-level, sys-mod | User-defined and system-defined G2 class names, instance names, workspace names, and module names |
| history-keeping-spec, temperature | User-defined and system-defined G2 attribute names |
| true, 1.234, ok, "Burlington, MA" | G2 attribute values and values specified or viewed through dialogs |
| Main Menu > Start KB Workspace > New Object create subworkspace Start Procedure | G2 menu choices and button labels |
| conclude that the x of y ... | Text of G2 procedures, methods, functions, formulas, and expressions |
| <i>new-argument</i> | User-specified values in syntax descriptions |
| <u>text-string</u> | Return values of G2 procedures and methods in syntax descriptions |
| File Name, OK, Apply, Cancel, General, Edit Scroll Area | GUIDE and native dialog fields, button labels, tabs, and titles |
| File > Save Properties | GMS and native menu choices |
| workspace | Glossary terms |

| Convention Examples | Description |
|---|-------------------------------------|
| <i>c:\Program Files\Gensym\</i> | Windows pathnames |
| <i>/usr/gensym/g2/kbs</i> | UNIX pathnames |
| <i>spreadsh.kb</i> | File names |
| <i>g2 -kb top.kb</i> | Operating system commands |
| <i>public void main() gsi_start</i> | Java, C and all other external code |

Note Syntax conventions are fully described in the *G2 Reference Manual*.

Procedure Signatures

A procedure signature is a complete syntactic summary of a procedure or method. A procedure signature shows values supplied by the user in *italics*, and the value (if any) returned by the procedure underlined. Each value is followed by its type:

```
g2-clone-and-transfer-objects
(list: class item-list, to-workspace: class kb-workspace,
 delta-x: integer, delta-y: integer)
-> transferred-items: g2-list
```

Related Documentation

G2 Core Technology

- *G2 Bundle Release Notes*
- *Getting Started with G2 Tutorials*
- *G2 Reference Manual*
- *G2 Language Reference Card*
- *G2 Developer? Guide*
- *G2 System Procedures Reference Manual*

- *G2 System Procedures Reference Card*
- *G2 Class Reference Manual*
- *Telewindows User? Guide*
- *G2 Gateway Bridge Developer? Guide*

G2 Utilities

- *G2 ProTools User? Guide*
- *G2 Foundation Resources User? Guide*
- *G2 Menu System User? Guide*
- *G2 XL Spreadsheet User? Guide*
- *G2 Dynamic Displays User? Guide*
- *G2 Developer? Interface User? Guide*
- *G2 OnLine Documentation Developer? Guide*
- *G2 OnLine Documentation User? Guide*
- *G2 GUIDE User? Guide*
- *G2 GUIDE/UII Procedures Reference Manual*

G2 Developers' Utilities

- *Business Process Management System User? Guide*
- *Business Rules Management System User? Guide*
- *G2 Reporting Engine User? Guide*
- *G2 Web User? Guide*
- *G2 Event and Data Processing User? Guide*
- *G2 Run-Time Library User? Guide*
- *G2 Event Manager User? Guide*
- *G2 Dialog Utility User? Guide*
- *G2 Data Source Manager User? Guide*
- *G2 Data Point Manager User? Guide*
- *G2 Engineering Unit Conversion User? Guide*
- *G2 Error Handling Foundation User? Guide*
- *G2 Relation Browser User? Guide*

Bridges and External Systems

- *G2 ActiveXLink User? Guide*
- *G2 CORBALink User? Guide*
- *G2 Database Bridge User? Guide*
- *G2-ODBC Bridge Release Notes*
- *G2-Oracle Bridge Release Notes*
- *G2-Sybase Bridge Release Notes*
- *G2 JMail Bridge User? Guide*
- *G2 Java Socket Manager User? Guide*
- *G2 JMSLink User? Guide*
- *G2-OPC Client Bridge User? Guide*
- *G2 PI Bridge User? Guide*
- *G2-SNMP Bridge User? Guide*
- *G2-HLA Bridge User? Guide*
- *G2 WebLink User? Guide*

G2 JavaLink

- *G2 JavaLink User? Guide*
- *G2 DownloadInterfaces User? Guide*
- *G2 Bean Builder User? Guide*

G2 Diagnostic Assistant

- *GDA User? Guide*
- *GDA Reference Manual*
- *GDA API Reference*

Customer Support Services

You can obtain help with this or any Gensym product from Gensym Customer Support. Help is available online, by telephone, by fax, and by email.

To obtain customer support online:

➔ Access G2 HelpLink at www.gensym-support.com.

You will be asked to log in to an existing account or create a new account if necessary. G2 HelpLink allows you to:

- Register your question with Customer Support by creating an Issue.
- Query, link to, and review existing issues.
- Share issues with other users in your group.
- Query for Bugs, Suggestions, and Resolutions.

To obtain customer support by telephone, fax, or email:

➔ Use the following numbers and addresses:

| | Americas | Europe, Middle-East, Africa (EMEA) |
|--------------|--|--|
| Phone | (781) 265-7301 | +31-71-5682622 |
| Fax | (781) 265-7255 | +31-71-5682621 |
| Email | service@gensym.com | service-ema@gensym.com |

Introduction to G2 Web

Provides an overview of the G2 Web (GWEB) module.



Introduction

G2 Web (GWEB) defines out-of-the-box Web pages and SOAP services, as well as classes and APIs enabling G2 to implement an HTTP server and serve HTML pages, XML structures, SOAP services, and files. URLs are matched against request objects, which are instances of `gweb-request` and its subclasses and are defined for each URL that G2 will serve.

GWEB supports using both G2 WebLink and the G2 integrated HTTP server. The APIs for both are similar, including defining Web pages providing a level of abstraction to use either HTTP server. The only difference is that the G2 built-in HTTP server supports SSL, SOAP, and WSDL, whereas the G2 WebLink HTTP server does not.

As a summary, GWEB supports:

- Two HTTP servers with an abstraction to switch from one to the other.
- Defining and implementing Web request objects used to serve specific URL requests. Web requests may be fulfilled by sending the content of files, HTML pages, Comma Separated Values (CVS), or XML documents, or as SOAP documents.

Loading GWEB

To use the GWEB module, you must load or merge in *gweb.kb*, which is located in the *g2i\kbs* directory.

Module Settings

Describes the G2 Web (GWEB) module settings.

Introduction 3

gweb-module-settings 4



Introduction

The gweb-module-settings object inherits GFR module settings. Upon startup, GFR locates one module settings object as the active setting, which is typically the instance in the highest level module. The active module is determined when G2 is started. Several APIs take the active module settings object into account during execution.

gweb-module-settings

Manages system configurations for the GWEB module.

Class Inheritance Path

gweb-module-settings, gfr-module-settings, object, item

Attributes

| Attribute | Description |
|---------------------------------|---|
| require-login | Whether connecting to the HTTP server requires a login. |
| <i>Allowable values:</i> | truth-value |
| <i>Default value:</i> | true |
| allow-anonymous-login | Whether to allow anonymous logins. |
| <i>Allowable values:</i> | truth-value |
| <i>Default value:</i> | true |
| session-timeout | The timeout for disconnecting from the HTTP server when idle. |
| <i>Allowable values:</i> | integer, formatted as an interval |
| <i>Default value:</i> | 10 minutes |
| session-maximum-duration | The maximum duration for the HTTP session. |
| <i>Allowable values:</i> | integer, formatted as an interval |
| <i>Default value:</i> | 8 hours |

| Attribute | Description |
|--|--|
| auto-updating-page-refresh-rate | The default value for configuring the interval for refreshing Web pages. Note that configuring this value does not enforce an automatic refresh for each page. |
| <i>Allowable values:</i> | integer, formatted as an interval |
| <i>Default value:</i> | 120 |
| server-http-status-descriptions | A symbol naming a tabular-function-of-1-arg that stores the HTTP server status descriptions. |
| <i>Allowable values:</i> | symbol |
| <i>Default value:</i> | gweb-http-status-description-of-code |

Built-in Web Support

Describes the built-in Web support in the GWEB module.

| | |
|-----------------|----|
| Introduction | 7 |
| Web Pages | 8 |
| SOAP Operations | 12 |
| WSDL Demo | 17 |



Introduction

GWEB includes a few URL response objects providing out-of-the-box Web pages and Web services enabling users to view and interact with operator messages and tabular reports. You can access the built-in Web pages and use them as a starting point for building your own Web pages.

Prior to accessing the built-in Web pages and SOAP operations, you need to configure and enable an HTTP server. You can either use G2 WebLink or the built-in G2 server, although note that SOAP operations are only supported when using the built-in G2 HTTP server. You must also ensure that you configure the `http-server-root-directory` attribute of the HTTP server to "`install-dir\g2\data`". This directory contains images and a CAB file that the HTTP server will serve upon request.

Web Pages

GWEB includes a simple Web-based user interface enabling you to:

- Log in to the G2 server.
- Display several Web pages.
- Display a browser of the user's message queue if GEVM is merged in, including viewing message details, deleting messages, and acknowledging messages.
- Display a list of defined reports and viewing the content of the report in tabular format
- Launch a connection to the G2 server, using Telewindows Next Generation, including downloading the required software on the client.

To enable this functionality, start an HTTP server, then use the following URL to access the login page:

```
http://g2-server:g2-http-port/login
```

For example:

```
http://my-host:8085/login
```

Tip Choose Start > Programs > Gensym G2 8.3r0 > Examples > G2 Developers' Utilities > G2 Server Console to go to the default G2 server login page.

