

# G2-HLA Bridge

## User's Guide

Version 2.1 Rev. 1



G2-HLA Bridge User's Guide, Version 2.1 Rev. 1

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: DOC104-220

# Contents

---

	<b>Preface</b>	<b>vii</b>
	About this Guide	vii
	Audience	vii
	Conventions	viii
	Related Documentation	ix
	Customer Support Services	xii
<b>Chapter 1</b>	<b>Introduction to the G2-HLA Bridge</b>	<b>1</b>
	Introduction	1
	Loading the G2-HLA Bridge	2
<b>Chapter 2</b>	<b>Using the G2-HLA Bridge</b>	<b>3</b>
	Introduction	3
	G2-HLA Bridge Federate Example	4
	G2-HLA Bridge Hello World Federate Example	5
<b>Chapter 3</b>	<b>G2-HLA API</b>	<b>7</b>
	Introduction	9
	GHLA Interface	10
	Request Methods	11
	ghla-interface::ghla-attribute-ownership-acquisition	13
	ghla-interface::ghla-attribute-ownership-acquisition-if-available	14
	ghla-interface::ghla-attribute-ownership-release-response	15
	ghla-interface::ghla-cancel-attribute-ownership-acquisition	16
	ghla-interface::ghla-cancel-negotiated-attribute-ownership-divestiture	17
	ghla-interface::ghla-create-and-join-federation	18
	ghla-interface::ghla-delete-object-instance	19
	ghla-interface::ghla-disable-asynchronous-delivery	20
	ghla-interface::ghla-disable-attribute-relevance-advisory-switch	21
	ghla-interface::ghla-disable-class-relevance-advisory-switch	22
	ghla-interface::ghla-disable-interaction-relevance-advisory-switch	23

ghla-interface::ghla-disable-time-constrained 24  
 ghla-interface::ghla-disable-time-regulation 25  
 ghla-interface::ghla-enable-asynchronous-delivery 26  
 ghla-interface::ghla-enable-attribute-relevance-advisory-switch 27  
 ghla-interface::ghla-enable-class-relevance-advisory-switch 28  
 ghla-interface::ghla-enable-interaction-relevance-advisory-switch 29  
 ghla-interface::ghla-enable-time-constrained 30  
 ghla-interface::ghla-enable-time-regulation 32  
 ghla-interface::ghla-flush-queue-request 34  
 ghla-interface::ghla-get-object-instance-name 35  
 ghla-interface::ghla-is-attribute-owned-by-federate 36  
 ghla-interface::ghla-modify-lookahead 37  
 ghla-interface::ghla-negotiate-attribute-ownership-divestiture 38  
 ghla-interface::ghla-next-event-request 39  
 ghla-interface::ghla-next-event-request-available 40  
 ghla-interface::ghla-publish-interaction-class 41  
 ghla-interface::ghla-publish-object-class 43  
 ghla-interface::ghla-query-attribute-ownership 45  
 ghla-interface::ghla-query-federate-time 46  
 ghla-interface::ghla-query-lbts 47  
 ghla-interface::ghla-query-lookahead 48  
 ghla-interface::ghla-query-min-next-event-time 49  
 ghla-interface::ghla-register-federation-synchronization-point 50  
 ghla-interface::ghla-register-object-instance 51  
 ghla-interface::ghla-request-class-attribute-value-update 52  
 ghla-interface::ghla-request-object-attribute-value-update 53  
 ghla-interface::ghla-resign-and-destroy-federation 54  
 ghla-interface::ghla-send-interaction 55  
 ghla-interface::ghla-subscribe-interaction-class 56  
 ghla-interface::ghla-subscribe-object-class-attributes 58  
 ghla-interface::ghla-synchronization-point-achieved 60  
 ghla-interface::ghla-time-advance-request 61  
 ghla-interface::ghla-time-advance-request-available 63  
 ghla-interface::ghla-unconditional-attribute-ownership-divestiture 65  
 ghla-interface::ghla-unpublish-interaction-class 66  
 ghla-interface::ghla-unpublish-object-class 67  
 ghla-interface::ghla-unsubscribe-interaction-class 68  
 ghla-interface::ghla-unsubscribe-object-class 69  
 ghla-interface::ghla-update-attribute-values 70

#### Callback Methods 71

ghla-interface::ghla-cb-announce-synchronization-point 72  
 ghla-interface::ghla-cb-attribute-is-not-owned 73  
 ghla-interface::ghla-cb-attribute-owned-by-rti 74  
 ghla-interface::ghla-cb-attribute-ownership-acquisition-notification 75  
 ghla-interface::ghla-cb-attribute-ownership-divestiture-notification 76  
 ghla-interface::ghla-cb-attribute-ownership-unavailable 77

