

# G2 WebLink

## User's Guide

Version 2020



G2 WebLink User's Guide, Version 2020

May 2020

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 © 1985-2020 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 © 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.

Ignite Technologies, Inc.  
401 Congress Ave., Suite 2650  
Austin, TX 78701 USA  
Telephone: +1-800-248-0027  
Email: [success@ignitetech.com](mailto:success@ignitetech.com)

Part Number: DOC035-1200

# Contents

---

	<b>Preface</b>	<b>vii</b>
	About this Guide	vii
	Audience	vii
	Organization	viii
	A Note About the API	viii
	Conventions	ix
	Related Documentation	x
	Customer Support Services	xiii
<b>Chapter 1</b>	<b>Introduction</b>	<b>1</b>
	Introduction	1
	Communicating with G2 WebLink	2
	Configuring the Behavior of G2 WebLink	2
	G2 WebLink Processes	3
	Using G2 WebLink as an HTTP Server	3
	Identifying Resources	3
	Serving Resources	4
	Data Types Used by G2 WebLink	4
	Defining the Default Character Set	5
	Transcripts of G2 WebLink Operation	5
<b>Chapter 2</b>	<b>Configuring G2 WebLink</b>	<b>7</b>
	Introduction	7
	Running G2 WebLink	7
	Configuring G2 WebLink	9
	Configuring the Connection between G2 and G2 WebLink	9
	Configuring G2 WebLink Behavior	10
	Checking the Status of the Connection to the Bridge	10
	Optional Steps for Configuring G2 WebLink	11
	Ensuring a Reliable Connection	11

Logging Host Names 11

### **Chapter 3 Remote Procedures 13**

Introduction 13

Summary of Remote Procedures 14

Return Values and Signals of Remote Procedures 15

gw-add-to-url-reply 16

gw-configure-bridge 18

gw-decline-to-serve-url 19

gw-finish-url-reply 21

gw-get-attribute-of-request 23

    MIME Headers 24

    CGI Arguments 24

    Other Argument Names 25

gw-kill-bridge 29

gw-make-base64-encoded 30

gw-ping 31

gw-set-header-field 32

gw-set-url-reply-charset 33

gw-start-http-server 35

gw-tender-resources 37

    Specifying an Index 38

    Specifying a Pattern 38

    Specifying a Response 38

    The call Response Format 38

    The serve Response Format 40

### **Chapter 4 Example KBs 41**

Introduction 41

Example of a Simple Server 42

Example of a Server Providing a Choice of Resources 44

Example of Setting the Header Field and URL Character Set 48

### **Appendix A Using HTML and SGML 49**

Introduction 49

Using Hypertext Markup Language (HTML) 49

Using Standard Generalized Markup Language 50

**Glossary 53**

**Index 55**



# Preface

---

*Describes this guide and the conventions that it uses.*

About this Guide	vii
Audience	vii
Organization	viii
A Note About the API	viii
Conventions	ix
Related Documentation	x
Customer Support Services	xiii



## About this Guide

This guide provides an introduction to the [capabilities](#) of G2 WebLink. It explains how to [configure](#) G2 WebLink, and how to use [remote procedures](#) in your G2 application to communicate with it. It describes [examples](#) of applications that use G2 WebLink. An appendix provides an overview of [HTML and SGML](#), and a [glossary](#) explains technical terms used in this document.

## Audience

This guide is written for G2 Knowledge Base developers, who are addressed throughout as “you.”

# Organization

This guide contains five chapters and one appendix:

<b>Title</b>	<b>Description</b>
<a href="#">1 Introduction</a>	<a href="#">Describes the capabilities of G2 WebLink, and explains basic principles and concepts underlying the operation of G2 WebLink.</a>
<a href="#">2 Configuring G2 WebLink</a>	<a href="#">Describes steps that you must always take to configure G2 WebLink, as well as some optional steps that you can take to customize how G2 WebLink behaves.</a>
<a href="#">3 Remote Procedures</a>	<a href="#">Describes the remote procedures that are provided with G2 WebLink.</a>
<a href="#">4 Example KBs</a>	Describes the example KBs provided with G2 WebLink.
<a href="#">A Using HTML and SGML</a>	<a href="#">Describes the use of Hypertext Markup Language (HTML) and Standard Generalized Markup Language (SGML).</a>

## A Note About the API

The G2 WebLink API, as described in this guide, is not expected to change significantly in future releases, but exceptions may occur. A detailed description of any changes will accompany the G2 WebLink release that includes them.

However, it is essential that you use G2 WebLink exclusively through its API, as described in this guide. If you bypass the API, you cannot rely on your code to work in the future, since G2 WebLink may change, or in the present, because the code may not correctly manage the internal operations of G2 WebLink.

If G2 WebLink does not seem to provide the capabilities that you need, contact Gensym Customer Support at +1-855-453-8174 for further information.



# 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
<b>workspace</b>	Glossary terms

Convention Examples	Description
c:\Program Files\Gensym\ /usr/gensym/g2/kbs	Windows pathnames UNIX pathnames
spreadsh.kb	File names
g2 -kb top.kb	Operating system commands
public void main() gsi_start	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 OPCLink User? Guide*
- *G2 PI Bridge User? Guide*
- *G2-SNMP Bridge User? Guide*
- *G2 CORBALink 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 and by email.

## To obtain customer support online:

➔ Access Ignite Support Portal at <https://support.ignitetechnology.com>.

You will be asked to log in to an existing account or create a new account if necessary. Ignite Support Portal 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 or email:

➔ Use the following numbers and addresses:

**United States Toll-Free** +1-855-453-8174

**United States Toll** +1-512-861-2859

**Email** [support@ignitetechnology.com](mailto:support@ignitetechnology.com)



# Introduction

---

*Describes the capabilities of G2 WebLink, and explains basic principles and concepts underlying the operation of G2 WebLink.*

Introduction	1
Communicating with G2 WebLink	2
Using G2 WebLink as an HTTP Server	3
Data Types Used by G2 WebLink	4
Defining the Default Character Set	5
Transcripts of G2 WebLink Operation	5



## Introduction

G2 WebLink enables Internet users to access G2 applications via Web browsers on their desktop machines. With G2 WebLink, you can use commonly available Web browsers, such as Microsoft's Internet Explorer or Netscape Communicator, as client front ends for information available within a G2 knowledge base (KB).

Using Web browsers, a large number of people can access a G2 application on their desk top machines. People who want to make fuller use of the G2 application can convert to a Telewindows interface.

You use G2 WebLink to implement an [HTTP server](#). You can configure G2 WebLink to satisfy requests that it receives from an HTTP client either by fetching a resource from the file system or by passing the request to G2, which can then generate appropriate responses.

G2 WebLink consists of the following software components:

- A G2 WebLink executable.
- G2 Knowledge Bases that provide useful utilities as well as examples that illustrate how G2 WebLink can be used.

G2 WebLink also requires a browser such as Microsoft's Internet Explorer or Netscape Communicator to use for testing.

## Communicating with G2 WebLink

G2 and G2 WebLink communicate with each other by making [remote procedure calls](#). The remote procedure calls perform operations such as sending configuration information to G2 WebLink, and pinging and killing G2 WebLink. Some of these procedures return meaningful values or signals when errors occur. For detailed descriptions of these procedures, see [Remote Procedures](#).

G2 connects to G2 WebLink through a GSI Interface Object. The KB `gwlow.kb` contains a fully configured GSI Interface Object, which you can use without modification for most purposes. This KB also contains objects that you must edit to configure the connection between G2 and G2 WebLink.