For an example of creating and connecting an HTTP server, see [Example](#).

The [built-in web pages](#) provide a simple starting point for building your own Web-based user interface. The user navigates to the individual pages by selecting choices in a combo box. The following figures show some examples of the built-in Web pages.

Built-in Web Pages

Here are the built-in Web pages that GWEB provides:

Login Page:



Home page:



You connected to the G2 server running version Version 8.3 Rev. 0 Intel NT. Out of the box this server supports a few prebuilt pages, but provides a framework to easily customize and extend it. To navigate through the different pages accessible from this server, select an option in the top right combobox.

You may also access the server using Telewindows. The connection to the server with automatic download of the latest Telewindows software is accessible from the combobox as well. This enables zero installation and maintenance on the client side and greatly simplifies deployment.

Message Browser:

The following table lists the content of the messages stored in message queue anonymous.messages.

| Date-Time | Category | Actions | Message |
|---------------------|------------------------------------|---|---|
| 10/19/2006 13:47:04 | Network Connection | Del Ack | Network interface gdem-sql-db has lost its connection to the bridge process. |
| 10/19/2006 13:47:04 | Network Connection | Del Ack | Network interface offline-db-interface has lost its connection to the bridge process. |

Message detail:

Message details
GDSM-SQL-DB - Network Connection

Type: Internal Fault
Message: Network interface gdsm-sql-db has lost its connection to the bridge process.
Detail:
Advice: Contact your administrator about loss of connection of the network interface gdsm-sql-db.
Comment:
Priority: 9
Repetition: 1
Target: GDSM-SQL-DB
Initiator: GDSM-SQL-DB
Created: 10/19/2006 13:47:04
Last Updated: 10/19/2006 13:47:04
Assigned To: unspecified
Acknowledged By: unspecified
Acknowledged When:

You may access the [home](#) page or the list of messages in the [message](#) queue.

Reports:

The following table lists all available reports.

| Type | Label | Description |
|------------------------|--|-------------|
| GRPE-ALARM-REPORT | my Alarm Report | |
| GRPE-DATASERIES-REPORT | Heater dataseries report | |

Report detail:



ANONYMOUS

Heater dataseries report

| Time | fo2-fo-furnace-damper-process-variable | FO2-10-FUEL-PRESSURE-PROCESS-VARIABLE | FO2-10-TEMPERATURE-102-SENSOR-PROCESS-VARIABLE | FO2-10-FUEL-CONTROL-VALVE-PROCESS-VARIABLE | FO2-10-PUMP1-PROCESS-VARIABLE | FO2-10-FLOW-SENSOR-101-PROCESS-VARIABLE | FO2-10-FLOW-SENSOR-102-PROCESS-VARIABLE | FO2-10-FLOW-SENSOR-103-PROCESS-VARIABLE |
|-------|--|---------------------------------------|--|--|-------------------------------|---|---|---|
| 0 | 2.5 | 62.986 | 497.673 | 68.81 | 154.97 | 74.949 | 71.475 | |
| 0.4 | 2.5 | 69.863 | 538.293 | 69.641 | 184.471 | 71.982 | 69.04 | |
| 0.583 | 2.5 | 65.599 | 539.081 | 70.361 | 162.751 | 69.999 | 69.332 | |
| 0.85 | 2.5 | 60.45 | 509.207 | 70.188 | 183.23 | 68.685 | 72.872 | |
| 1.067 | 2.5 | 67.51 | 501.302 | 73.91 | 194.056 | 67.955 | 68.905 | |
| 1.25 | 2.5 | 62.788 | 513.968 | 68.162 | 196.068 | 69.323 | 66.466 | |


Downloading TWNG CAB and auto connecting to server:



ANONYMOUS Jump to: Select Index

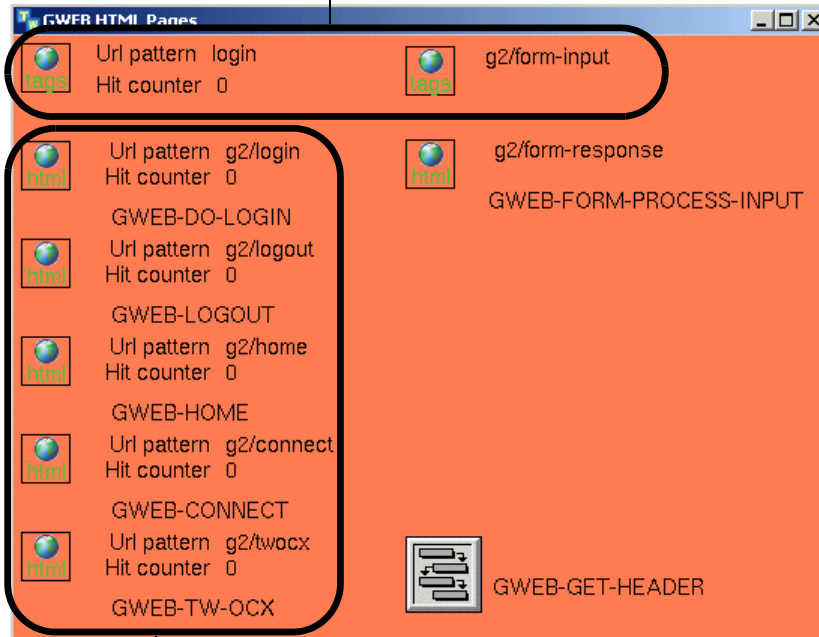
Checking for the latest version of the software

Depending on your connection speed, this might take a minute. During this time, you may receive one or more security warnings. Review each security warning to ensure that the content is signed by Gensym, and then click **Yes** to install the software.



To display the objects used to provide these built-in Web pages, see `gweb-top-level > Programmers' Interface > HTML Pages`:

Instances of `gweb-request-serve-with-tag-replacement` for making login requests and inputting values into forms.



Instances of `gweb-request-dynamic-response` for displaying the built-in Web pages.

SOAP Operations

GWEB provides two sample SOAP operations. The examples contain action buttons to make the SOAP requests from G2; however, the requests could also be called from another G2 or from an application written in Visual Basic, C, C#, or Java, for example.

To enable this functionality, start a HTTP server, then use the following URL to access demo page:

```
http://g2-server:g2-http-port/soap/demo
```

For example:

```
http://my-host:8085/soap/demo
```

For an example of creating and connecting an HTTP server, see [Example](#).

Basic SOAP Demo

The basic SOAP example demonstrates how to perform an operation between two numbers. You can add, subtract, multiply, or divide two numbers supplied in the request, and the result is returned as a SOAP message.

To access the demo, choose **gweb-top-level > Programmers' Interface > SOAP Operations**:



The screenshot shows a window titled "GWEB SOAP Operations" with a list of operations and their configurations:

- GWEB-CALL-SOAP-DEMO-REQUEST**: Represented by a SOAP icon.
- GWEB-REMOVE-WHITESPACE**: Represented by a SOAP icon.
- Url pattern soap/demo**: Represented by a SOAP icon.
- Hit counter 0**: Represented by a SOAP icon.
- GWEB-SOAP-BASIC-HANDLER**: Represented by a SOAP icon.
- gweb_demo_input_msg**: Represented by a MSG icon.
- gweb_demo_output_msg**: Represented by a MSG icon.

Below the list, there are two sections for adding values:

Add values

```
start gweb-call-soap-demo-
request("http://localhost:8085/soap/demo",
structure(
action: "http://gensym.com/soap/demo",
body-entries: sequence(
structure(tag-name: "operation", children:
sequence("add")),
structure(tag-name: "arg-1", children:
sequence("[random(1,1000)]")),
structure(tag-name: "arg-2", children:
sequence("[random(1,1000)]"))
)))
```

Subtract Values

```
start gweb-call-soap-demo-
request("http://localhost:8085/soap/demo",
structure(
action: "http://gensym.com/soap/demo",
body-entries: sequence(
structure(tag-name: "operation", children:
sequence("sub")),
structure(tag-name: "arg-1", children:
sequence("[random(1,1000)]")),
structure(tag-name: "arg-2", children:
sequence("[random(1,1000)]"))
)))
```

A **gweb-request-soap-operation** whose **url-pattern** is **soap/demo**.

The **gweb-soap-basic-handler** is a **gweb-request-soap-operation**, which implements a SOAP handler for basic SOAP requests whose URL pattern is **soap/demo**. For details, see [SOAP Operation](#).