ghla-interface::ghla-cb-confirm-attribute-ownership-acquisition-cancellation 78  
ghla-interface::ghla-cb-discover-object-instance 79  
ghla-interface::ghla-cb-error-handler 80  
ghla-interface::ghla-cb-federation-synchronized 81  
ghla-interface::ghla-cb-inform-attribute-ownership 82  
ghla-interface::ghla-cb-provide-attribute-value-update 83  
ghla-interface::ghla-cb-receive-interaction 84  
ghla-interface::ghla-cb-reflect-attribute-values 85  
ghla-interface::ghla-cb-remove-object-instance 86  
ghla-interface::ghla-request-attribute-ownership-assumption 87  
ghla-interface::ghla-request-attribute-ownership-release 88  
ghla-interface::ghla-start-registration-for-object-class 89  
ghla-interface::ghla-stop-registration-for-object-class 90  
ghla-interface::ghla-synchronization-point-registration-failed 91  
ghla-interface::ghla-synchronization-point-registration-succeeded 92  
ghla-interface::ghla-time-advance-grant 93  
ghla-interface::ghla-time-constrained-enabled 94  
ghla-interface::ghla-time-regulation-enabled 95  
ghla-interface::ghla-turn-interactions-off 96  
ghla-interface::ghla-turn-interactions-on 97  
ghla-interface::ghla-turn-updates-off-for-object-instance 98  
ghla-interface::ghla-turn-updates-on-for-object-instance 99

**Index 101**



# Preface

---

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

About this Guide **vii**

Audience **vii**

Conventions **viii**

Related Documentation **ix**

Customer Support Services **xii**



## About this Guide

This guide describes the G2-HLA Bridge. This module provides an interface to the High Level Architecture (HLA) 1.3.

## 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, as well as HLA.

# 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's Guide*
- *G2 System Procedures Reference Manual*

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

## **G2 Utilities**

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

## **G2 Developers' Utilities**

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

## **Bridges and External Systems**

- *G2 ActiveXLink User's Guide*
- *G2 CORBALink User's Guide*
- *G2 Database Bridge User's Guide*
- *G2-ODBC Bridge Release Notes*
- *G2-Oracle Bridge Release Notes*
- *G2-Sybase Bridge Release Notes*
- *G2 JMail Bridge User's Guide*
- *G2 Java Socket Manager User's Guide*
- *G2 JMSLink User's Guide*
- *G2 OPCLink User's Guide*
- *G2 PI Bridge User's Guide*
- *G2-SNMP Bridge User's Guide*
- *G2 CORBALink User's Guide*
- *G2 WebLink User's Guide*

## **G2 JavaLink**

- *G2 JavaLink User's Guide*
- *G2 DownloadInterfaces User's Guide*
- *G2 Bean Builder User's Guide*

## **G2 Diagnostic Assistant**

- *GDA User's 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.ignitotech.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@ignitotech.com](mailto:support@ignitotech.com)

# Introduction to the G2-HLA Bridge

---

*Describes the G2-HLA Bridge.*

Introduction 1

Loading the G2-HLA Bridge 2



## Introduction

The formal definition of the High Level Architecture (HLA) 1.3 comprises three main components: the HLA rules, the HLA interface specification, and the HLA object model template (OMT). It provides a language independent specification (LIS) and multiple language bindings to support inter-simulation communication in a distributed simulation domain. To learn more about HLA, visit the HLA web site.

The High Level Architecture (HLA) is an integrated architecture that has been developed to provide a common architecture for Modeling and Simulation (M & S). The HLA requires that inter-federate interactions use a standard API. The specification defines the standard services and interfaces to be used by the federates in order to support efficient information exchange when participating in a distributed federation execution and reuse of the individual federates.

The RTI (Run Time Infrastructure) provides services to federates in a way that is analogous to how a distributed operating system provides services to applications. These interfaces are arranged into eight basic RTI service groups:

- Federation management.
- Declaration management.
- Object management.

- Ownership management.
- Time management.
- Data Distribution management.
- Support Services.
- Management Object Model (MOM).

The eight service groups describe the interface between the federates and the RTI, and the software services provided by the RTI for use by HLA federates. The initial set of these services was carefully chosen to provide those functions most likely to be required across multiple federations. As a result, federate applications require most of the services described. The RTI requires a set of services from the federate that are referred to as “RTI callbacks methods.”

The G2-HLA Bridge provides the communication interface between the RTI and G2. It provides methods and callback methods to any G2 application. The definition and basic implementation of these methods is provided in the `ghla.kb` module. This module requires the `sys-mod.kb`, `uilroot.kb`, and `grtl.kb` modules. You should merge the `ghla.kb` module into your application and make it a required module.

Before using this package, you should familiarize yourself with HLA and RTI. You should review the documentation available from the HLA web site. In addition and since this implementation is based on the C++ libraries of the RTI, the documentation of these libraries available from the HLA web site is a good resource for additional information.

## Loading the G2-HLA Bridge

The G2-HLA Bridge runs on the Windows platform only.

Before installing the G2-HLA Bridge, you must install MÄK RTI 4.x. It contains the core technology and implementation of HLA 1.3.

Set the `MAK_RTIDIR` environment variable to point to the location of the installed library and include files, for example, `C:\MAK\makRti4.4.1` (This is usually done automatically during the installation of MÄK RTI).