G2 WebLink provides a procedure, [monitor-gsi-interface-to-gw\(\)](#), that ensures reliable and continuous service from G2 WebLink. For most purposes, you can use this procedure without modification. For information about `monitor-gsi-interface-to-gw()`, see [Ensuring a Reliable Connection](#).

## Configuring the Behavior of G2 WebLink

After your G2 application establishes a connection to G2 WebLink, it must configure G2 WebLink by calling the procedure `gw-configure-bridge()`. [gw-configure-bridge\(\)](#) references an instance of the G2 class `gw-bridge-configuration`. Attributes of the instance of `gw-bridge-configuration` specify details of how G2 WebLink operates.

The top-level workspace of the KB `gwlow.kb` contains an instance of `gw-bridge-configuration`. You must edit this instance to specify the settings required by your site.

G2 must call `gw-configure-bridge()` before it attempts to have *any* other interactions with G2 WebLink. The KB `gwlow.kb` does this automatically. You can also call `gw-configure-bridge()` at any time while G2 WebLink is running to reconfigure its behavior.



## G2 WebLink Processes

G2 WebLink is designed to ensure that remote procedure calls do not back up. If a remote procedure request can not return immediately, a task is spawned inside G2 WebLink to finish the request.

## Using G2 WebLink as an HTTP Server

To enable G2 to provide resources to HTTP clients the G2 must ask G2 WebLink to establish a listener for server requests. The procedure [gw-start-http-server\(\)](#) is used to do this.

You may need to direct your users to configure their Web browsers to use this server.

Requests from those clients are then fulfilled in the following ways:

- Resources are obtained from a file system that is visible to the computer that G2 WebLink is running on.
- Resources are obtained from G2.

## Identifying Resources

A Universal Resource Locator (URL) is a pointer to a resource in the Internet. Each URL contains enough information to enable that resource to be located and retrieved.

HTTP clients use URLs to identify resources in G2. In the interface between G2 and G2 WebLink, URLs are represented as **text**.

Each URL has three parts:

- The name of the protocol used to access the resource, followed by a colon (:).
- A double slash (//), followed by the name of the Internet server where the resource can be retrieved.
- A single slash (/) followed by a path to the resource. Interpretation of the path is entirely up to the server.

The following examples illustrate the format of URLs:

```
ftp://www.ai.mit.edu/
http://tycho.usno.navy.mil/time.html
http://www.gensym.com/
gopher://weather.service.net/ma/boston/current-conditions
```

In the last example, the protocol is `gopher`, the server is `weather.service.net`, and the path to the resource on that server is `/ma/boston/current-conditions`.

For more information about URLs, see the Internet standards documentation.

The HTTP server established by G2 WebLink provides resources to its clients named by URLs, such as the following:

```
http://server:port-number/path/to/resource.html
```

where *server* is the machine name of the machine that your server is running on, and *port-number* is the number of the port on which the server is listening for requests. For information about how to configure the server to listen on a particular port, see [Configuring the Connection between G2 and G2 WebLink](#).

## Serving Resources

After launching the server, you must configure the server to specify the resources that G2 can provide to Internet clients. To do this, call the procedure `gw-tender-resources()`.

G2 configures G2 WebLink to directly serve resources by installing patterns that G2 WebLink then matches against the requested URLs. The procedure [gw-tender-resources\(\)](#) installs a pattern for a resource that G2 WebLink can provide.

You can install patterns to provide resources stored in files visible to G2 WebLink, or by calling back to G2.

## Data Types Used by G2 WebLink

The following table summarizes the data types used by G2 WebLink.

**Data Types Used by G2 WebLink**

<b>Data Type</b>	<b>Use</b>
text	Characters are the individual elements of text or text strings.
integer	Arguments of certain procedures.
symbol	Symbols are used in the bridge utility KB to denote various errors.
text	The G2 data type <code>text</code> is also a MIME Type and G2 WebLink requests servers to encode the characters in text they provide, using the Unicode UTF-8. This text is then converted into the text encoding used in G2 as outlined in the <i>G2 Reference Manual</i> . Line terminators are normalized, and tab characters are replaced with a single space.

## Defining the Default Character Set

By default, G2 WebLink uses UTF-8 as the default character set for Web pages generated from G2. You can specify one of a subset of the supported G2 character sets, either as the default character set, using a command-line option or for individual resources, using an RPC.

Use the following command-line option to change the default character set:

```
-default-charset charset-name
```

The options for *charset-name* are:

- unicode-utf-8
- us-ascii
- latin-1
- latin-2
- latin-3
- latin-4
- latin-cyrillic
- latin-arabic
- latin-greek
- latin-hebrew
- latin-5
- latin-6
- jis-euc
- shift-jis
- ksc-euc

To override the default character set for a given computed Web page, use [gw-set-url-reply-charset](#).

## Transcripts of G2 WebLink Operation

While G2 WebLink is running, it displays information about its operation on the background screen of the console where it was invoked. This transcript or log is formatted to include the date and the IP address of clients that make HTTP requests.

If G2 WebLink suffers a crash or internal error, the background transcript will include a backtrace that attempts to record exactly what G2 WebLink was doing

at the time of the failure. When reporting the failure please include this information along with as much of the transcript as possible.

# Configuring G2 WebLink

---

*Describes steps that you must always take to configure G2 WebLink, as well as some optional steps that you can take to customize how G2 WebLink behaves.*

Introduction 7

Running G2 WebLink 7

Configuring G2 WebLink 9

Optional Steps for Configuring G2 WebLink 11



## Introduction

This chapter describes how to run and configure G2 WebLink.

## Running G2 WebLink

You can run G2 WebLink from the command line or from the Start menu on Windows platforms. The command takes an optional argument to specify the TCP/IP port number where it listens for G2 to connect. By default, it listens on the default port, which is 7736.

**To run G2 WebLink on Windows platforms:**

➔ Choose Start > Programs > Gensym G2 8.3r0 > Bridges > G2 Weblink.

**To run G2 WebLink from a command line:**

- ➔ In a command window or shell, execute `run-gw.bat` or `gw`, depending on your platform, which are located in the `gw` subdirectory of your G2 product directory.

On UNIX platforms, you must set the `GSI_ROOT` environment variable to run the script. For more information, see `readme-g2.html`.

**To test the WebLink shortcut:**

- 1 Start G2 and start G2 Weblink, in either order.  
G2 WebLink starts the process running.
- 2 In G2, load `gwex1.kb`, which is located in the `kbs` subdirectory of the `gw` directory in your G2 Bundle installation directory.
- 3 Start G2 running from the G2 Main Menu.
- 4 On the workspace named HTTP Server Machinery, click the button labelled `start-and-configure-http-server()`.
- 5 Launch a Web browser on the local machine, and enter this URL:

`http://localhost:8080/home.htm`

You should see something this on your page:

GWEX1.KB -- an example Home Page

---

A very simple page!

---

Time: 19 Dec 2001 12:12:23 p.m.

---

Gensym's Home Page

## Configuring G2 WebLink

You can configure G2 WebLink for your site and application by editing objects in the workspace `gw-configuration-workspace`, which is the top-level workspace of the module `gflow`. The `gflow` module is stored in the module file `gflow.kb`.

Before you run G2 WebLink, you *must* edit a free text named `gw-desired-connection-configuration`, which configures the connection between G2 and G2 WebLink.

You can optionally edit the following objects on `gw-configuration-workspace`:

- A GW Bridge Configuration (a `gw-bridge-configuration`), which specifies how G2 WebLink behaves.
- A procedure named `monitor-gsi-interface-to-gw()` and an initially rule that starts this procedure. The procedure maintains and initializes the connection. In most situations, you will not need to modify this rule or procedure.