Here is the `gweb-soap-basic-handler` SOAP handler:

```
gweb-soap-basic-handler(http-server: class item, soap-request: structure,
    session: class gweb-user-session) = (structure)
soap-reply: structure = structure();
operation, arg1, arg2: text;
elt: structure;
result: quantity;
begin
  for elt = each structure in the body-entries of soap-request do
    if the tag-name of elt = "operation" then operation = the first text in the children
      of elt;
    if the tag-name of elt = "arg-1" then arg1 = the first text in the children of elt;
    if the tag-name of elt = "arg-2" then arg2 = the first text in the children of elt;
  end;
  case (operation) of
    "add": result = quantity(arg1) + quantity(arg2);
    "sub": result = quantity(arg1) - quantity(arg2);
    "mult": result = quantity(arg1) * quantity(arg2);
    "div": result = quantity(arg1) / quantity(arg2);
    otherwise: result = 0.0;
  end;
  { --- Build reply }
  soap-reply = structure(body-entries: sequence(
    structure(tag-name: "operation", children: sequence ( operation) ),
    structure(tag-name: "arg1", children: sequence ( arg1)),
    structure(tag-name: "arg2", children: sequence (arg2)),
    structure(tag-name: "result", children: sequence ("[result]"))));
  return soap-reply;
end
```

Here is the `gweb-call-soap-demo-request` procedure, which calls `g2-send-soap-request` to send a SOAP request on a given URL. For details on this system procedure, see the *G2 System Procedures Reference Manual*.

```
gweb-call-soap-demo-request(URL: text, request: structure)
response: structure;
body-entries: sequence;
begin
  { --- Call the operation }
  response = call g2-send-soap-request(URL, request);
  body-entries = call gweb-remove-whitespace(the body-entries of
    response);
  post "SOAP response
    [body-entries]";
end
```

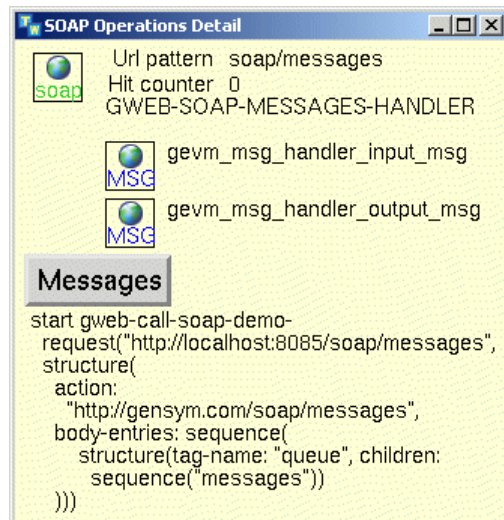

Here is the result of clicking the Add Values button:

```
#151 2:57:27 p.m. SOAP response
sequence (structure (TAG-NAME: "operation",
  CHILDREN: sequence ("add")),
  structure (TAG-NAME: "arg1",
    CHILDREN: sequence ("325")),
  structure (TAG-NAME: "arg2",
    CHILDREN: sequence ("629")),
  structure (TAG-NAME: "result",
    CHILDREN: sequence ("954")))
```

Operator Message Demo

This example enables you to query the message content of a GEVM message queue. The SOAP request specifies the queue label, and the SOAP reply contains the key attributes of all messages contained in the queue in XML format. This demo requires that you merge in *gevm.kb*.

To access the demo, choose **gevm-top-level > Programmers' Interface > SOAP Operations**:



The `gweb-soap-messages-handler` is a `gweb-request-soap-operation`, which implements a SOAP handler that handles GEVM messages whose URL pattern is `soap/demo`. For details, see [SOAP Operation](#).

Here is the gweb-soap-messages-handler SOAP handler:

```
gweb-soap-messages-handler(http-server: class item, soap-request: structure,
    session: class gweb-user-session) = (structure)
soap-reply: structure = structure();
key, queue-key: text;
queue-exists: truth-value;
queue: class gevmm-gqs-queue;
iv: item-or-value;
events: sequence;
event: class gevmm-event;
event-exists: truth-value;
events-xml: sequence = sequence ( );
elt: structure;
grtl-settings: class grtl-module-settings;
Target, Source: class item;
creationTimeStamp, updateTimeStamp, ackTimeStamp, eventType, eventText,
targetName, SourceName, entryType, category, msg, detail, advice, assigned-to,
acked-by, comment: text;
begin
    grtl-settings = call gfr-get-active-setting (the symbol grtl-module-settings,
        gfr-default-window);
    { --- Fetch queue }
    queue-key = "";
    for elt = each structure in the body-entries of soap-request do
        if the tag-name of elt = "queue" then queue-key = the first text in the children
            of elt;
    end;
    queue-exists, iv = call grtl-get-item-by-key-with-existence-check(
        gevmm-gqs-queue, queue-key, false);
    if queue-exists and iv is a gevmm-gqs-queue then begin
        queue = iv;
        events = call gevmm-get-collected-events(queue);
        for event = each gevmm-event in events do
            creationTimeStamp = call grtl-g2-time-to-isox-date-time(the
                gevmm-creation-timestamp of Event);
            category = call grtl-remove-enclosing-quotes-from-text (
                the gevmm-category of Event);
            msg = call grtl-remove-enclosing-quotes-from-text (the gevmm-message of
                Event);
            if the gevmm-acknowledgement-required of Event and the gevmm-acknowledged
                of Event then
                ackTimeStamp = call grtl-g2-time-to-isox-date-time (
                    the gevmm-acknowledged-timestamp of Event)
            else
                ackTimeStamp = "";
            events-xml = insert-at-end(events-xml, structure (
                tag-name: "message", children: sequence (
                    structure ( tab-name: "key", children: sequence (
                        the gevmm-key of event )),
                    structure ( tab-name: "CreationTimestamp", children: sequence (
```

```

        creationTimestamp )),
        structure ( tab-name: "Category", children: sequence ( category )),
        structure ( tab-name: "Message", children: sequence ( msg )),
        structure ( tab-name: "AcknowledgementTimestamp", children:
            sequence ( ackTimeStamp ))));
    end;
end;

{ --- Build reply }
soap-reply = structure(body-entries: sequence(
    structure(tag-name: "Queue", children: sequence ( queue-key) ),
    structure(tag-name: "Messages", children: sequence (events-xml ))));
return soap-reply;
end

```

WSDL Demo

You can use the following URL to dynamically query a WSDL for the current gweb-request configuration in G2:

<http://g2-server:g2-http-port/wsdl>

For example, using default configurations, you can use:

<http://localhost:8085/wsdl>

Below is an WSDL example returned by this query:

```

<?xml version="1.0" encoding='utf-8' ?>

<definitions targetNamespace="http://g2tempuri.org/"
xmlns:tns="http://g2tempuri.org/" xmlns:http="http://schemas.xmlsoap.org/wsdl/http/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:wSDL="http://schemas.xmlsoap.org/wsdl/" xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" xmlns="http://schemas.xmlsoap.org/wsdl/">
  <message name="Gevm_Msg_Handler_Input_Msg">
    <part name="queue" type="xsd:string" />
  </message>
  <message name="Gevm_Msg_Handler_Output_Msg" />
  <message name="Gweb_Demo_Input_Msg">
    <part name="operation" type="xsd:string" />
    <part name="arg-1" type="xsd:double" />
    <part name="arg-2" type="xsd:double" />
  </message>
  <message name="Gweb_Demo_Output_Msg">
    <part name="operation" type="xsd:string" />
    <part name="arg-1" type="xsd:double" />
  </message>

```

```

    <part name="arg-2" type="xsd:double" />
    <part name="result" type="xsd:double" />
  </message>
  <portType name="Gevm">
    <operation name="Get-Messages">
      <documentation>This is a SOAP operation demonstrates how to query
operator messages contained in a gevm-gqs-queue.</documentation>
      <input message="tns:Gevm_Msg_Handler_Input_Msg" />
      <output message="tns:Gevm_Msg_Handler_Output_Msg" />
    </operation>
  </portType>
  <portType name="Gweb_Demo">
    <operation name="Demo">
      <documentation>This is a basic demonstration to build SOAP handlers
in G2. Note that if configured correctly, these SOAP handlers are
automaticall included in G2 auto generated WSDL.</documentation>
      <input message="tns:Gweb_Demo_Input_Msg" />
      <output message="tns:Gweb_Demo_Output_Msg" />
    </operation>
  </portType>
  <binding name="GevmSoapDocument" type="tns:Gevm">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http"
style="document" />
    <operation name="Get-Messages">
      <soap:operation soapAction="Gensym.Gweb.WebServices/Get-Messages"
style="document" />
      <input>
        <soap:body use="encoded" namespace="http://g2tempuri.org"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
      </input>
      <output>
        <soap:body use="encoded" namespace="http://g2tempuri.org"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
      </output>
    </operation>
  </binding>
  <binding name="Gweb_DemoSoapDocument" type="tns:Gweb_Demo">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http"
style="document" />
    <operation name="Demo">
      <soap:operation soapAction="Gensym.Gweb.WebServices/Demo"
style="document" />
      <input>
        <soap:body use="encoded" namespace="http://g2tempuri.org"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
      </input>
      <output>
        <soap:body use="encoded" namespace="http://g2tempuri.org"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
      </output>
    </operation>
  </binding>

```

```
</binding>
<service name="G2Services">
  <port name="GevmSoapDocumentService" binding="tns:GevmSoapDocument">
    <soap:address location="http://pprintz-1164:8085/soap/messages" />
  </port>
  <port name="Gweb_DemoSoapDocumentService" binding="tns:Gweb_
DemoSoapDocument">
    <soap:address location="http://pprintz-1164:8085/soap/demo" />
  </port>
</service>
</definitions>
```


Web Servers

Describes how to create and connect to the HTTP servers, service URL requests, and manage user sessions.

| | |
|------------------------|----|
| Introduction | 21 |
| G2-Built HTTP Server | 23 |
| G2 WebLink HTTP Server | 28 |
| URL Requests | 33 |
| User Sessions | 48 |



Introduction

GWEB provides two types of HTTP servers:

- G2 built-in HTTP servers, which require no external bridge to run and support secure communication, using SSL, and SOAP requests.
- G2 WebLink based HTTP servers, which require a connection to a G2 WebLink to run.

Note The G2 WebLink HTTP server does not support secure communication or SOAP requests. Also, it does not support the HTTP *POST* method, which is often used in HTTP forms, whereas while the G2-built HTTP server supports both *POST* and *GET* methods.