# Using the G2-HLA Bridge

---

*Describes how to use the G2-HLA Bridge.*

Introduction 3

G2-HLA Bridge Federate Example 4

G2-HLA Bridge Hello World Federate Example 5



## Introduction

Before using this package, make yourself familiar with HLA. You can find information about HLA on the HLA web site. You should be familiar with the concepts of HLA and how to start the Federation executive (RtiExec). Also before attempting to run a G2 application using the `g2-hla.kb` module, ensure that the G2-HLA Bridge and the RtiExec process are running.

The G2-HLA Bridge KB module includes the `ghla-interface` interface object definition and a set of methods for this object definition. The G2-HLA Bridge defines two type of methods:

- Request methods that your application can call to send information to the RTI, whose names start with `ghla-`.
- Callback methods that are called by the G2-HLA Bridge to inform the G2 application of changes, whose names start with `ghla-cb-`.

Your G2 application should create a subclass of `ghla-interface`. You must also ensure that this interface connects to the G2-HLA Bridge process. To do this, edit the `gsi-connection-configuration` attribute of instances of your subclass. The format of the entry is `tcp-ip host bridge-host-name port-number bridge-port-number`. You should replace `bridge-host-name` with the computer name where the

bridge process is running and *bridge-port-number* with the TCP port number the bridge is using to communicate with G2. If you launch the G2-HLA Bridge, the computer name and port number are given in the bridge process window in the line `TCP_IP:bridge-host-name:bridge-port-number`.

In addition, you may create your custom methods for any method of `ghla-interface` as required by your application. If you customize federation callback methods (names start with `ghla-cb-`), your application gets informed of changes that occur in the federation.

If you customize federation request methods, ensure that your newly created method includes `call next method` statement. This will ensure your request is processed and sent to the Federate executive.

While developing your application, you will also need to create a FED file describing your simulation and modeling application. It must save it in the `config` subdirectory of your RTI installation directory. If you develop a federation called `G2Simulation` for example, your fed filename should be `G2Simulation.fed`. Please consult the HLA web pages for additional information in regards to the format and content of FED files.

The installation package of this toolkit includes two G2 based HLA Federation examples. The following sections give a brief overview of these examples; studying the G2 code of both applications may give you ideas on how to build yours.

Additional information can be found in the HLA specification documentation and RTI programmers manual. To make it easier to cross reference implementations, the G2 API names are based on the C++ RTI libraries. The G2-HLA Bridge is using the C++ RTI libraries to communicate with the RTI.

For details on the G2-HLA Bridge application programmers' interface, see [G2-HLA API](#).

## G2-HLA Bridge Federate Example

This G2 application demonstrates how to use the G2-HLA Bridge. The application is a client/server simulation package. The Federation is composed of three Federates. The G2 client Federate creates and publishes vehicles with desired initial values and relies on simulation Federate servers to update the vehicle location and fuel level at each clock tick. The client collects and displays the current location and fuel level of each vehicle and acts as a simulation controller. It also launches any required process including `Rtiexec` and the G2-HLA Bridge, and coordinates the simulation time advance activities. The application is stored in the `ghla-demo.kb` file.

The client (Federate 1) and servers communicate via HLA. Simulation servers may either be written in C++ or in G2.

This G2 KB includes a simulation server (Federate 2) written in G2. This server manages and updates the location of each vehicle. Even if the code is located in the same G2 process, the G2 server will communicate with the client through HLA and therefore have its own G2-HLA Bridge process. It has been kept in the same G2 KB for convenience.

A second server (Federate 3) is used to manage and update the fuel level of vehicles. This Federate has been written in C++ in order to demonstrate the collaboration of Federates written in different programming languages.

## G2-HLA Bridge Hello World Federate Example

This G2 application demonstrates how to use the G2-HLA Bridge. The application publishes an HLA object containing a country name and its population count. The main loop will perform a user defined number of times the following actions: generate a new population count, inform the Federation of the new count, request a simulation time advance and wait until the simulation time advance is granted. In addition once the population has been update 50 times, it will request a Federation synchronization. The main loop will wait until the Federation is synchronized before performing additional population updates.

If other Federates join this Federation and generate a population for other countries, this Federate is informed of updates. The application creates objects at run time for these countries to keep track of their population.

The G2-HLA Bridge includes a Federate written in C++ that you might use. It will join the same Federation and generate a population for a different country. You may run several instances of the program simultaneously if desired. The executable is stored in the bin directory. To start it, open a command shell, go to the bin installation directory, and enter the following command:

```
G2HelloWorld country-name initial-population
```

If you run the program several times simultaneously, ensure that the country name argument is different for each of them.

Alternatively, you could also run several times and simultaneously the G2 HelloWorld application. Again ensure that the Federate name of each application is different. Also each G2 application must connect to a separate G2-HLA Bridge process.

Before running this application, ensure that the G2-HLA Bridge and the RTI executive (RtiExec) processes are running. The G2-HLA Bridge process executable is located in the installation bin directory and the process name is `ghla`, while the RtiExec program is located in the RTI installation directory.

While running, this G2 application will post messages on the message board to describe the evolution of the execution.