The following sections explain how to edit the objects on `gw-configuration-workspace` to configure your G2 WebLink application.

### Configuring the Connection between G2 and G2 WebLink

To configure the connection between G2 WebLink and G2, edit the *free-text* that appears near the top of `gw-configuration-workspace`. For most purposes, editing this *free-text* is the only thing that you need to do to configure the connection. G2 WebLink uses the *free-text* to set the value of the `gsi-connection-configuration` attribute of a GSI Interface Object, which configures the connection.

The initial value of the *free-text* is:

```
tcp-ip host "localhost" port-number 7736
```

When you edit the *free-text*, use the syntax required for the `gsi-connection-configuration` attribute of a GSI Interface Object. The syntax for the expression depends upon the protocol that the computer running G2 uses to connect to the computer running G2 WebLink.

In order for G2 to establish a connection to the G2 WebLink process, the `gsi-connection-configuration` attribute must match the name of the computer that runs the bridge process.

The following expression specifies a G2 WebLink process with port number 22041 that runs on the computer `localhost`, and to which G2 connects through a network connection over the TCP/IP protocol:

```
tcp-ip host "localhost" port-number 22041
```

## Configuring G2 WebLink Behavior

To configure G2 WebLink behavior, follow these steps:

- 1 Open the attribute table of the GW Bridge Configuration object that you find on `gw-configuration-workspace`.
- 2 Edit the attributes of the GW Bridge Configuration object to specify details of how G2 WebLink operates.

The following table lists attributes of a GW Bridge Configuration object that you can set. The default values listed are those of the preconfigured GW Bridge Configuration object provided with `gflow.kb`.

### Attributes of GW Bridge Configuration Object

---

<b>maximum-count-of-simultaneous-transactions</b>	Specifies how many clients the G2 WebLink server can serve at one time. The server will decline client requests in excess of this number.
---	---

*Allowable values:* Any integer

*Default value:* 25

<b>message-pool-size</b>	Not used.
--------------------------	-----------

*Allowable values:*

*Default value:* 100

After your G2 application establishes a connection to G2 WebLink, it must call the remote procedure [gw-configure-bridge\(\)](#) to send the configuration information in the GW Bridge Configuration to G2 WebLink.

For more information about `gw-configure-bridge()` and the attributes of a `gw-bridge-configuration`, see [Remote Procedures](#).

## Checking the Status of the Connection to the Bridge

You can monitor the status of the connection between G2 and the bridge by checking the value of the symbolic variable `gw-status-of-connection-to-bridge`, in `gw-configuration-workspace`.

This variable is maintained by the `gflow` module and is used by code in your KB to determine the state of the connection to the bridge.



`gw-status-of-connection-to-bridge` can take the following symbolic values:

```
OK
KB-WAS-RESET
AWAITING-GSI-CONNECTION-TO-BRIDGE
CONNECTED-BUT-AWAITING-CONFIGURATION-OF-BRIDGE
```

Remote procedure calls to the bridge succeed only when the value `gw-status-of-connection-to-bridge` of is OK.

## Optional Steps for Configuring G2 WebLink

You can optionally modify how G2 maintains the connection to G2 WebLink by customizing the G2 procedure `monitor-gsi-interface-to-gw()`, which ensures the reliable operation of the connection between G2 and G2 WebLink.

The version of `monitor-gsi-interface-to-gw()` provided with G2 WebLink should be suitable for almost all purposes. Do not modify this procedure unless you have a specific need to do so, and are certain that you understand how the procedure works.

### Ensuring a Reliable Connection

The maintenance of a reliable connection to G2 WebLink is supported by the procedure `monitor-gsi-interface-to-gw()`. One instance of this procedure should be launched for the `gsi` connection. It calls `gw-ping()` at regular intervals, and if it finds that the connection is broken, it attempts to reestablish the connection after closing the old one.

The OS shell script that runs G2 WebLink should usually be configured to restart G2 WebLink automatically if it shuts down. This improves reliability if G2 WebLink were to crash. It is also useful if the G2 shuts G2 WebLink down explicitly.

---

**Note** The version of `monitor-gsi-interface-to-gw()` provided with G2 WebLink should be suitable for almost all purposes. Do not modify this procedure unless you have a specific need to do so, and are certain that you understand how the procedure works.

---

### Logging Host Names

With the release of G2 WebLink 1.0 Rev. 3, the G2 WebLink bridge stopped logging the names of clients that make HTTP requests to G2 WebLink. By default, the bridge only logs the numeric IP address. In most cases, this will dramatically improve the response time of G2 WebLink.

To restore the pre-1.0 Rev. 3 behavior, one must start G2 Weblink with the `-enable-hostname-lookups` flag.

# Remote Procedures

---

*Describes the remote procedures that are provided with G2 WebLink.*

Introduction	13
gw-add-to-url-reply	16
gw-configure-bridge	18
gw-decline-to-serve-url	19
gw-finish-url-reply	21
gw-get-attribute-of-request	23
gw-kill-bridge	29
gw-make-base64-encoded	30
gw-ping	31
gw-set-header-field	32
gw-set-url-reply-charset	33
gw-start-http-server	35
gw-tender-resources	37



## Introduction

G2 communicates with G2 WebLink by making remote procedure calls. Your G2 application can invoke these remote procedures from user-defined procedures, action buttons, or in other standard ways.

G2 WebLink can make remote procedure calls into G2 to request services from G2 for Internet clients, or to notify G2 of the status of the bridge.

## Summary of Remote Procedures

The following tables lists the remote procedures that you use to configure, manage, and run G2 WebLink.

### Remote Procedures for G2 WebLink

Purpose	Procedure	Description
Configure G2 WebLink	<a href="#">gw-configure-bridge</a>	(Required) Configures G2 WebLink with information required for its operation.
Support G2 as Internet Server	<a href="#">gw-start-http-server</a>	Starts an HTTP server, enabling an Internet client to access data in G2.
	<a href="#">gw-tender-resources</a>	Configures a resource that G2 WebLink, as a server, can provide to HTTP clients.
	<a href="#">gw-get-attribute-of-reques</a>	Gets an attribute from an HTTP request.
	<a href="#">gw-add-to-url-reply</a>	Builds up a resource in response to a request by G2 WebLink for a particular URL.
	<a href="#">gw-finish-url-reply</a>	Closes a connection and performs cleanup tasks. Call this procedure when G2 has successfully generated a resource in response to a request from G2 WebLink.
Support G2 as Internet Client	<a href="#">gw-decline-to-serve-url</a>	Declines a request for a URL for which G2 is unable to provide a resource.
	<a href="#">gw-make-base64-encoded</a>	Converts a text string (generally, a user name and password) into a slightly encoded text string.
Miscellaneous Procedures	<a href="#">gw-ping</a>	Determines if G2 WebLink is running.
	<a href="#">gw-kill-bridge</a>	Shuts down G2 WebLink.

### Remote Procedures for G2 WebLink

Purpose	Procedure	Description
	<a href="#">gw-set-url-reply-charset</a>	Sets the character set for an individual resource.
	<a href="#">gw-set-header-field</a>	Sets the header field.

## Return Values and Signals of Remote Procedures

Some remote procedures return meaningful values, such as signals raised when errors occur. If a procedure can return a value or a signal, the value or signal is listed with the procedure.

For information about how your G2 application can handle signals, see the *G2 Reference Manual*.

# gw-add-to-url-reply

Builds up a resource in response to a request by G2 WebLink for a particular URL.