Once you have started an HTTP server, you can send URL requests, which the server handles automatically, based on pattern matching. You define URL

requests accessible by the HTTP server by creating request objects, which are instances of `gweb-request-object` or its subclasses. You define the URL requests by using the following techniques:

- The contents of a file, where the file reader uses a text conversion style specified in request object to service the request.
- A predefined document in which tags are replaced with values, where the file reader uses the US-ASCII text conversion style.
- A G2 workspace embedded in an HTTP page.
- A procedure that dynamically computes a response.
- A procedure that dynamically computes a SOAP response.
- A procedure that dynamically computes a custom response.

GWEB creates a user session to keep track of logins to HTTP servers and grant access to URL requests.

G2-Built HTTP Server

The `gweb-g2-http-server` class defines and configures an HTTP server that is based on the G2-built HTTP server. It requires no external bridge to run, and supports secure communication and authentication, based on SSL.

Class

Class Inheritance Path

`gweb-g2-http-server`, object

Attributes

| Attribute | Description |
|---|--|
| default-page | The default text for the Web page. <i>Allowable values:</i> text, formatted as free text <i>Default value:</i> "default" |
| logging-enabled | Whether logging is enabled. <i>Allowable values:</i> truth-value <i>Default value:</i> false |
| add-http-request-attributes-to-log | Whether to add HTTP request attribute to the log file. <i>Allowable values:</i> truth-value <i>Default value:</i> false |
| log-file | The name of the log file, which is created in the <code>http-server-root-directory</code> . <i>Allowable values:</i> text <i>Default value:</i> "g2-http-server-log.txt" |

| Attribute | Description |
|--|--|
| http-server-port | The HTTP server port. |
| | <i>Allowable values:</i> integer |
| | <i>Default value:</i> 80 |
| http-server-ssl-enabled | Whether to enable SSL authentication. |
| | <i>Allowable values:</i> truth-value |
| | <i>Default value:</i> false |
| http-server-ssl-certificate-file | The name of the SSL certificate file. |
| | <i>Allowable values:</i> text |
| | <i>Default value:</i> "" |
| http-server-root-directory | The name of the HTTP server root directory. |
| | <i>Allowable values:</i> text |
| | <i>Default value:</i> "C:\temp" |
| http-server-status | (Read-only) The status of the HTTP server. |
| | <i>Allowable values:</i> One of the following symbols: gweb-http-server-connection-error gweb-http-server-disconnected gweb-http-server-initialized gweb-http-server-awaiting-gweb-http-initialization gweb-http-server-awaiting-gsi-connection |
| | <i>Default value:</i> gweb-http-server-disconnected |
| http-server-started-and-initialized | (Read-only) Whether the HTTP server has been started and initialized. |
| | <i>Allowable values:</i> truth-value |

| Attribute | Description |
|------------------------|--|
| | <i>Default value:</i> <code>false</code> |
| http-server-url | The default URL to which the HTTP server should connect. |
| | <i>Allowable values:</i> <code>text</code> |
| | <i>Default value:</i> <code>""</code> |

APIs

`gweb-http-initialize`

(*io:* class `gweb-g2-http-server`)

Initializes the G2 HTTP server, using the attributes specified in the server object.

`gweb-http-start-and-configure-server`

(*io:* class `gweb-g2-http-server`)

Starts and initializes the G2 HTTP server, using the attributes specified in the server object.

`gweb-http-stop-server`

(*io:* class `gweb-g2-http-server`)

Stops the G2 HTTP server.

`gweb-http-update-interface-state`

(*io:* class `gweb-g2-http-server`)

-> *status:* symbol

Updates the interface state of the G2 HTTP server and returns the status information, which is one of the following symbols: `connected`, `not-connected`, `in-transition`, `timed-out`, or `connection-lost`.

`gweb-http-tender-resources`

(*io:* class `gweb-g2-http-server`, *index-number-of-this-tender-offer:* integer, *pattern-to-match-against-requests,* *target:* class `item`)

Registers a Web request URL resource with the G2 HTTP server.

gweb-log-activity

(*io*: class gweb-g2-http-server, *log-type*: symbol, *request-id*: integer, *log-details*)

Logs activity to the log file specified in the G2 HTTP server if logging-enabled is true.

gweb-http-get-attribute-of-request

(*io*: class gweb-g2-http-server, *request*: structure, *attribute-name*)
-> attribute

Returns the value of the specified attribute of the *request* structure of the Web request, where *request* has this syntax:

structure
(*method*: symbol,
headers: structure,
entity: structure | text)

where:

- **method** – The HTTP method of the request, such as **get** or **post**.
- **headers** – A structure of HTTP header attributes included in the request.
- **entity** – If provided as a **structure**, the attributes of the structure are encoded using the application/x-www-form-urlencoded MIME type. If provided as a **text**, the body of the request, which is assumed to already be encoded into 8-bit characters. Note that providing the **entity** as a text is only valid if the method is **post**.

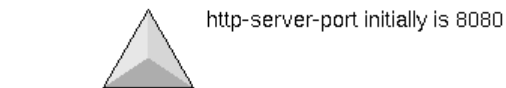
gweb-http-decline-to-serve-url

(*io*: class gweb-g2-http-server, *request*: structure, *http-status-code*: integer, *http-status-description*, *apology*)

Declines to server the URL request and returns an HTTP error specified by *http-status-code*. The *http-status-description* should contain the description of the HTTP status code, and *apology* can provide additional information about the error.

Example

This examples a `gweb-g2-http-server` class definition that specifies the default `http-server-port` as `8080` and an instance of that class that is connected. To connect the G2 HTTP server manually, choose `start http server` on the instance. To disconnect, choose `stop http server`.



http-server-port initially is 8080

MY-HTTP-SERVER-CLASS



MY-G2-HTTP-SERVER

| MY-G2-HTTP-SERVER, a my-http-server-class | |
|---|--|
| Notes | MY-HTTP-SERVER-CLASS-XXX-7: OK |
| Names | MY-G2-HTTP-SERVER |
| Default page | default |
| Logging enabled | false |
| Add request reply details to log | false |
| Add http request attributes to log | false |
| Log file | \$(APPLICATION-ROOT-DIRECTORY)/logs/g2-http-server-log.txt |
| GWEB-HTTP-SERVER::Http server port | 8080 |
| Http server root directory | C:\Program Files\Gensym\g2-8.3r0\g2\data\ |
| Http server status | gweb-http-server-initialized |
| Http server started and initialized | true |
| Http server url | http://nsalvo-1165:8080 |
| Http server ssl enabled | false |
| Http server ssl certificate file | |
| Http server port | 8080 |

G2 WebLink HTTP Server

The `gweb-weblink-http-server` class defines and configures an HTTP server that is based on G2 WebLink. This server configuration requires a connection to a G2 WebLink process running on the network. It does not support SOAP or SSL.

In some circumstances, it might be advantageous to run the HTTP server outside of a firewall, while the G2 process communicating with and serving the pages runs inside the firewall. In this configuration, only the G2 WebLink process would communicate with G2; internet traffic would not reach G2. G2 WebLink and G2 communicate using the Gensym GSI protocol.

Class

Class Inheritance Path

`gweb-weblink-http-server`, `gsi-interface`, `object`

Attributes

| Attribute | Description |
|---|--|
| default-page | The default text for the Web page. <i>Allowable values:</i> text, formatted as free text <i>Default value:</i> "default" |
| logging-enabled | Whether logging is enabled. <i>Allowable values:</i> truth-value <i>Default value:</i> false |
| add-http-request-attributes-to-log | Whether to add HTTP request attribute to the log file. <i>Allowable values:</i> truth-value <i>Default value:</i> false |
| log-file | The name of the log file, which is created in the <code>http-server-root-directory</code> . |

| Attribute | Description |
|--|---|
| <i>Allowable values:</i> | text |
| <i>Default value:</i> | "g2-http-server-log.txt" |
| weblink-configuration | The G2 WebLink configuration object. |
| <i>Allowable values:</i> | an instance of a gw-bridge-configuration |
| <i>Default value:</i> | an instance of a gw-bridge-configuration |
| http-server-port | The HTTP server port. |
| <i>Allowable values:</i> | integer |
| <i>Default value:</i> | 80 |
| http-server-root-directory | The name of the HTTP server root directory. |
| <i>Allowable values:</i> | text |
| <i>Default value:</i> | "C:\temp" |
| http-server-status | (Read-only) The status of the HTTP server. |
| <i>Allowable values:</i> | One of the following symbols: gweb-http-server-connection-error gweb-http-server-disconnected gweb-http-server-initialized gweb-http-server-awaiting-gweb-http-initialization gweb-http-server-awaiting-gsi-connection |
| <i>Default value:</i> | gweb-http-server-disconnected |
| http-server-started-and-initialized | (Read-only) Whether the HTTP server has been started and initialized. |
| <i>Allowable values:</i> | truth-value |

| Attribute | Description |
|--------------------------|--|
| <i>Default value:</i> | false |
| http-server-url | The default URL to which the HTTP server should connect. |
| <i>Allowable values:</i> | text |
| <i>Default value:</i> | "" |

APIs

gweb-http-initialize

(*io*: class gweb-weblink-http-server)

Initializes the G2 WebLink HTTP server, using the attributes specified in the server object.

gweb-http-start-and-configure-server

(*io*: class gweb-weblink-http-server)

Starts and initializes the G2 WebLink HTTP server, using the attributes specified in the server object.

gweb-http-stop-server

(*io*: class gweb-weblink-http-server)

Stops the G2 WebLink HTTP server.