# G2-HLA API

---

*Describes the G2-HLA Bridge interface object, request methods, and callback methods.*

Introduction **9**

GHLA Interface **10**

Request Methods **11**

- ghla-interface::ghla-attribute-ownership-acquisition **13**
- ghla-interface::ghla-attribute-ownership-acquisition-if-available **14**
- ghla-interface::ghla-attribute-ownership-release-response **15**
- ghla-interface::ghla-cancel-attribute-ownership-acquisition **16**
- ghla-interface::ghla-cancel-negotiated-attribute-ownership-divestiture **17**
- ghla-interface::ghla-create-and-join-federation **18**
- ghla-interface::ghla-delete-object-instance **19**
- ghla-interface::ghla-disable-asynchronous-delivery **20**
- ghla-interface::ghla-disable-attribute-relevance-advisory-switch **21**
- ghla-interface::ghla-disable-class-relevance-advisory-switch **22**
- ghla-interface::ghla-disable-interaction-relevance-advisory-switch **23**
- ghla-interface::ghla-disable-time-constrained **24**
- ghla-interface::ghla-disable-time-regulation **25**
- ghla-interface::ghla-enable-asynchronous-delivery **26**
- ghla-interface::ghla-enable-attribute-relevance-advisory-switch **27**
- ghla-interface::ghla-enable-class-relevance-advisory-switch **28**
- ghla-interface::ghla-enable-interaction-relevance-advisory-switch **29**
- ghla-interface::ghla-enable-time-constrained **30**
- ghla-interface::ghla-enable-time-regulation **32**
- ghla-interface::ghla-flush-queue-request **34**
- ghla-interface::ghla-get-object-instance-name **35**
- ghla-interface::ghla-is-attribute-owned-by-federate **36**
- ghla-interface::ghla-modify-lookahead **37**
- ghla-interface::ghla-negotiate-attribute-ownership-divestiture **38**
- ghla-interface::ghla-next-event-request **39**
- ghla-interface::ghla-next-event-request-available **40**
- ghla-interface::ghla-publish-interaction-class **41**

ghla-interface::ghla-publish-object-class 43  
ghla-interface::ghla-query-attribute-ownership 45  
ghla-interface::ghla-query-federate-time 46  
ghla-interface::ghla-query-lbts 47  
ghla-interface::ghla-query-lookahead 48  
ghla-interface::ghla-query-min-next-event-time 49  
ghla-interface::ghla-register-federation-synchronization-point 50  
ghla-interface::ghla-register-object-instance 51  
ghla-interface::ghla-request-class-attribute-value-update 52  
ghla-interface::ghla-request-object-attribute-value-update 53  
ghla-interface::ghla-resign-and-destroy-federation 54  
ghla-interface::ghla-send-interaction 55  
ghla-interface::ghla-subscribe-interaction-class 56  
ghla-interface::ghla-subscribe-object-class-attributes 58  
ghla-interface::ghla-synchronization-point-achieved 60  
ghla-interface::ghla-time-advance-request 61  
ghla-interface::ghla-time-advance-request-available 63  
ghla-interface::ghla-unconditional-attribute-ownership-divestiture 65  
ghla-interface::ghla-unpublish-interaction-class 66  
ghla-interface::ghla-unpublish-object-class 67  
ghla-interface::ghla-unsubscribe-interaction-class 68  
ghla-interface::ghla-unsubscribe-object-class 69  
ghla-interface::ghla-update-attribute-values 70

#### Callback Methods 71

ghla-interface::ghla-cb-announce-synchronization-point 72  
ghla-interface::ghla-cb-attribute-is-not-owned 73  
ghla-interface::ghla-cb-attribute-owned-by-rti 74  
ghla-interface::ghla-cb-attribute-ownership-acquisition-notification 75  
ghla-interface::ghla-cb-attribute-ownership-divestiture-notification 76  
ghla-interface::ghla-cb-attribute-ownership-unavailable 77  
ghla-interface::ghla-cb-confirm-attribute-ownership-acquisition-cancellation 78  
ghla-interface::ghla-cb-discover-object-instance 79  
ghla-interface::ghla-cb-error-handler 80  
ghla-interface::ghla-cb-federation-synchronized 81  
ghla-interface::ghla-cb-inform-attribute-ownership 82  
ghla-interface::ghla-cb-provide-attribute-value-update 83  
ghla-interface::ghla-cb-receive-interaction 84  
ghla-interface::ghla-cb-reflect-attribute-values 85  
ghla-interface::ghla-cb-remove-object-instance 86  
ghla-interface::ghla-request-attribute-ownership-assumption 87  
ghla-interface::ghla-request-attribute-ownership-release 88  
ghla-interface::ghla-start-registration-for-object-class 89  
ghla-interface::ghla-stop-registration-for-object-class 90  
ghla-interface::ghla-synchronization-point-registration-failed 91  
ghla-interface::ghla-synchronization-point-registration-succeeded 92

ghla-interface::ghla-time-advance-grant	93
ghla-interface::ghla-time-constrained-enabled	94
ghla-interface::ghla-time-regulation-enabled	95
ghla-interface::ghla-turn-interactions-off	96
ghla-interface::ghla-turn-interactions-on	97
ghla-interface::ghla-turn-updates-off-for-object-instance	98
ghla-interface::ghla-turn-updates-on-for-object-instance	99



## Introduction

This chapter describes the G2-HLA Bridge Application Programmers' Interface (API). The APIs are divided into these functional categories:

- Request methods.
- Callback methods.