## Synopsis

gw-add-to-url-reply  
(*fetch-id*: integer, *more-data*: text)

Argument	Description
<i>fetch-id</i>	Should match the <i>fetch-id</i> argument of the user-defined response procedure that generates the resource. You specify this procedure in the <i>response</i> argument of the call to <a href="#">gw-tender-resources</a> .  You must reference this same <i>fetch-id</i> in the call to <a href="#">gw-finish-url-reply</a> that closes the connection when your G2 application has finished building and sending the reply.
<i>more-data</i>	Text added to the reply sent to the HTTP client. Usually, this data is in HTML format.

## Description

When the bridge receives a request from an HTTP client, and the URL used in the request matches one of the patterns tendered by your G2 application, the bridge calls the response procedure.

In the response procedure, calls to `gw-add-to-url-reply` build up a reply. After each call, the bridge passes the additional data on to the HTTP client, and the client usually displays that additional bit to its user.

When your G2 application has provided the entire reply, it should call [gw-finish-url-reply](#). This procedure closes the connection and performs cleanup tasks associated with the closing of the connection.

## Signals

Signal	Description
invalid-url-request-number	Indicates that when the call was received by G2 WebLink, no client request with this ID was in progress. This may indicate that the client closed the connection before G2 finished it.

## Related Procedures

Procedure	Description
gw-tender-resources	Configures the resources that G2 WebLink, as a server, can provide to HTTP clients.
gw-finish-url-reply	Closes a connection and performs cleanup tasks.
gw-decline-to-serve-url	Declines a request for a URL for which G2 is unable to provide a resource.

# gw-configure-bridge

Configures G2 WebLink using configuration information in a specified instance of the G2 class `gw-bridge-configuration`.

## Synopsis

`gw-configure-bridge`  
(*configuration*: class `gw-bridge-configuration`)

Argument	Description
<i>configuration</i>	An instance of the G2 class <a href="#">gw-bridge-configuration</a> , whose attributes specify how the bridge operates.

## Description

A G2 application must call the remote procedure `gw-configure-bridge` *before* it has any other interaction with the bridge. Your G2 application cannot communicate with the bridge until the bridge has been properly configured with the information in the instance of `gw-bridge-configuration`.

The KB `gwlow` automatically configures the bridge for you when the bridge first connects to G2.

G2 can invoke `gw-configure-bridge` at any time during a bridge session to change the configuration.

In the following example, the bridge is initialized to the settings found in the instance named `my-bridge-configuration`:

```
gw-configure-bridge(my-bridge-configuration)
```



# gw-decline-to-serve-url

Invoked by a response procedure in G2 when G2 WebLink requests a URL for which G2 is unable to provide a resource.

## Synopsis

gw-decline-to-serve-url  
(*fetch-id*: integer, *apology*: text)

Argument	Description
<i>fetch-id</i>	Should match the number provided in the call made by G2 WebLink to G2 requesting the resource.
<i>apology</i>	Text that is included in the result that is returned to the HTTP client. The result returned to the client is marked as an error.

## Description

When the bridge requests a resource from G2, the URL that it uses in the request must match one of the tendered patterns that G2 provided through [gw-tender-resources](#). If G2 cannot provide a resource for a URL that matches a tendered pattern, it calls gw-decline-to-serve-url to notify the client making the request.

A G2 application may not be able to serve some requests even though the URLs for those requests match patterns that the G2 application has tendered. This can happen because a tendered pattern can match many URLs – including URLs for resources that the G2 application does not provide. A URL can also request a resource that the G2 application does in fact provide, but at a time when the resource is not available. In these cases, G2 notifies the client requesting the resource by calling gw-decline-to-serve-url.

## Signals

Signal	Description
invalid-url-request-number	Indicates that when the call was received by G2 WebLink, no client request with this ID was in progress. This may indicate that the client closed the connection before G2 finished it.

## Related Procedures

<b>Procedure</b>	<b>Description</b>
gw-tender-resources	Configures the resources for which G2 WebLink is a server to HTTP clients.
gw-add-to-url-reply	Builds up a resource in response to a request by G2 WebLink for a particular URL.
gw-finish-url-reply	Closes a connection and performs cleanup tasks. Call this procedure when G2 has successfully generated a resource in response to a request from G2 WebLink.

# gw-finish-url-reply

Called by a response procedure when it has finished generating the resource requested of it.

## Synopsis

gw-finish-url-reply  
(*fetch-id*: integer)

Argument	Description
<i>fetch-id</i>	Should match the number provided when G2 WebLink invoked the routine specified with the URL pattern in the call to <a href="#">gw-tender-resources</a> .

## Description

When a response procedure finishes generating a resource in response to a request from the bridge, it must call [gw-finish-url-reply](#). This procedure closes the connection to the Internet client requesting that resource and performs clean up tasks associated with the closing of that connection.

Any response procedure must either call [gw-finish-url-reply](#) when it has successfully generated a resource, or [gw-decline-to-serve-url](#) if it cannot generate the resource.

Response procedures are invoked through the procedure [gw-tender-resources](#). A response procedure in G2 generates a resource by making one or more calls to [gw-add-to-url-reply](#).

## Signals

Signal	Description
invalid-url-request-number	Indicates that when the call was received by G2 WebLink, no client request with this ID was in progress. This may indicate that the client closed the connection before G2 finished it.

## Related Procedures

Procedure	Description
gw-tender-resources	Configures the resources that G2 WebLink, acting as a server, can provide to HTTP clients.
gw-add-to-url-reply	Invoked by G2 one or more times to build up a resource in response to a request by G2 WebLink for a particular URL.
gw-decline-to-serve-url	Invoked by a response procedure in G2 when G2 WebLink requests a URL for which G2 is unable to provide a resource.

# gw-get-attribute-of-request

Called by a response procedure in G2 to obtain information about a request to which it is responding.

## Synopsis

```
gw-get-attribute-of-request
  (fetch-id: integer, attribute-name: text)
  -> text: text
```

Argument	Description
<i>fetch-id</i>	Should match the number provided when G2 WebLink invoked the routine specified with the URL pattern in the call to <a href="#">gw-tender-resources</a> .
<i>attribute-name</i>	The name of the attribute whose value is to be returned by this call to <code>gw-get-attribute-of-request</code> . For information about how to specify attribute names, see Description below.

## Description

`gw-get-attribute-of-request` gets an attribute from an HTTP request. A response procedure can call `gw-get-attribute-of-request` to obtain more information about a request to which it is responding.

Information about a request can also be passed to a response procedure through the response procedure's arguments, as specified in a call to `gw-tender-resources`. The *attribute-name* arguments described below can all be passed as arguments to a response procedure.

You can call `gw-get-attribute-of-request` during the interval between the invocation of the response procedure and the call terminating the response. A response is terminated by a call either to `gw-decline-to-serve-url` or to `gw-finish-url-reply`.

For the *attribute-name* argument in a call to `gw-get-attribute-of-request`, you can specify:

- The name of a MIME header.
- The name of a CGI argument.
- Other pre-defined argument names, which are listed below.

The following paragraphs describe how to specify *attribute-name* in these ways.

### **MIME Headers**

To get an individual MIME headers from the HTTP request, specify an attribute name beginning with the prefix HTTP-, followed by the rest of the name of the MIME header. For example, the attribute name:

"HTTP-User-Agent"

returns the header named "HTTP-User-Agent", which is a description of the client's Web browser, if the browser provides one. An example of such as header is "Mozilla/2.0 (X11;I;SunOS 4.1.4 Sun4m)".