gweb-http-ping

(*io*: class gweb-weblink-http-server)

Pings the G2 WebLink HTTP server.

gweb-http-tender-resources

(*io*: class gweb-weblink-http-server, *index-number-of-this-tender-offer*: integer, *pattern-to-match-against-requests*, *cgi-pattern*)

Registers a web request URL resource with the G2 WebLink server. For details, see the *G2 WebLink User? Guide*.

gweb-log-activity

(*io*: class gweb-weblink-http-server, *log-type*: symbol, *request-id*: integer, *log-details*)

Logs activity to the log file specified in the G2 WebLink HTTP server if *logging-enabled* is true.

gweb-http-get-attribute-of-request

(io: class gweb-weblink-http-server, request-id: integer, attribute-name)

-> attribute

Returns the value of the specified attribute of the *request* structure of the Web request, where *request* has this syntax:

structure

(method: *symbol*,

headers: *structure*,

entity: *structure* | *text*)

where:

- **method** – The HTTP method of the request, such as **get** or **post**.
- **headers** – A structure of HTTP header attributes included in the request.
- **entity** – If provided as a **structure**, the attributes of the structure are encoded using the application/x-www-form-urlencoded MIME type. If provided as a **text**, the body of the request, which is assumed to already be encoded into 8-bit characters. Note that providing the **entity** as a text is only valid if the method is **post**.

gweb-http-decline-to-serve-url

(io: class gweb-weblink-http-server, request-id: integer,

http-status-code: integer, http-status-description, apology)

Declines to server the URL request and returns an HTTP error specified by *http-status-code*. The *http-status-description* should contain the description of the HTTP status code, and *apology* can provide additional information about the error.

gweb-http-add-to-url-reply

(io: class gweb-weblink-http-server, request-id: integer,

additional-resource-content)

Adds text to the response. This is a lower-level API that does not typically need to be called.

gweb-http-finish-url-reply

(io: class gweb-weblink-http-server, request-id: integer)

Indicates the end of the response for the URL. This is a lower-level API that does not typically need to be called.

gweb-set-header-field

(io: class gweb-weblink-http-server, request-id: integer, header: text, data: text)

Sets the HTTP header field. This is a lower-level API that does not typically need to be called.

`gweb-set-url-reply-charset`

(*io*: class `gweb-weblink-http-server`, *request-id*: integer, *charset-name*: symbol)

Sets the URL character set. This is a lower-level API that does not typically need to be called.

`gweb-http-make-base64-encoded`

(*io*: class `gweb-weblink-http-server`, *text-to-encode*: text)

-> text

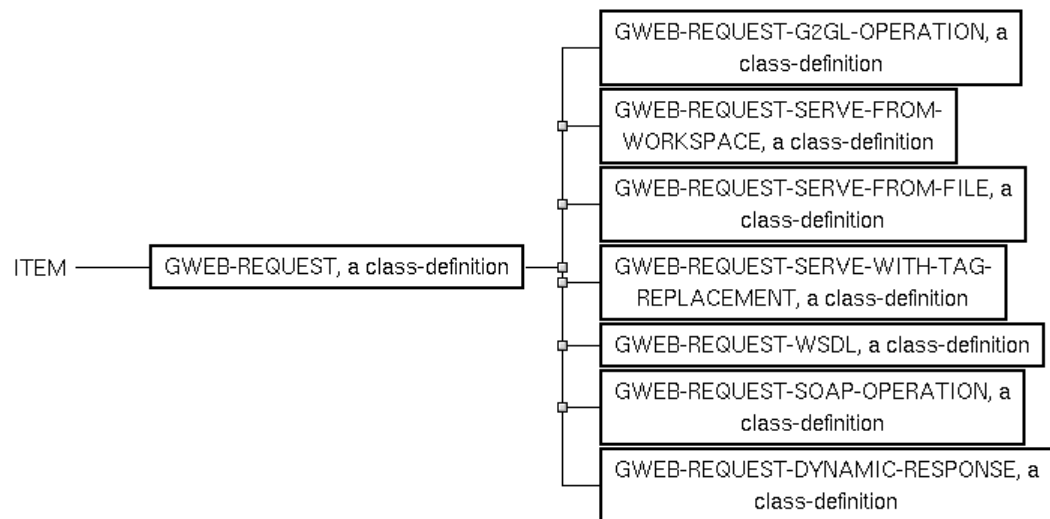
Encodes text into base64.

URL Requests

URL request objects define how HTTP servers respond to URL requests. They are automatically invoked from active HTTP servers when the server receives and dispatches URL requests.

The HTTP server locates the appropriate request object, based on pattern matching of the URL of the request and the URL pattern defined on each URL request object in the `url-pattern` attribute.

GWEB defines the following URL request classes:



Base Response Object

The gweb-request class is the root abstract class of all URL response classes.

Class Inheritance Path

gweb-request, object

Attributes

| Attribute | Description |
|--|---|
| url-pattern | The URL pattern for matching the URL request with the URL request object. <i>Allowable values:</i> text <i>Default value:</i> "" |
| content-type | The content type of the URL request. <i>Allowable values:</i> text <i>Default value:</i> "text/html" |
| online | When true, the request is accessible and can be called via its URL; otherwise, it is offline. <i>Allowable values:</i> truth-value <i>Default value:</i> true |
| restricted-to-user-modes | The URL request is restricted to these user modes. <i>Allowable values:</i> sequence <i>Default value:</i> sequence() |
| restricted-to-valid-user-sessions | Whether the URL request is restricted to valid user sessions. <i>Allowable values:</i> truth-value |

| Attribute | Description |
|--------------------------------------|--|
| | <i>Default value:</i> true |
| resource-id | Used to prioritize resources and typically is automatically assigned upon startup of the HTTP server. |
| | <i>Allowable values:</i> integer |
| | <i>Default value:</i> 1 |
| hit-counter | (Read-only) A counter of the number of requests to this Web page. |
| | <i>Allowable values:</i> integer |
| | <i>Default value:</i> 0 |
| last-access-time | (Read-only) The time of the last URL request. |
| | <i>Allowable values:</i> quantity |
| | <i>Default value:</i> 0.0 |
| wSDL-binding | The binding to set up in WSDL descriptions auto-generated by G2. (See the WSDL 1.1 specification.) |
| | <i>Allowable values:</i> http-get http-post soap-document soap-rpc |
| | <i>Default value:</i> http-get |
| wSDL-input-message-definition | The name of a gweb-wSDL-message-definition instance describing the input message of SOAP operations. (See the WSDL 1.1 specification.) |

| Attribute | Description |
|--|--|
| <p><i>Allowable values:</i> symbol</p> <p><i>Default value:</i> none</p> | |
| <p>wSDL-output-message-definition</p> | <p>The name of a <code>gweb-wSDL-message-definition</code> instance describing the output message of SOAP operations if any is required. (See the WSDL 1.1 specification.)</p> |
| <p><i>Allowable values:</i> symbol</p> <p><i>Default value:</i> none</p> | |
| <p>wSDL-fault-message-definition</p> | <p>The name of a <code>gweb-wSDL-message-definition</code> instance describing the fault message associated with the SOAP <code>wSDL-fault-name</code>, if any. (See the WSDL 1.1 specification.)</p> |
| <p><i>Allowable values:</i> symbol</p> <p><i>Default value:</i> none</p> | |
| <p>wSDL-fault-name</p> | <p>The name of a SOAP operation that might be returned by the called SOAP operation.</p> |
| <p><i>Allowable values:</i> symbol</p> <p><i>Default value:</i> none</p> | |
| <p>wSDL-port-type</p> | <p>The WSDL port type this operation is part of. All <code>gweb-request</code> instances with the <code>wSDL-port-type</code> using the same value are automatically grouped together in the WSDL file within a single WSDL port type section. (See the WSDL 1.1 specification.)</p> |
| <p><i>Allowable values:</i> symbol</p> <p><i>Default value:</i> none</p> | |

| Attribute | Description |
|-----------------------|---|
| wSDL-operation | The WSDL operation name for the <code>gweb-request</code> , which should be unique within a <code>wSDL-port-type</code> group. (See the WSDL 1.1 specification.) <i>Allowable values:</i> <code>symbol</code> <i>Default value:</i> <code>none</code> |
| documentation | Any documentation to be included in the WSDL file to describe the operation. <i>Allowable values:</i> <code>text</code> <i>Default value:</i> <code>""</code> |

APIs

`gweb-http-validate-access`

(*web-request*: class `gweb-request`, *url*, *cgi-arguments*: structure, *session*: class `gweb-user-session`, *request*: item-or-value, *io*: class item)
-> *access-granted*: truth-value, *status-code*: integer

When the URL is received by the HTTP server, the HTTP server calls this method to grant access to this resource or not.

`gweb-http-generate-url-reply`

(*web-request*: class `gweb-request`, *cgi-arguments*: structure, *session*: class `gweb-user-session`, *request*: item-or-value, *io*: class item)
-> *response*: structure

When the URL is received by the HTTP server, and after calling `gweb-http-validate-access` and it returns `true`, the HTTP server calls this method to process the URL request. This method needs to compute the response.

The *response* typically has this format:

```
structure
(status-code: 200,
headers: structure (content-type: content-type),
entity: "")
```

URL Response Object from Files

The `gweb-request-serve-from-file` class is the URL response class serving a response that is based on the contents of a file. The filename and pathname are relative to the root of the HTTP server directory, which is pre-pended dynamically at run time. The file reader uses the text conversion style specified in the `stream-text-conversion-style` attribute.