# GHLA Interface

Here are the attributes of the ghla-interface:

<b>Attribute</b>	<b>Description</b>
federate-time-advance-granted	(Read-only) This attribute is set to <b>false</b> when an API is called requesting a simulation time change. It is set to <b>true</b> when the RTI granted the time advance request.  <i>Allowable values:</i> <b>truth-value</b>  <i>Default value:</i> <b>false</b>
federate-time	(Read-only) Contains the last granted time by the RTI. It is automatically update by the G2-HLA APIs when the RTI granted any time change.  <i>Allowable values:</i> <b>float</b>  <i>Default value:</i> <b>0.0</b>
debug	If set to <b>true</b> , the default G2-HLA callback methods will post inform statements when called by the RTI via the bridge.  <i>Allowable values:</i> <b>truth-value</b>  <i>Default value:</i> <b>true</b>

# Request Methods

ghla-interface::ghla-attribute-ownership-acquisition  
ghla-interface::ghla-attribute-ownership-acquisition-if-available  
ghla-interface::ghla-attribute-ownership-release-response  
ghla-interface::ghla-cancel-attribute-ownership-acquisition  
ghla-interface::ghla-cancel-negotiated-attribute-ownership-divestiture  
ghla-interface::ghla-create-and-join-federation  
ghla-interface::ghla-create-and-join-federation  
ghla-interface::ghla-disable-asynchronous-delivery  
ghla-interface::ghla-disable-attribute-relevance-advisory-switch  
ghla-interface::ghla-disable-class-relevance-advisory-switch  
ghla-interface::ghla-disable-interaction-relevance-advisory-switch  
ghla-interface::ghla-disable-time-constrained  
ghla-interface::ghla-disable-time-regulation  
ghla-interface::ghla-enable-asynchronous-delivery  
ghla-interface::ghla-enable-attribute-relevance-advisory-switch  
ghla-interface::ghla-enable-class-relevance-advisory-switch  
ghla-interface::ghla-enable-interaction-relevance-advisory-switch  
ghla-interface::ghla-enable-time-constrained  
ghla-interface::ghla-enable-time-regulation  
ghla-interface::ghla-flush-queue-request  
ghla-interface::ghla-get-object-instance-name  
ghla-interface::ghla-is-attribute-owned-by-federate  
ghla-interface::ghla-modify-lookahead  
ghla-interface::ghla-negotiate-attribute-ownership-divestiture  
ghla-interface::ghla-next-event-request  
ghla-interface::ghla-next-event-request-available  
ghla-interface::ghla-publish-interaction-class  
ghla-interface::ghla-publish-object-class  
ghla-interface::ghla-query-attribute-ownership

ghla-interface::ghla-query-federate-time  
ghla-interface::ghla-query-lbts  
ghla-interface::ghla-query-lookahead  
ghla-interface::ghla-query-min-next-event-time  
ghla-interface::ghla-register-federation-synchronization-point  
ghla-interface::ghla-register-object-instance  
ghla-interface::ghla-request-class-attribute-value-update  
ghla-interface::ghla-request-object-attribute-value-update  
ghla-interface::ghla-resign-and-destroy-federation  
ghla-interface::ghla-send-interaction  
ghla-interface::ghla-subscribe-interaction-class  
ghla-interface::ghla-subscribe-object-class-attributes  
ghla-interface::ghla-synchronization-point-achieved  
ghla-interface::ghla-time-advance-request  
ghla-interface::ghla-time-advance-request-available  
ghla-interface::ghla-unconditional-attribute-ownership-divestiture  
ghla-interface::ghla-unpublish-interaction-class  
ghla-interface::ghla-unpublish-object-class  
ghla-interface::ghla-unsubscribe-interaction-class  
ghla-interface::ghla-unsubscribe-object-class  
ghla-interface::ghla-update-attribute-values

# ghla-interface::ghla-attribute-ownership-acquisition

Implements the “attribute ownership acquisition” service as specified by HLA.

## Synopsis

```
ghla-interface::ghla-attribute-ownership-acquisition
(hla-interface: ghla-interface, hla-object-handle: float,
 hla-attribute-names: symbol-array, tag: text )
-> error-code: symbol, error-message: text
```

Argument	Description
<i>hla-interface</i>	The HLA interface connection to use for the request.
<i>hla-object-handle</i>	The HLA object handle for which to change the attribute ownership.
<i>hla-attribute-names</i>	The list of attribute names to take ownership with HLA.
<i>tag</i>	The user application-specific tag.
Return Value	Description
<u><i>error-code</i></u>	The error code or NO-ERROR if no error occurred.
<u><i>error-message</i></u>	A message describing the error.