---

**Note** The text of MIME headers is case-sensitive.

---

### **CGI Arguments**

Arguments in a URL formatted in conformance with the Common Gateway Interface (CGI) standard can be specified by name. For example, in the following URL, the attribute value "data" is returned if the attribute name "filter" is specified as *argument-name* in the call to *gw-get-attribute-of-request*:

"http://www.grsmithinc.com/query?id=123&filter=data"

## Other Argument Names

You can also specify the following names for *argument-type* in a call to `gw-get-attribute-of-request`:

Argument Type	Description
"auth-type"	The name of the authorization scheme used in this request. For example: "BASIC". This is an empty string if no authorization is included with the request. See the Internet standard "Hypertext Transfer Protocol -- HTTP/1.0" for additional details about terms used in this definition.
"content-length"	Returns the number of bytes in the content. This is the value provided in the headers passed with the request. Remember that the client is not required to provide an accurate value. For example: "1037".  <b>Note:</b> When the content length is used as an argument to a response procedure, the data type passed is an integer. The value -1 is passed if the length is not specified in the request's header.
"content-type"	Returns the empty string in most cases. If the request is using the request-method "PUT" and a MIME header that explicitly states a content type, then this string will provide that type. For example: "IMAGE/GIF".
"fetch-id"	Returns the fetch id for the current request.  <b>Note:</b> When the fetch id is used as an argument to a response procedure, the data type passed is an integer.
"gateway-interface"	This is always the empty string. The Internet Draft Standard for CGI/1.1 defines a number of environment variables that do not have a useful implementation in the bridge at this time. These return the empty string ("").

Argument Type	Description
"path-info"	<p>The name of the resource requested by the client. For example, in the URL "http://www.grsmithinc.com:8080/pages/index.html", the path-info is the string "pages/index.html". In the URL "http://www.grsmithinc.com:8080/", the path-info is "".</p> <p><b>Note:</b> path-info is the hpath token defined in RFC 1738 and the Path-info described in the CGI/1.1 spec.</p>
"path-translated"	<p>This is always an empty string. The Internet Draft Standard for CGI/1.1 defines a number of environment variables that do not have a useful implementation in the bridge at this time. These are included in the list above and return the empty string ("").</p>
"query-string"	<p>This is the text following the "?" in CGI queries. For example, in the URL "http://www.casper.com/query?id=123&amp;filter=data" the query string is "id=123&amp;filter=data" Generally, it is not necessary to obtain this, because the query string is automatically parsed and bound to arguments when the response procedure is invoked through the call cgi syntax.</p> <p><b>Note:</b> In RFC 1738, query-string is referred to as the "&lt;searchpart&gt;" and as the "query string".</p>
"remote-addr"	<p>This string reports the numeric address of the client's machine. Note this might not be the machine the user is on, because the user's request may have been forwarded through a proxy server. If the address cannot be obtained – a rare event – an empty string is returned.</p>
"remote-host"	<p>This string reports the name of the client's machine. Note this might not be the machine the user is on, because the user's request may have been forwarded through a proxy server. If the address cannot be obtained – a rare event – an empty string is returned.</p>



Argument Type	Description
"remote-ident"	This is always an empty string. The Internet Draft Standard for CGI/1.1 defines a number of environment variables that do not have a useful implementation in the bridge at this time. These are included in the list above and return the empty string (i.e. "").
"remote-user"	If the request includes authorization information, and the <b>auth-type</b> is one of those supported by the bridge, this string identifies the user.
"request-method"	This is one of three values: "GET", "PUT", or "HEAD". See the Internet standard "Hypertext Transfer Protocol -- HTTP/1.0" for additional details about terms used in this definition.
"request-mimi-headers"	This is the text of MIME headers included with the request. This is exactly as provided by the client software. This can return the empty string. For example, the MIME header returned can be: <pre> Connection: Keep-Alive User-Agent: Mozilla/2.0 (X11; I; SunOS 4.1.4 sun4m) Pragma: no-cache Host: tha:8080 Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, */* " </pre>
"script-name"	This is name of the response procedure invoked to handle this request.
"server-name"	This is the TCP/IP port upon which the server received this request. For example: "www.grsmithinc.com:8080".
"server-port"	This is the port number upon which the server is listening for incoming requests. For example: "8080".
"server-protocol"	This is the machine name upon which the server is listening for incoming requests. For example: "www.grsmithinc.com".

Argument Type	Description
"server-software"	This returns the name of the the bridge software, including the version number. For example: "G2 WebLink v1.3 - c23"
"url-details"	This the same as "path-info".

## Return Values

Return Value	Description
<u>text</u>	The value of the attribute specified by <i>attribute-name</i> . G2 WebLink can pass this attribute to the G2 procedure that it invokes to generate the resource requested by the HTTP request.

# gw-kill-bridge

Shuts down G2 WebLink.

## Synopsis

```
gw-kill-bridge  
( )
```

## Description

The remote procedure `gw-kill-bridge` shuts down the bridge. It takes no arguments returns no return value or signal.

# gw-make-base64-encoded

Converts a text string into a slightly encoded text string.

## Synopsis

```
gw-make-base64-encoded  
  (text-to-encode: text)  
  -> text: text
```

Argument	Description
<i>text-to-encode</i>	The user name and password that you want to encode.

## Description

The remote procedure `gw-make-base64-encoded` is needed when G2 requests resources from a server that requires a user name and password.

In this situation, the request from G2 to the server includes headers that identify the user to the server. While many techniques can be used to encode this information, the only technique specified in HTTP version 1.0 is known as the "Basic Authentication Scheme." The encoding done is specified in RFC 1521.

For example, if the user named "Aladdin" wishes to obtain a resource from a server where his password is "open sesame", he can include in his request a header of the form:

```
Authorization: Basic QWxhZGRpbjpvYVUuIHNIc2FtZQ==
```

For example, the following call:

```
coded = call gw-make-base-64-encoded("Aladdin:open sesame");
```

results in text value "QWxhZGRpbjpvYVUuIHNIc2FtZQ==" being stored into `coded`.

## Return Values

Return Value	Description
<u>text</u>	The encoded user name and password.

# gw-ping

Determines whether or not G2 WebLink is running.

## Synopsis

```
gw-ping
  ()
```

## Description

The remote procedure `gw-ping` takes no arguments, and returns no values. The G2 knowledge base can use it to find out whether or not the bridge is running. It should return almost immediately.

Its failure to be invoked indicates that the interface is disconnected. Its failure to return indicates that, for some reason the bridge is hung.

## Signals

Signal	Description
bridge-connection-down	Indicates that the G2 WebLink connection is no longer active.

## Related Procedures

Procedure	Description
gw-kill-bridge	Shuts down G2 WebLink.

# gw-set-header-field

Sets the header field for an individual G2 WebLink resource.

## Synopsis

gw-set-header-field

(*fetch-id*: integer, *header*: text, *data*: text)

Argument	Description
<i>fetch-id</i>	Should match the number provided when G2 WebLink invoked the routine specified with the URL pattern in the call to <a href="#">gw-tender-resources</a> .
<i>header</i>	The name of the header, for example, "content-type".
<i>data</i>	The value of the header, for example, "text/html".

## Description

For example, the following procedure call:

```
gw-set-header-field(1, "content-type", "text/html")
```

emits the following in the header fields of the resulting HTML page as presented in the browser:

```
content-type: text/html
```

You must call this procedure before your first call to gw-add-to-url-reply; otherwise, an error is generated.