Class Inheritance Path

`gweb-request-serve-from-file`, `gweb-request`, `object`

Attributes

| Attribute | Description |
|-------------------------------------|--|
| source-filename | The name of the source file that is the request. |
| <i>Allowable values:</i> | <code>text</code> |
| <i>Default value:</i> | <code>""</code> |
| stream-text-conversion-style | The name of a G2 <code>text-conversion-style</code> object used to convert the text file. See the <i>G2 Reference Manual</i> . |
| <i>Allowable values:</i> | <code>text</code> |
| <i>Default value:</i> | <code>""</code> |

APIs

`gweb-http-generate-url-reply`

(*web-request*: class `gweb-page-from-file`, *cgi-arguments*: structure, *session*: class `gweb-user-session`, *request*: item-or-value, *io*: class item)
-> *response*: structure

When the URL is received by the HTTP server, and after calling `gweb-http-validate-access` and it returns true, the HTTP server calls this method to process the URL request. This method needs to compute the response.

The *response* typically has this format:

```
structure
(status-code: 200,
headers: structure (content-type: content-type),
entity: "")
```

URL Response Object with Tag Replacement

The `gweb-request-serve-with-tag-replacement` class is the URL response class serving a response that is based on a predefined document in which tags are replaced with values. The filename and pathname are relative to the root of the HTTP server directory, which is pre-pended dynamically at run time. The file reader uses the US-ASCII text conversion style.

Class Inheritance Path

`gweb-request-serve-with-tag-replacement`, `gweb-request`, `object`

Attributes

| Attribute | Description |
|--------------------------|--|
| entity | The entity of the request, if any. |
| <i>Allowable values:</i> | text |
| <i>Default value:</i> | "" |
| source-filename | The name of the source file that is the request. |
| <i>Allowable values:</i> | text |
| <i>Default value:</i> | "" |

APIs

`gweb-http-generate-url-reply`

```
(web-request: class gweb-page-with-tags, cgi-arguments: structure,
session: class gweb-user-session, request: item-or-value, io: class item)
-> response: structure
```

When the URL is received by the HTTP server, and after calling `gweb-http-validate-access` and it returns `true`, the HTTP server calls this method to process the URL request. This method needs to compute the response.

The *response* typically has this format:

```
structure
(status-code: 200,
headers: structure (content-type: content-type),
entity: "")
```

URL Response Object from Workspace

The `gweb-request-serve-from-workspace` class is the URL response object that embeds a G2 workspace within the entity an HTML page.

Class Inheritance Path

`gweb-request-serve-from-workspace`, `gweb-request`, `object`

Attributes

| Attribute | Description |
|-------------------------------|---|
| entity | The entity of the request, if any. |
| <i>Allowable values:</i> | text |
| <i>Default value:</i> | "" |
| source-filename | The name of the source file that is the request. |
| <i>Allowable values:</i> | text |
| <i>Default value:</i> | "" |
| workspace-name-or-uuid | The name or UUID of the kb-workspace to show in the Web page. |
| <i>Allowable values:</i> | value |
| <i>Default value:</i> | "" |

APIs

`gweb-http-generate-url-reply`

(*web-request*: class `gweb-page-from-workspace`, *cgi-arguments*: structure, *session*: class `gweb-user-session`, *request*: item-or-value, *io*: class `item`)
-> *response*: structure

When the URL is received by the HTTP server, and after calling `gweb-http-validate-access` and it returns `true`, the HTTP server calls this method to process the URL request. This method needs to compute the response.

The *response* typically has this format:

```
structure
(status-code: 200,
headers: structure (content-type: content-type),
entity: "")
```

Dynamic URL Response Object

The `gweb-request-dynamic-response` class is the URL response object that dynamically computes the response for a URL request. The logic is implemented as a procedure with this signature:

```
my-page
(session: class gweb-user-session, cgi-arguments: structure,
request: item-or-value, io: class item)
-> response: structure
```

When the URL is received by the HTTP server, and after calling `gweb-http-validate-access` and it returns `true`, the HTTP server calls this procedure to process the URL request. This procedure needs to compute the response.

The *response* typically has this format:

```
structure
(status-code: 200,
headers: structure (content-type: content-type),
entity: "")
```

Class Inheritance Path

`gweb-request-dynamic-response`, `gweb-request`, `object`

APIs

```
gweb-http-generate-url-reply
(web-request: class gweb-request-dynamic-response,
cgi-arguments: structure, session: class gweb-user-session,
request: item-or-value, io: class item)
-> response: structure
```

When the URL is received by the HTTP server, and after calling `gweb-http-validate-access` and it returns `true`, the HTTP server calls this method to process the URL request. This method needs to compute the response.

The *response* typically has this format:

```
structure
(status-code: 200,
headers: structure (content-type: content-type),
entity: "")
```

SOAP Operation

The `gweb-request-soap-operation` class is the URL response object for SOAP requests that dynamically compute the SOAP response. The logic is implemented as a procedure with this signature:

```
my-soap-handler
(http-server: class item, soap-request: structure,
 session: class gweb-user-session)
 response: structure
```

For a description of the *response* structure, see the `g2-handle-http-request-as-soap` system procedure.

Class Inheritance Path

`gweb-request-soap-operation`, `gweb-request`, `object`

APIs

```
gweb-http-generate-url-reply
(soap-handler: class gweb-dynamic-soap-page,
 cgi-arguments: structure, session: class gweb-user-session,
 request: item-or-value, io: class item)
-> response: structure
```

When the URL is received by the HTTP server, and after calling `gweb-http-validate-access` and it returns true, the HTTP server calls this method to process the URL request. This method needs to compute the response.

The *response* typically has this format:

```
structure
(status-code: 200,
headers: structure (content-type: content-type),
entity: "")
```

WSDL

The `gweb-request-wsdl` class is the URL response object for generating and returning WSDL for all defined `gweb-request` objects in G2 that are configured with WSDL information.

Class Inheritance Path

`gweb-request-wsdl`, `gweb-request`, `object`

Attributes

| Attribute | Description |
|--------------------------|------------------------|
| namespace | The WSDL namespace. |
| <i>Allowable values:</i> | text |
| <i>Default value:</i> | "http://g2tempuri.org" |

APIs

`gweb-http-generate-url-reply`

(*wsdl-handler*: class `gweb-request-wsdl`, *cgi-arguments*: structure, *session*: class `gweb-user-session`, *request*: item-or-value, *io*: class item)
-> *reply*: structure

Called when a URL is received that matches an instance of `gweb-request-wsdl`. It generates a WSDL and returns it as a reply to the HTTP/URL request.

`gweb-generate-wsdl`

(*wsdl-handler*: class `gweb-request-wsdl`, *io*: class `gweb-http-server`)
-> *wsdl*: structure

Generates a WSDL as XML value elements structure and returns it as an XML value element.

`gweb-save-wsdl`

(*wsdl-handler*: class `gweb-request-wsdl`, *io*: class `gweb-http-server`, *filename*: text)

Generates a WSDL and saves it in XML text format to the specified filename.

G2GL Operation

The `gweb-request-g2gl-operation` class is the URL response object for HTTP or SOAP requests that invoke G2GL/BPMS processes.

The API method `gweb-http-generate-url-reply` is implemented to automatically convert input and output data and to invoke the G2GL process.

For an example, see `bpms-demo.kb` in `g2i\examples`.

Class Inheritance Path

`gweb-request-g2gl-operation`, `gweb-request`, `object`

Attributes

| Attribute | Description |
|----------------------------|---|
| service-switch-name | The service switch to which the G2GL process is assigned. |
| <i>Allowable values:</i> | symbol |
| <i>Default value:</i> | g2gl-standard-service-switch |

For HTTP requests and replies, the message communication with the G2GL process, which includes input and output messages and message variable types, is of type `gweb-g2gl-http-message`, which has these attributes:

| Attribute | Description |
|--------------------------|--|
| request | The HTTP request. |
| <i>Allowable values:</i> | item-or-value |
| <i>Default value:</i> | "" |
| request-xml | The entity of the request as XML value elements. |
| <i>Allowable values:</i> | structure |
| <i>Default value:</i> | structure() |

| Attribute | Description |
|---------------------------|--|
| request-query-info | The query info of the request, typically the cgi arguments in an HTTP Get request. |
| <i>Allowable values:</i> | structure |
| <i>Default value:</i> | structure() |
| reply-status-code | The HTTP reply status that can be set by the G2GL process. |
| <i>Allowable values:</i> | integer |
| <i>Default value:</i> | 200 |
| reply-headers | The HTTP reply header that can be set by the G2GL process. |
| <i>Allowable values:</i> | structure |
| <i>Default value:</i> | structure(content-type: "application/xml") |
| reply-entity | The HTTP reply entity that can be set by the G2GL process. If the reply-entity is a structure, its assumed to contain XML value elements that will be converted to the XML text representation. If the reply-entity contains text, it is included as is in the HTTP reply; otherwise, the content is converted to text and is included in the reply. |
| <i>Allowable values:</i> | value |
| <i>Default value:</i> | "" |

For SOAP requests and replies, the message communication with the G2GL process is of type `gweb-g2gl-soap-message`, which has these attributes:

| Attribute | Description |
|--------------------------|-------------------|
| soap-request | The SOAP request. |
| <i>Allowable values:</i> | structure |
| <i>Default value:</i> | structure() |
| soap-reply | The SOAP reply. |
| <i>Allowable values:</i> | structure |
| <i>Default value:</i> | structure() |

Custom Response Object

To implement a custom response object, create a subclass of `gweb-request` and implement the `gweb-http-generate-url-reply`. Optionally, you can implement the `gweb-http-validate-access` method to validate a URL request and grant access.