# ghla-interface::ghla-attribute-ownership-acquisition-if-available

Use by a federate to attempt to take ownership of attributes of objects by implementing the “attribute ownership acquisition” service as specified by HLA.

## Synopsis

ghla-interface::ghla-attribute-ownership-acquisition-if-available

(*ghla-interface*: ghla-interface, *ghla-object-handle*: float,  
*ghla-attribute-names*: symbol-array)

-> *error-code*: symbol, *error-message*: text

Argument	Description
<i>ghla-interface</i>	The HLA interface connection to use for the request.
<i>ghla-object-handle</i>	The HLA object handle for which to change the attribute ownership.
<i>ghla-attribute-names</i>	The list of attribute names to take ownership with HLA.

Return Value	Description
<i>error-code</i>	The error code or NO-ERROR if no error occurred.
<i>error-message</i>	A message describing the error.

## ghla-interface::ghla-attribute-ownership-release-response

Implements the “attribute ownership release response” service as specified by HLA.

### Synopsis

ghla-interface::ghla-attribute-ownership-release-response  
 (*ghla-interface*: ghla-interface, *ghla-object-handle*: float,  
*ghla-attribute-names*: symbol-array )  
 -> *error-code*: symbol, *error-message*: text, *attribute-list*: symbol-array

Argument	Description
<i>ghla-interface</i>	The HLA interface connection to use for the request.
<i>ghla-object-handle</i>	The HLA object handle for which to change the attribute ownership.
<i>ghla-attribute-names</i>	The list of attribute names to take ownership with HLA.
Return Value	Description
<i>error-code</i>	The error code or NO-ERROR if no error occurred.
<i>error-message</i>	A message describing the error.
<i>attribute-list</i>	The attribute list returned by the RTI API.

# ghla-interface::ghla-cancel-attribute-ownership-acquisition

Implements the “cancel attribute ownership acquisition” service as specified by HLA.

## Synopsis

ghla-interface::ghla-cancel-attribute-ownership-acquisition  
(*hla-interface*: ghla-interface, *hla-object-handle*: float,  
*hla-attribute-names*: symbol-array)  
-> *error-code*: symbol, *error-message*: text

Argument	Description
<i>hla-interface</i>	The HLA interface connection to use for the request.
<i>hla-object-handle</i>	The HLA object handle for which to change the attribute ownership.
<i>hla-attribute-names</i>	The list of attribute names to take ownership with HLA.

Return Value	Description
<i>error-code</i>	The error code or NO-ERROR if no error occurred.
<i>error-message</i>	A message describing the error.

# ghla-interface::ghla-cancel-negotiated-attribute-ownership-divestiture

Implements the “cancel negotiate attribute ownership divestiture” service as specified by HLA.

## Synopsis

ghla-interface::ghla-cancel-negotiated-attribute-ownership-divestiture  
 (*hla-interface*: ghla-interface, *hla-object-handle*: float,  
*hla-attribute-names*: symbol-array)  
 -> *error-code*: symbol, *error-message*: text

Argument	Description
<i>hla-interface</i>	The HLA interface connection to use for the request.
<i>hla-object-handle</i>	The HLA object handle for which to change the attribute ownership.
<i>hla-attribute-names</i>	The list of attribute names to take ownership with HLA.
Return Value	Description
<i>error-code</i>	The error code or NO-ERROR if no error occurred.
<i>error-message</i>	A message describing the error.

# ghla-interface::ghla-create-and-join-federation

Creates a named Federation (FedExec) and registers it with the RTI executive (RtiExec). In addition, this method requests permission to participate in the named Federation. Upon a successful return from this method, your G2 application joined the Federation. Your G2 application may interact with the Federation via the methods of the *hla-interface* argument. Your G2 application will receive update information from the Federation through callback methods of the same *hla-interface* argument you specified. Callback methods automatically called by the RTI start with the name *ghla-cb-*. You should not call them, but you may write custom methods in order to process the information forwarded by the RTI. If the named Federation already exists when calling this method, the G2-HLA Bridge does not consider it an error and will join the already existing Federation. The Federate name has to be unique for each application joining a Federation. If the G2 application is joining the same Federation through several HLA interfaces, you must ensure that the Federate name for each is unique.

## Synopsis

ghla-interface::ghla-create-and-join-federation

(*hla-interface*: ghla-interface, *federation-name*: text, *federate-name*: text)

-> *error-code*: symbol, *error-message*: text

Argument	Description
<i>hla-interface</i>	The HLA interface connection to use for the request.
<i>federation-name</i>	The Federation name to create and join.
<i>federate-name</i>	The Federate name of the application joining the Federation.

Return Value	Description
<u><i>error-code</i></u>	The error code or NO-ERROR if no error occurred.
<u><i>error-message</i></u>	A message describing the error.

## ghla-interface::ghla-delete-object-instance

Informs the Federation that an HLA object should be removed.