The header fields are presented to the browser in the inverse order in which they are presented with gw-set-header-field. From the HTTP specification: "The order in which header fields with differing field names are received is not significant. However, it is good practice to send general-header fields first, followed by request-header or response-header fields, and ending with the entity-header fields. For details, see <http://www.w3.org/Protocols/rfc2616/rfc2616-sec4.html#sec4.2>.

# gw-set-url-reply-charset

Sets the character set for an individual G2 WebLink resource.

## Synopsis

gw-set-url-reply-charset  
*(fetch-id: integer, charset-name: symbol)*

Argument	Description
<i>fetch-id</i>	Should match the number provided when G2 WebLink invoked the routine specified with the URL pattern in the call to <a href="#">gw-tender-resources</a> .
<i>charset-name</i>	The character set to use for the individual resource. See Description for the list of options. The default value is <code>unicode-utf-8</code> .

## Description

Use this remote procedure to override the default character set for a given computed Web page. You must call this procedure before any calls to `gw-add-to-url-reply`; otherwise, an error is generated.

The options for *charset-name* are:

- `unicode-utf-8`
- `us-ascii`
- `latin-1`
- `latin-2`
- `latin-3`
- `latin-4`
- `latin-cyrillic`
- `latin-arabic`
- `latin-greek`
- `latin-hebrew`
- `latin-5`
- `latin-6`

- jis-euc
- shift-jis
- ksc-euc



# gw-start-http-server

Starts an HTTP server.

## Synopsis

gw-start-http-server  
(*port-number*: integer)

Argument	Description
<i>port-number</i>	<p>The port number where the HTTP server appears.</p> <p>The Internet default port number for HTTP servers is 80. Most operating systems require special privileges for the use of this port number.</p>

## Description

The remote procedure `gw-start-http-server` starts up an HTTP server. Clients in the Internet, such as Web browsers, can then use this server to access resources that G2 makes available.

After starting the server, G2 should invoke [gw-tender-resources](#) to configure the resources that the server offers. The HTTP server offers only the resources that are configured by the call to `gw-tender-resources`.

## Signals

Signal	Description
http-startup-failed	<p>Signalled when <code>gw-start-http-server</code> is unable to start a server. This can happen if, for example, the port where the server is to listen for requests is already in use, or if the operating system is configured to prevent listening on that port. Some operating systems do not allow listening on a port until one or two minutes after the most recent use of that port.</p>

## Related Procedures

Procedure	Description
gw-tender-resources	Configures a resource that G2 WebLink, acting as a server, can provide to an HTTP client.

# gw-tender-resources

Configures a resource that G2 WebLink, acting as a server, can provide to an HTTP client.

## Synopsis

gw-tender-resources

(*index*: integer, *pattern*: text, *response*: text)

Argument	Description
<i>index</i>	A unique integer value that distinguishes this tender offer from all others.  The range for valid index values is 0 - 1000, inclusive.
<i>pattern</i>	Matches a URL used to request a resource.
<i>response</i>	Specifies the action taken in G2 in response to the URL that matches <i>pattern</i> .

## Description

gw-tender-resources instructs the bridge how to respond to requests from HTTP clients for resources that G2 can provide. Each call to gw-tender-resources defines a pattern that can match one or more of the URLs used by HTTP clients to request resources.

When a URL used in a client request matches a tendered pattern, and G2 can provide a resource in response to that request, the resource is generated in a way specified by gw-tender-resources. A resource can be generated by a user-defined response procedure in G2, or it can be obtained from the file system.

You must call gw-tender-resources once for each class of resources that you want to make available to Internet clients. The bridge can respond to a request for a G2 resource only if that resource has been configured by a call to gw-tender-resources.

gw-tender-resources must be called *after* G2 has started the server by calling [gw-start-http-server](#).

## Specifying an Index

The *index* argument makes it possible to distinguish patterns from each other. Using an *index* to distinguish patterns has the following advantages:

- If more than one pattern matches the same URL, the bridge responds to the URL by using the pattern with the lowest *index* value.
- You can replace a pattern by tendering a different pattern at the same index.

---

**Note** Each *index* value must be unique.

---

## Specifying a Pattern

The *pattern* argument must match part of the URL that requests the resource. For example, the following URL:

```
http://www.grsmithinc.com/news/homeoffice
```

can be matched by any of the following *pattern* arguments:

```
"news"  
"news/home"  
"news/homeoffice"
```

As this example shows, more than one *pattern* can match the same URL.

An empty pattern ("") matches any URL. You can use an empty pattern to tender a resource for all requests that do not match any other tendered patterns. In this case, the call that specifies the empty pattern should also specify a higher index value than other calls, so that a request is matched to the empty pattern only after that request has failed to match any of the other patterns.

## Specifying a Response

The *response* string specifies how the bridge generates the resource. You can specify a response using **call** or **serve** formats:

- The **call** response formats are used to request dynamically computed resources from G2. These resources are generated by response procedures in G2.
- The **serve** formats are used to request existing resources that G2 can provide without computation. These resources are obtained from the file system.

## The call Response Format

This format specifies a user-defined G2 procedure that generates the requested resource. This procedure is known as a [response procedure](#). You are responsible for providing this procedure in your G2 application.

The response procedure is invoked by the bridge when the bridge receives an HTTP request that matches the pattern specified in the call to `gw-tender-resources`. As arguments to the G2 procedure, the bridge passes a set of attributes that it produces by analyzing the HTTP request. To get attributes from an HTTP request, the response procedure invokes the procedure [gw-get-attribute-of-request](#). See the description of this procedure for information about the attributes in an HTTP request that can be returned by `gw-get-attribute-of-request`.

The response procedure provides the requested resource by making calls to the procedures `gw-add-to-url-reply` and `gw-finish-url-reply`. It can decline the request by making a single call to `gw-decline-to-serve-url`.

There are two call response formats:

`call url g2-function-name (arguments...)`

`call cgi g2-function-name (arguments...)`

`call url` responds to a URL in any format.

`call cgi` responds to a URL formatted in conformance with the Common Gateway Interface (CGI) standard. The `call cgi` format notifies the bridge that the URL conforms to CGI standards. The bridge extracts arguments from the URL and makes them available as attributes of the request.

`g2-function-name` is a user-defined response procedure that handles the request. The bridge calls the procedure by making a remote procedure call to G2.

`arguments...` represents one or more argument values passed to `g2-function-name`. These arguments provide G2 with the details about the resource requested. Each `argument` has the following syntax:

`argument-name : argument-type`

where:

`argument-name` is one of the attributes of the request that the response procedure needs in order to generate its response. As an `argument-name`, you can specify any of the possible values for an `attribute-name` argument in a call to [gw-get-attribute-of-request](#).

The list of argument name values must include one argument named `fetch-id`. The `fetch-id` argument denotes which request is being fulfilled when G2 is generating a reply to an HTTP request. Because the bridge can handle many requests simultaneously, each separate response must specify a unique `fetch-id` argument.

`argument-type` can be integer, text, or symbol.

The following example illustrates the use of the `call cgi` format:

```
gw-tender-resources( 300, "queries/lookup?"
  "call cgi lookup-personnel-file(fetch-id:integer, first:text, last:text)" )
```

This call to `gw-tender-resources` handles the URL by extracting the values of the `first` and `last` parameters from the URL, and passing these values to the user-defined procedure `lookup-personnel-file` in G2. Thus, the bridge handles the following URL:

```
http://www.genym.com/queries/lookup?last=smith&first=john
```

by calling `lookup-personnel-file` with `john` and `smith` as arguments.