Class Inheritance Path

my-weblet-class, `gweb-request`, object

APIs

You must implement the following APIs for your custom response class:

`gweb-http-generate-url-reply`

(*soap-handler*: class *my-weblet-class*,
cgi-arguments: structure, *session*: class `gweb-user-session`,
request: item-or-value, *io*: class item)
-> *response*: structure

Called upon matching the request. This method must implement the logic of the request and return the HTTP server response structure. The *response* typically has this format:

```
structure
(status-code: 200,
headers: structure (content-type: content-type),
entity: "")
```

`gweb-http-validate-access`

(*web-request*: class `gweb-request`, *url*, *cgi-arguments*: structure,
session: class `gweb-user-session`, *request*: item-or-value, *io*: class item)
-> *grant-access*: truth-value, *status-code*: integer

Called prior to calling `gweb-http-generate-url-reply`. If access is not granted, an error response is automatically returned.

User Sessions

The `gweb-user-session` class tracks user logins and are used to grant access to URL requests.

Class

Class Inheritance Path

`gweb-user-session`, `object`

Attributes

| Attribute | Description |
|--------------------------|--|
| session-id | The session ID. |
| <i>Allowable values:</i> | integer |
| <i>Default value:</i> | 0 |
| user-name | The user name for logging in to the HTTP server. |
| <i>Allowable values:</i> | symbol |
| <i>Default value:</i> | none |
| user-mode | The user mode for logging into the HTTP server. |
| <i>Allowable values:</i> | symbol |
| <i>Default value:</i> | none |
| user-password | The user password for logging in to the HTTP server. |
| <i>Allowable values:</i> | text |
| <i>Default value:</i> | "" |

| Attribute | Description |
|---------------------------|---|
| user-logged-in | (Read-only) Whether the user is currently logged in to the HTTP server. <i>Allowable values:</i> truth-value <i>Default value:</i> false |
| remote-client-host | (Read-only) The host of the remote client from which the user is logging into the HTTP server. <i>Allowable values:</i> text <i>Default value:</i> "" |
| creation-time | (Read-only) The creation time of the user session. <i>Allowable values:</i> quantity <i>Default value:</i> 0 |
| update-time | (Read-only) The last update time of the user session. <i>Allowable values:</i> quantity <i>Default value:</i> 0 |

APIs

gweb-user-login

(*user-name*: symbol, *password*, *http-client-host-name*,
request: item-or-value)

-> *success*: truth-value, *session*: class gweb-user-session

Logs into an HTTP server with the given *user-name*, *password*, *http-client-host-name*, and *request*. Returns **true** if the login was successful and returns the **gweb-user-session**.

In a non secure G2, the user name and password are validated with the corresponding attributes in `gweb-user-sessions` instances.

In a non secure G2, the user name and password are validated using the `g2-validate-user-and-password`, which validates them based on specifications defined in the G2 OK file for a secure G2. A `gweb-user-sessions` instance with the same user name is returned if the login is successful.

`gweb-get-user-session`

(*sid*, *request*: item-or-value)

-> success: truth-value, session: class `gweb-user-session`

Returns the user session with the given *sid*, which is the session ID. If the session is not found because the user does not have access rights, it returns the default guest session.

`gweb-user-logout`

(*session*: class `gweb-user-session`, *request*: item-or-value)

Logs out of the specified *session*, using the specified *request*.

`gweb-get-user-session-header`

(*session*: class `gweb-user-session`)

-> header

Returns the header of the specified *session*.

`gweb-get-user-session-footer`

(*session*: class `gweb-user-session`)

-> footer

Returns the footer of the specified *session*.

HTML Support

Describes the APIs used to support HTML.

| | |
|---------------------------------|----|
| Introduction | 51 |
| HTML Pages | 52 |
| Layout and Formatting | 53 |
| Hyperlinks | 54 |
| Images | 54 |
| Text Formatting | 55 |
| Lists | 56 |
| Tables | 56 |
| Frames | 58 |
| Forms | 59 |
| Creating HTML Content | 61 |
| Extracting Form Response Values | 61 |



Introduction

GWEB provides numerous functions and procedures for interacting with HTML Web pages. The supported HTML is Version 4.0.

HTML Pages

Use the following functions to create HTML page tags.

gweb-html-begin

()