### Synopsis

ghla-interface::ghla-delete-object-instance

(*hla-interface*: ghla-interface, *hla-object-handle*: float, *hla-time*: float, *tag*: text)

-> *error-code*: symbol, *error-message*: text, *retraction-handle*: float

Argument	Description
<i>hla-interface</i>	The HLA interface connection to use for the request.
<i>hla-object-handle</i>	The HLA object handle of the object to delete.
<i>hla-time</i>	The Federate time.
<i>tag</i>	An application-specific tag. The tag is sent to all Federate receiving this message.

  

Return Value	Description
<u><i>error-code</i></u>	The error code or NO-ERROR if no error occurred.
<u><i>error-message</i></u>	A message describing the error.
<u><i>retraction-handle</i></u>	The retraction handle for this call.

# ghla-interface::ghla-disable-asynchronous-delivery

Disables the delivery of receive-ordered events to the federate in the absence of a time-advancement service. This only applies to a time-constrained Federate; if a Federate is not time-constrained, events may be delivered during any invocation of the tick() service. The invocation of tick() is implemented in the bridge and is transparent to the G2 application. Asynchronous delivery is disabled by default for a Federate, so this service should only be used to undo the effects of an hla-enable-asynchronous-delivery method invocation.

## Synopsis

ghla-interface::ghla-disable-asynchronous-delivery

(*hla-interface*: ghla-interface)

-> error-code: symbol, error-message: text

Argument	Description
<i>hla-interface</i>	The HLA interface connection to use for the request.

Return Value	Description
<u>error-code</u>	The error code or NO-ERROR if no error occurred.
<u>error-message</u>	A message describing the error.

# ghla-interface::ghla-disable-attribute-relevance-advisory-switch

Instructs the RTI to stop notifying the federate of changes in attribute instance update relevance. Relevance advisory is disabled by default.

## Synopsis

ghla-interface::ghla-disable-attribute-relevance-advisory-switch  
 (*hla-interface*: ghla-interface)  
 -> *error-code*: symbol, *error-message*: text

Argument	Description
<i>hla-interface</i>	The HLA interface connection to use for the request.
Return Value	Description
<i>error-code</i>	The error code or NO-ERROR if no error occurred.
<i>error-message</i>	A message describing the error.

# ghla-interface::ghla-disable-class-relevance-advisory-switch

Instructs the RTI to disable the class relevance advisory switch. For additional information, see [ghla-interface::ghla-enable-class-relevance-advisory-switch](#).

## Synopsis

ghla-interface::ghla-disable-class-relevance-advisory-switch

(*hla-interface*: ghla-interface)

-> *error-code*: symbol, *error-message*: text

Argument	Description
<i>hla-interface</i>	The HLA interface connection to use for the request.

Return Value	Description
<i>error-code</i>	The error code or NO-ERROR if no error occurred.
<i>error-message</i>	A message describing the error.

# ghla-interface::ghla-disable-interaction-relevance-advisory-switch

Instructs the RTI to disable the interaction relevance advisory switch. For additional information, see [ghla-interface::ghla-enable-interaction-relevance-advisory-switch](#).

## Synopsis

ghla-interface::ghla-disable-interaction-relevance-advisory-switch  
 (*hla-interface*: ghla-interface)  
 -> *error-code*: symbol, *error-message*: text

Argument	Description
<i>hla-interface</i>	The HLA interface connection to use for the request.

Return Value	Description
<i>error-code</i>	The error code or NO-ERROR if no error occurred.
<i>error-message</i>	A message describing the error.

# ghla-interface::ghla-disable-time-constrained

Instructs the LRC not to constrain the advancement of the federate's time based on the federation's time. Subsequent invocations of time-advancement methods such as `ghla-next-event-request` and `ghla-time-advance-request` will trivially succeed, as there are no longer any time-stamp-ordered events in the Federation from the perspective of a non-time-constrained federate. (That is, for any arbitrary point on the federation time axis, it can be guaranteed that no time-stamp-ordered events will be delivered with an earlier time stamp.) Any events subsequently received by the LRC of a non-time-constrained federate will be queued for receive-order delivery to the Federate, regardless of the time-ordering policy associated with the event by its sender. Events may be automatically delivered to the federate, obviating the need to invoke time-advancement services. If there are any events remaining in the time-stamp-ordered queue at the point time-constraint is disabled, they will be delivered to the Federate, in ascending order by time stamp, before any receive-ordered events are delivered. Time-constraint is disabled by default for a federate. This service should only be used after time-constraint has been enabled using `enableTimeConstrained()` and has been achieved, as indicated by a `ghla-cb-time-constrained-enabled` callback method.

## Synopsis

`ghla-interface::ghla-disable-time-constrained`  
(*hla-interface*: `ghla-interface`)  
-> *error-code*: symbol, *error-message*: text

Argument	Description
<i>hla-interface</i>	The HLA interface connection to use for the request.

  

Return Value	Description
<u><i>error-code</i></u>	The error code or NO-ERROR if no error occurred.
<u><i>error-message</i></u>	A message describing the error.

## ghla-interface::ghla-disable-time-regulation

Time-regulation is disabled by default for a Federate. This method should only be used after time-regulation has been enabled using the `ghla-enable-time-regulation` method and has been achieved, as indicated by the `ghla-cb-time-regulation-enabled` callback method.