## The serve Response Format

The `serve` response format is of the form:

```
serve from file system path
```

`Serve from file system path` is used to pass stable resources to the net that were authored previously and placed in the file system. For example, the following calls can be used to publish resources from different users at your site.

```
gw-tender-resources(490, "users/joe/",  
  "serve from file system: /home/joe/pub/")
```

```
gw-tender-resources(480, "users/sam/",  
  "serve from file system: /home/sam/pub/")
```

```
gw-tender-resources(470,  
  "users/doc/", "serve from file system: /home/projects/doc/www/")
```

## Signals

Signal	Description
tender-failed	This signal can be reported if the HTTP server is not currently running, either because you have not called <code>gw-start-http-server</code> , or because some error has forced the server to shut down. Errors can include use of an <i>index</i> value that is not between 0 ... 1000 (inclusive), or other syntax errors.

## Related Procedures

Procedure	Description
gw-add-to-url-reply	Builds up a resource in response to a request by G2 WebLink for a particular URL.

# Example KBs

---

*Describes the example KBs provided with G2 WebLink.*

Introduction **41**

Example of a Simple Server **42**

Example of a Server Providing a Choice of Resources **44**

Example of Setting the Header Field and URL Character Set **48**



## Introduction

The example KBs are located in the `kbs` subdirectory in the `gw` directory of your G2 Bundle installation directory.

# Example of a Simple Server

The example KB `gwex1.kb` consists of a server that provides a simple home page which includes a timestamp and a link to the Gensym home page.

**To use this example KB:**

- 1 Open the read-me workspace:

## READ-ME

This example KB is a very simple HTTP server that serves only one page. It is too simple to use as a starting point for your own server. The example `gwex2.kb` is a better for that.

1. Start the bridge, and notice the port number it is listening for G2 on.
2. Set up `GW-CONFIGURATION-WORKSPACE`.
3. Start G2 and notice that `GW-CONFIGURATION-WORKSPACE` should become a light white color, indicating it has connected to the bridge
4. The blue workspace `HTTP-SERVER-MACHINERY` is used to startup and configure the server. If you hit the button there the server will start up.
5. Ask your browser to fetch the page from the server. Use the URL "`http://server:8080/home.htm`" replacing "server" with the host name where you are running the server. (see note 1 below if this doesn't work).

Note 1: If your browser is configured to use a proxy server - so you can pass thru an internet firewall - you will need to add to its configuration to avoid the firewall when accessing your server. In Netscape, for example, this is done in the dialog where the proxy server was setup in a line labeled "No Proxy". There one would enter "server:8080".




**2** Follow the directions outlined in the `read-me` workspace.


These directions indicate how to:

- a** Start and configure the bridge.
- b** Start G2.
- c** Start the HTTP proxy server.
- d** Access the home page in your browser.

This example uses a G2 procedure named `start-and-configure-http-server` to start and configure the server. The G2 procedure `start-and-configure-http-server` illustrates the use of the remote procedures `gw-start-http-server` and `gw-tender-resources`:

```
 start start-and-configure-http-server()
```

```
START-AND-CONFIGURE-HTTP-SERVER
```

```
 start-and-configure-http-server()
tcp-port-for-http-server:integer = 8080;
begin

  { Ask the bridge to fire up a Web Server }
  call gw-start-http-server(tcp-port-for-http-
    server);

  { Tender-resource configures the bridge's
    handling of URL's }
  call gw-tender-resources
    (710, "home.htm", "call url generate-home-
      page(fetch-id:integer)");

end
```

The call to `gw-tender-resources` invokes the following response procedure, which generates the simple home page that you see in your browser:

#### GENERATE-HOME-PAGE



```
generate-home-page(fetch-id:integer)
{ This is invoked by WebLink when it wants
  to generate
  the page named "home.htm". That
  happens because of
  code in start-and-configure-http-server.}
begin
  {We build our reply via a series of calls,
   which are passed on to
   the user's web browser right away}
  call gw-add-to-url-reply(fetch-id,
    "<H1>GWEX1.KB -- an example Home
    Page</H1>
    <hr>
    <p>A <b>very</b> simple page!
    <p><hr>");
  { We can generate the text using state
   stored in G2 }
  call gw-add-to-url-reply(fetch-id,
    "Time: [the current real time as a time
    stamp]<hr>");
  { We can create links to other pages }
  call gw-add-to-url-reply(fetch-id,
    "<a
     href=http://www.gensym.com>Gensym's
     Home Page</a>");
  call gw-finish-url-reply(fetch-id);
end
```

## Example of a Server Providing a Choice of Resources

The example KB `gwex2.kb` consists of a server that provides a home page through which you can access a variety of resources in G2.

#### To use this example KB:

- 1 Open the read-me workspace.

The read-me workspace looks like this:

### READ-ME

This example KB is a simple HTTP server built using the bridge. Using it should be simple.

1. Start the bridge, notice the port number it is listening for G2 on.
2. Set up the GW-CONFIGURATION-WORKSPACE.
3. Start G2, and notice that GW-CONFIGURATION-WORKSPACE should become a light white color, indicating it has connected to the bridge
4. The blue workspace HTTP-SERVER-MACHINERY is used to startup and configure the server. If you hit the button there the server will start up.
5. Ask your browser to fetch a page from the server, using this URL "http://server:8080/home.htm", replacing "server" with the host name where you are running the server. (see note 1 below if this doesn't work).
6. The green workspace HTTP-SERVER-LOG will show activity taking place in G2 in support of the server.
7. The tan workspace GENERATING-URL contains the routines that actually generate resources for the server.

Note 1: If your browser is configured to use a proxy server - so you can pass thru an internet firewall - you will need to add to its configuration to avoid the firewall when accessing your server. In Netscape, for example, this is done in the dialog where the proxy server was setup in a line labeled "No Proxy". There one would enter "server:8080".

Note 2: The server is setup to tender some resources from the file system. In these examples that is only used for an image that appears on the generated pages. When this image (a picture of a bridge) doesn't appear on the generated pages then you will have to change the tender in startup-and-configure-bridge to point to the correct place in your file system.

- 2** Follow the directions outlined in the `read-me` workspace. These directions indicate how to:
  - a** Start and configure the bridge.
  - b** Start G2.
  - c** Start the HTTP proxy server.
  - d** Access the home page in your browser.

This example uses a G2 procedure named `start-and-configure-http-server` to start and configure the server. The G2 procedure `start-and-configure-http-server` looks like this:

```

start start-and-configure-http-server()

START-AND-CONFIGURE-HTTP-SERVER
start-and-configure-http-server()
tcp-port-for-http-server:integer = 8080;
begin
call log-http-server-activity("Launching HTTP
Server");

{ Ask the bridge to fire up a Web Server}
call gw-start-http-server(tcp-port-for-http-
server);

{ Tender-resource configures the bridge's
handling of URL's }

{ Some URL's are passed directly to GW }
call gw-tender-resources
(700,
"misc/",
"call url generate-misc-url(fetch-id:text, http-
client-host-name: text, url-details:text)");
call gw-tender-resources
(710,
"home.htm",
"call url generate-misc-url(fetch-id:text,
http-client-host-name: text, url-
details:text)");

{ Some URL's become RPC calls into G2. }
call gw-tender-resources
(800,
"query1?",
"call cgi generate-query1(fetch-id:integer,
kind:symbol, id:text)");

{ Most URL's fall thru to the file system }
call gw-tender-resources
(900,
"",
"serve from file system: /tmp/www");

call log-http-server-activity("HTTP Server
Launched.");
end

```

The G2 procedure `start-and-configure-http-server` calls `gw-tender-resources` four times. The calls illustrate how requests for resources, as specified by URLs, can be passed directly to G2, can be made through remote procedure calls to G2, or can be passed to the file system. The response procedures specified in the calls to `gw-tender-resources` are located on the `generating-url` workspace.