Marks the beginning of the HTML page by inserting the following as the header:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Frameset//
EN@" @http://www.w3.org/TR/REC-html40/frameset.dtd">
```

gweb-html-end

()

Marks the end of the HTML page.

gweb-page-header-begin

()

Marks the beginning of the Web page header.

gweb-page-header-end

()

Marks the end of the Web page header.

gweb-title

(title)

Inserts a title into the Web page. The web page title must appear between the header begin and header end tags.

gweb-meta

(meta-tag)

Inserts a meta tag into the Web page to specify the page creator and refresh rate.

gweb-page-body-begin

(background-url, background-color, text-color, link-color, visited-link-color, selected-link-color)

Marks the beginning of the body of the web page, specifying various colors for the Web page.

gweb-page-body-end

()

Marks the end of the body of the HTML page.

gweb-body-and-html-end
()

Marks the end of the body and HTML page.

Layout and Formatting

Use the following functions to insert layout and formatting tags, where *id* is the ID of the HTML tag, *class* is the HTML class, if any, and *style* is "absolute", "relative", or "static".

gweb-div
(*txt, id, class, style*)

Creates a logical division containing *txt*.

gweb-div-begin
(*id, class, style*)

Marks the beginning of a logical division in an HTML page. The *style* can be relative, absolute, or static.

gweb-div-end
()

Marks the end of a logical division in an HTML page.

gweb-span
(*txt, id, class, style*)

Creates a logical division containing *txt*.

gweb-span-begin
(*id, class, style*)

Marks the beginning of a logical division in an HTML page.

gweb-span-end
()

Marks the end of a logical division in an HTML page.

gweb-horizontal-rule
(*id, class, style*)

Inserts a straight line across the HTML page to provide separations in the page.

gweb-break
()

Inserts a linefeed into the HTML page.

gweb-tab

()

Inserts a tab character in the HTML page.

gweb-form-feed

()

Inserts a form feed into the HTML page.

gweb-space

()

Inserts a space character into the HTML page.

gweb-zero-width-space

()

Inserts a zero-width-space character into the HTML page.

Hyperlinks

Use the following function to insert a hyperlink.

gweb-hyperlink

(href, target, txt, image)

Inserts a hyperlink into the HTML page, where *href* is a hyperlink reference in the form of a valid URL such as "http://www.gensym.com", *target* is the target frame, *txt* is the text to present for the hyperlink, and *image* is an HTML reference to an image as created by **gweb-image**.

Images

Use the following functions to insert layout and formatting tags, where *id* is the ID of the HTML tag, *class* is the HTML class, if any, and *style* is "absolute", "relative", or "static".

gweb-image

(id, class, style, image, alternate-text, height, width)

Generates the text for an image to include in an HTML page, where *image* specifies the file location of the image. The *height* and *width* are the optimal height and width of the image. However, if the height and width measurements are smaller than that of the image, the image is reduced to fit the dimensions. The *alternate-text* appears when hovering the mouse over the image.

Text Formatting

Use the following functions to insert text formatting tags, where *id* is the ID of the HTML tag, *class* is the HTML class, if any, and *style* is "absolute", "relative", or "static".

gweb-align-center

(id, class, style, txt)

Aligns *txt* to the center of the HTML page.

gweb-bold

(id, class, style, txt)

Formats *txt* in bold.

gweb-italic

(id, class, style, txt)

Formats *txt* in italics.

gweb-big

(id, class, style, txt)

Formats *txt* to appear big.

gweb-small

(id, class, style, txt)

Formats *txt* to appear small.

gweb-quote

(id, class, style, txt)

Formats *txt* to appear as a block quotation.

gweb-paragraph

(id, class, style, txt)

Formats *txt* to wrap the text in a paragraph.

gweb-font

(id, class, style, size, txt)

Formats the size of *txt*, where *size* is an integer from 1 to 7, and 1 is tiny, 4 is average, and 7 is huge.

gweb-heading1

(id, class, style, txt)

Formats *txt* to be a heading 1.

gweb-heading2

(id, class, style, txt)

Formats *txt* to be a heading 2.

gweb-heading3

(id, class, style, txt)

Formats *txt* to be a heading 3.

gweb-heading4

(id, class, style, txt)

Formats *txt* to be a heading 4.

gweb-heading5

(id, class, style, txt)

Formats *txt* to be a heading 5.

gweb-heading6

(id, class, style, txt)

Formats *txt* to be a heading 6.

Lists

Use the following functions to insert list tags, where *id* is the ID of the HTML tag, *class* is the HTML class, if any, and *style* is "absolute", "relative", or "static".

gweb-ordered-list

(id, class, style, start-number, txt)

Adds *txt* to an ordered list, where *start-number* is the start number for the list entry.

gweb-unordered-list

(id, class, style, txt)

Adds *txt* to an unordered list.

gweb-list-entry

(id, class, style, n, txt)

Adds *txt* to a list, where *n* is the number of the list entry.

Tables

Use the following functions to insert table tags, where *id* is the ID of the HTML tag, *class* is the HTML class, if any, and *style* is "absolute", "relative", or "static".

gweb-table-top

(id, class, style, summary, caption-text, border-size, table-width)

Marks the start of a table and initializes key table attributes.

gweb-table-bottom*(caption-text)*

Marks the end of a table and optionally places a caption at the bottom of the table.

gweb-table-header*(id, class, style, header-text)*

Adds a table header.

gweb-table-footer*(id, class, style, footer-text)*

Adds a table footer.

gweb-table-body*(id, class, style, body-text)*

Adds a table body.

gweb-table-row*(id, class, style, row-text, horizontal-alignment, vertical-alignment, background-color, border-color)*

Wraps *row-text* with table row begin and table row end syntax, where *horizontal-alignment* as **left**, **right** or **center**, and *vertical-alignment* as **top**, **center**, **bottom** or **baseline**.

gweb-table-row-begin*(id, class, style, horizontal-alignment, vertical-alignment, background-color, border-color)*

Marks the beginning of a table row.

gweb-table-row-end*()*

Marks the end of a table row.

gweb-table-header-cell*(id, class, style, cell-text, width, horizontal-alignment, vertical-alignment, column-span, row-span, background-color, border-color)*

Creates a table header cell, where *horizontal-alignment* as **left**, **right** or **center**, and *vertical-alignment* as **top**, **center**, **bottom** or **baseline**.

gweb-table-header-cell-begin*(id, class, style, width, horizontal-alignment, vertical-alignment, column-span, row-span, background-color, border-color)*

Marks the beginning of a table header cell.

gweb-table-header-cell-end

()

Marks the end of a table header cell.

gweb-table-header-cell

(*id, class, style, cell-text, width, horizontal-alignment, vertical-alignment, column-span, row-span, background-color, border-color*)

Creates a table data cell, where *horizontal-alignment* as **left**, **right** or **center**, and *vertical-alignment* as **top**, **center**, **bottom** or **baseline**.

gweb-table-data-cell-begin

(*id, class, style, width, horizontal-alignment, vertical-alignment, column-span, row-span, background-color, border-color*)

Marks the beginning of a table data cell. Use **gweb-table-data-cell-end** to mark the end of the table data cell.

gweb-table-data-cell-end

()

Marks the end of a table data cell.

Frames

Use the following functions to insert frame tags, where *id* is the ID of the HTML tag, *class* is the HTML class, if any, and *style* is "absolute", "relative", or "static".

gweb-frameset

(*id, class, style, column-specification, row-specification, frame-text*)

Inserts a frameset and its contents.

gweb-frame

(*id, class, style, frame-name, margin-width, margin-height, no-resize, scrolling, src, frame-text*)

Inserts a frame and its contents. If *no-resize* is a non-empty string, resizing is disabled. The *scrolling* mode is one of "yes", "no", "auto", or an empty string. The *scr* is the URL source.

gweb-noframe

(*id, class, style, txt,*)

Inserts text to be displayed if frames are not supported.

gweb-iframe

(id, class, style, frame-name, width, height, margin-width, margin-height, no-resize, align, scrolling, src, frame-text)

Inserts a floating frame and its contents. If *no-resize* is a non-empty string, resizing is disabled. The options for *align* are "left", "right", or "center". The *scrolling* mode is one of **yes**, **no**, **auto**, or an empty string. The *scr* is the URL source.

Forms

Use the following functions to insert form tags, where *id* is the ID of the HTML tag, *class* is the HTML class, if any, and *style* is "absolute", "relative", or "static".

gweb-form-begin

(id, class, style, action, method, form-name, encryption-type)

Marks the beginning of an HTML form. The *action* is mapped to a procedure in G2. The *method* it is always "get". The *form-name* can be any name you choose but must be unique to the page. The *encryption-type* is currently not used.

gweb-form-end

()

Marks the end of an HTML form.

Use the following G2 functions to add input elements to forms:

gweb-form-checkbox

(id, class, style, control-name, input-value, checked, label, text-color)

Adds a check box to a form, where *input-value* is the value of the check box, *checked* is "true" or "false", *label* is the text label, and *text-color* is the label color.

gweb-form-file

(id, class, style, control-name, max-length, size)

Adds a file selection to a form.

gweb-form-hidden

(id, class, style, control-name, input-value, max-length, size)

Adds a hidden element to a form.

gweb-form-image

(id, class, style, control-name, src)

Adds an image input to a form, where *scr* is a URL to the source of the image.

gweb-form-password

(id, class, style, control-name, input-value, max-length, size)

Adds a password input to a form.

gweb-form-radio-button

(id, class, style, control-name, input-value, checked)

Adds a radio-button input to a form.

gweb-form-reset

(id, class, style, control-name, input-value)

Adds a reset button to a form.

gweb-form-submit

(id, class, style, control-name, input-value)

Adds a submit button to a form.

gweb-form-text

(id, class, style, control-name, input-value, max-length, size)

Adds a text box to a form, where *input-value* is the initial text value, *max-length* is the maximum length of the text, and *size* is the size of the text box.

gweb-form-textarea

(id, class, style, rows, columns, control-name, initial-value)

Adds a multi-line text input to a form, where *rows* is the height of the text area, *columns* is the width of the text area, and *initial-value* is the initial text to appear.

gweb-form-select-begin

(id, class, style, control-name, size)

Adds a dropdown box to a form. Follow **gweb-form-select-begin** with one or more calls to **gweb-form-select-option** and a single call to **gweb-form-select-end**.

gweb-form-select-option

(id, class, style, choice-name, choice-value, selected)

Adds a dropdown box option to a dropdown box, where *choice-name* is the name of the selection, *choice-value* is the initial value, and *selected* indicates whether the option is initially selected when "true" or "yes".

gweb-form-select-end

()

Marks the end of the dropdown box.

Creating HTML Content

Use the following procedures to create HTML content:

gweb-html-selection-from-sequence

(*control-name*: text, *selected-value*: value, *choices*: sequence, *size*: value)
-> selection

Creates a combo box populating it with *choices* and selecting the *selected-value* if included in list of choices. If *size* is a non-empty string, it should specify the size of the combo box as a *text* or a *quantity*.

gweb-html-table-from-sequence-of-sequences

(*background-color*: text, *border-color*: text, *border-size*: text, *table-width*: text, *align*: text, *caption-text*: text, *columns*: sequence, *table-values*: sequence)
-> text

Creates an HTML table for a G2 value array specified as sequences of sequences.

gweb-html-table-from-sequence-of-structures

(*background-color*: text, *border-color*: text, *border-size*: text, *table-width*: text, *align*: text, *caption-text*: text, *columns*: sequence, *table-values*: sequence)
-> text

Creates an HTML table for a G2 value array specified as sequences of structures.

gweb-html-table-from-structure-of-sequences

(*background-color*: text, *border-color*: text, *border-size*: text, *table-width*: text, *align*: text, *caption-text*: text, *columns*: sequence, *table-values*: structure)
-> text

Creates a HTML table for a G2 value array specified as structure of sequences.

Extracting Form Response Values

Use the following procedures to extract response values from forms, using a gweb-weblink-https-server:

gweb-get-cgi-value

(*io*: class gweb-weblink-http-server, *fetch-id*: integer, *control-name*: text, *default-value*: value)
-> changed: truth-value, control-value: value

Returns true and the value of a control if the input was changed; otherwise, returns false and the default value.

gweb-get-cgi-value-as-truth-value

(*io: class gweb-weblink-http-server, fetch-id: integer, control-name: text, default-value: truth-value*)

-> changed: truth-value, control-value: truth-value

Returns true and the value of a control if the input was changed; otherwise, returns false and the default value.

gweb-get-cgi-value-as-integer

(*io: class gweb-weblink-http-server, fetch-id: integer, control-name: text, default-value*)

-> changed: truth-value, control-value

Returns true and the value of a control if the input was changed; otherwise, returns false and the default value.

gweb-get-cgi-value-as-quantity

(*io: class gweb-weblink-http-server, fetch-id: integer, control-name: text, default-value: quantity*)

-> changed: truth-value, control-value: quantity

Returns true and the value of a control if the input was changed; otherwise, returns false and the default value.

gweb-get-cgi-value-as-symbol

(*io: class gweb-weblink-http-server, fetch-id: integer, control-name: text, default-value: symbol*)

-> changed: truth-value, control-value: symbol

Returns true and the value of a control if the input was changed; otherwise, returns false and the default value.

@ A B C D E F G H I J K L M
N O P Q R S T U V W X Y Z

C

customer support services

G

G2 Web (GWEB)
 built-in Web support
 HTML support
 introduction to
 loading
 module settings
 Web servers
G2GL URL requests
gweb.kb
gweb-g2-http-server
gweb-module-settings
gweb-request
gweb-request-dynamic-response
gweb-request-serve-from-file
gweb-request-serve-from-workspace
gweb-request-serve-with-tag-replacement
gweb-request-soap-operation
gweb-user-session
gweb-weblink-http-server

H

HTML
 creating content
 extracting form response values
 forms
 frames
 hyperlinks
 images
 layout and formatting
 lists
 page tags
 tables
 text formatting

S

SOAP
 basic SOAP demo
 operations
 operator message demo
 URL requests

U

URL requests
 base response object
 G2GL operations
 introduction to
 SOAP operations
 URL response object
 custom
 dynamically computed
 from files
 from workspace
 with tag replacement
WSDL

W

Web
 built-in support for Web pages
 G2 Web (GWEB)
 URL requests
 Web servers
Web pages, built-in
Web servers
 G2 WebLink
 G2-built
workspaces
 displaying in Web pages
WSDL
 demo
 requests