## Example of Setting the Header Field and URL Character Set

The example KB `gwex3.kb` shows how to use the `gw-set-header-field` system procedure. See the procedure named `generate-csv-page`.

The `gwex3.kb` also shows how to use the `gw-set-url-reply-charset`. See the procedure named `generate-cyrillic-page`.

See [Example of a Simple Server](#) for information on how the KB is configured.

# Using HTML and SGML

---

*Describes the use of Hypertext Markup Language (HTML) and Standard Generalized Markup Language (SGML).*

Introduction **49**

Using Hypertext Markup Language (HTML) **49**

Using Standard Generalized Markup Language **50**



## Introduction

This chapter describes how to use G2 WebLink with Hypertext Markup Language (HTML) and Standard Generalized Markup Language (SGML).

## Using Hypertext Markup Language (HTML)

HTML, or Hypertext Markup Language, enables you to organize the contents of a document into elements so that it can be formatted after it is received. Some elements contain attributes that allow hypertext linking to other resources in the Internet. The markings that delimit the elements and sub-elements of a HTML document follow the conventions of SGML.

The exact layout of an HTML document is specified in a SGML document type declaration. A simple example is shown here:

```
<html>
<head>
<title>User Manual</title>
</head>
<body>
<h1>User Manual</h1>
<p>An <strong>amazing</strong> range of features...
</body>
```

In the following example, the first line notifies the reader that the remainder of the file can be interpreted using HTML conventions. Then we see that the file is marked up into hierarchy of elements:

```
|
|-header
|  |- title
|
|-body
|
|  |- a header
|
|  |- a paragraph
|  - a bit of text
|  - a strong element
|    | a bit of text
|  - a bit of text
```

## Using Standard Generalized Markup Language

Standard Generalized Markup Language, or SGML, is the common name used for ISO 8879 - 1986 (also FIPS 152) an international standard for electronic document exchange. SGML is the standard upon which HTML was built.

An SGML document is a stream of characters that can be organized into a hierarchy of elements. For example most HTML documents are broken into two sub-elements a header and a body. The body might contain paragraphs. Individual paragraphs might containing elements in italic.

An SGML document can be parsed into this hierarchy of elements with the guidance of a declaration specific to a family of documents known as a document type definition, or DTD. With the help of a DTD an SGML parser can create a hierarchy of elements from the stream of characters that make up the document.

Elements in a document can have attributes in addition to sub-elements. In HTML, for example, a hypertext link is created by creating elements known as an anchor with the URL to link to as an attribute.



While HTML is the most common SGML DTD used in the World Wide Web, dozens if not hundreds of other DTD exist for purposes as widely varied as DOD specifications to musical notation.



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

**configuration:** An instance of the G2 class GW-Bridge-Configuration that specifies details of how G2 WebLink operates. You apply the information in a configuration by calling `gw-configure-bridge`.

## G

**gw-configuration-workspace:** The only top-level workspace of the module `gflow`. This workspace contains objects that you edit to configure the connection between G2 WebLink and G2, as well as the behavior of G2 WebLink.

**gw-desired-connection-configuration:** A free-text on the workspace `gw-connection-configuration` that the module `gflow` uses to configure the connection from G2 to G2 WebLink. The syntax of the free-text must conform to the syntax that you use to specify the `gsi-connection-configuration` attribute of a GSI Interface Object.

**gflow module:** A module containing class definitions, G2 procedures, rules, and other objects required for use of G2 WebLink. You must include `gflow` in the required modules of any application that uses G2 WebLink. The module is stored in the module file `gflow.kb`. The module has one top-level workspace named `gw-configuration-workspace`. You must edit the objects on this workspace to configure your G2 WebLink application.

**gw-status-of-connection-to-bridge:** A symbolic variable that is maintained by the `gflow` module and is used by code in your KB to determine the state of the connection to the bridge. It can take the following symbolic values:

```
OK  
KB-WAS-RESET  
AWAITING-GSI-CONNECTION-TO-BRIDGE  
CONNECTED-BUT-AWAITING-CONFIGURATION-OF-BRIDGE
```

Remote procedure calls to the bridge succeed only when the value `gw-status-of-connection-to-bridge` of is OK.

## R

**remote procedure calls:** Procedures in G2 that enable your G2 application to communicate with G2 WebLink.

**response procedure:** A user-defined G2 procedure that generates a resource for an HTTP client. The response procedure that generates a particular resource must be specified in the call to `gw-tender-resources` that tenders the resource.

## U

**Universal Resource Locator (URL):** A pointer to resources on the Internet. Each URL specifies the protocol used to access the resource, the name of the Internet server where the resource resides, and a path to the resource. The three following examples illustrate the format of URLs:

```
ftp://www.ai.mit.edu/  
http://tycho.usno.navy.mil/time.html  
gopher://weather.service.net/ma/boston/current-conditions
```

@ 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**

character sets, defining default  
 command-line options

-default-charset  
 -enable-hostname-lookups

configuring G2 WebLink

general  
 optional steps for

customer support services

**D**

data types used by G2 WebLink

**E**

-enable-hostname-lookups command-line  
 option

**G**

G2 WebLink

capabilities of  
 communicating with through remote  
 procedure calls  
 configuring behavior of  
 configuring connection to G2 from  
 configuring resources served by  
 data types used by  
 optional steps for configuring  
 sample applications that use  
 transcripts of operation of

gw-add-to-url-reply() remote procedure  
 syntax and arguments of

gw-configuration-workspace workspace

gw-configure-bridge() remote procedure  
 configuring connection to bridge with  
 configuring G2 WebLink with  
 syntax and arguments of

gw-decline-to-serve-url() remote procedure  
 syntax and arguments of

gwex1.kb

gwex2.kb

gwex3.kb

gw-finish-url-reply() remote procedure  
 syntax and arguments of

gw-get-attribute-of-request() remote procedure  
 syntax and arguments of

gw-initial-bridge-configuration sample GW  
 Bridge Configuration

gw-kill-bridge() remote procedure  
 syntax and arguments of

gw-make-base64-encoded() remote procedure  
 syntax and arguments of

gw-ping remote procedure

gw-ping() remote procedure  
 syntax and arguments of

gw-set-url-reply-charset remote procedure

gw-start-http-server() remote procedure  
 syntax and arguments of  
 using to establish listeners for HTTP  
 requests

gw-tender-resources() remote procedure  
 syntax and arguments of  
 using to specify resources that G2 provides  
 to Internet clients

**H**

host names, logging

HTML. *See* Hypertext Markup Language

Hypertext Markup Language (HTML)

**L**

logging host names

**M**

monitor-gsi-interface-to-gw() procedure

**R**

remote procedure calls

- gw-add-to-url-reply()
- gw-configure-bridge()
- gw-decline-to-serve-url()
- gw-finish-url-reply()
- gw-get-attribute-of-request()
- gw-kill-bridge()
- gw-make-base64-encoded()
- gw-ping
- gw-ping()
- gw-set-url-reply-charset
- gw-start-http-server()
- gw-tender-resources
- return values of
- signals returned by
- summary of
- using to communicate with G2 WebLink
- resources on Internet
- identifying with URLs

## S

- SGML. *See* Standard Generalized Markup Language
- Standard Generalized Markup Language (SGML)

## T

- transcripts of bridge operation

## U

- Unicode UTF-8, default character set
- Universal Resource Locators. *See* URLs
- URLs
  - format of
  - identifying Internet resources with