### Synopsis

```
ghla-interface::ghla-disable-time-regulation
(hla-interface: ghla-interface)
-> error-code: symbol, error-message: text
```

Argument	Description
<i>hla-interface</i>	The HLA interface connection to use for the request.
Return Value	Description
<i>error-code</i>	The error code or NO-ERROR if no error occurred.
<i>error-message</i>	A message describing the error.

## ghla-interface::ghla-enable-asynchronous-delivery

Instructs the LRC to deliver receive-ordered events to the Federate in the absence of an in-progress time-advancement service. Subsequent to invoking this service, receive-ordered events may be delivered to the Federate during any invocation of tick(). The invocation of tick() is implemented in the bridge and is transparent to the G2 application. This setting is only relevant for federates that are time-constrained; all events are always delivered asynchronously to non-time-constrained federates. The asynchronous delivery of receive-ordered events may be subsequently disabled using the ghla-disable-asynchronous-delivery method.

### Synopsis

```
ghla-interface::ghla-enable-asynchronous-delivery  
(hla-interface: ghla-interface)  
-> error-code: symbol, error-message: text
```

Argument	Description
<i>hla-interface</i>	The HLA interface connection to use for the request.

  

Return Value	Description
<u>error-code</u>	The error code or NO-ERROR if no error occurred.
<u>error-message</u>	A message describing the error.

## ghla-interface::ghla-enable-attribute-relevance-advisory-switch

Instructs the RTI to begin notifying the Federate of changes in attribute instance update relevance. The RTI does this by calling the `ghla-cb-turn-updates-on-for-object-instance` and `ghla-cb-turn-updates-off-for-object-instance` callback methods. By default, relevance advisory is disabled.

### Synopsis

ghla-interface::ghla-enable-attribute-relevance-advisory-switch  
 (*hla-interface*: ghla-interface)  
 -> *error-code*: symbol, *error-message*: text

Argument	Description
<i>hla-interface</i>	The HLA interface connection to use for the request.
Return Value	Description
<i>error-code</i>	The error code or NO-ERROR if no error occurred.
<i>error-message</i>	A message describing the error.

# ghla-interface::ghla-enable-class-relevance-advisory-switch

Instructs the RTI to begin notifying the Federate of changes in object class registration relevance. The RTI does this by calling the `ghla-cb-start-registration-for-object-class` and `ghla-cb-stop-registration-for-object-class` callback methods. Registration of a class is considered relevant if at least one other Federate has subscribed to at least one attribute of a HLA class published by your G2 Federate. By default, the relevance advisory switch is enabled.

## Synopsis

ghla-interface::ghla-enable-class-relevance-advisory-switch

(*hla-interface*: ghla-interface)

-> error-code: symbol, error-message: text

Argument	Description
<i>hla-interface</i>	The HLA interface connection to use for the request.

Return Value	Description
<u>error-code</u>	The error code or NO-ERROR if no error occurred.
<u>error-message</u>	A message describing the error.

## ghla-interface::ghla-enable-interaction-relevance-advisory-switch

Instructs the RTI to enable the interaction relevance advisory switch. It instructs the LRC to begin notifying the Federate of changing interaction-class relevance, using the `ghla-cb-turn-interactions-on` and `ghla-cb-turn-interactions-off` callback methods. Generation of an interaction class is considered to be relevant for a Federate when at least one remote Federate has subscribed to the interaction class. This includes subscriptions to more general superclasses of the specified interaction class. Relevance advising is enabled, by default, so this service is only required when re-enabling relevance advising after a `ghla-disable-interaction-relevance-advisory-switch` invocation. The Federate will not be retroactively notified of changes in relevance that occurred while relevance advising was disabled. Invoking this service while interaction-class relevance advising is already enabled has no effect.

### Synopsis

ghla-interface::ghla-enable-interaction-relevance-advisory-switch

(*hla-interface*: ghla-interface)

-> *error-code*: symbol, *error-message*: text

Argument	Description
<i>hla-interface</i>	The HLA interface connection to use for the request.
Return Value	Description
<i>error-code</i>	The error code or NO-ERROR if no error occurred.
<i>error-message</i>	A message describing the error.

## ghla-interface::ghla-enable-time-constrained

Instructs the LRC to deliver time-stamp-ordered events to the Federate in non-decreasing order according to their associated time stamp. Events occurring simultaneously in logical time are subject to an arbitrary order of delivery. Types of events that are potentially subject to time-stamp ordering are updates, interactions, object deletions, and federation saves. For a particular event to be subject to time-stamp ordering, the following criteria must be met:

- For all events except saves, the sender must be time regulating at the point in execution at which the event is generated.
- For updates and interactions, the ordering policy (as established by the FED file and subsequent invocations of `changeAttributeOrderType()` and `changeInteractionOrderType()`, respectively) in effect for the relevant instance-attributes or interaction class at the sender must be a time-stamp-ordered policy.
- For deletions, the ordering policy (as established by the FED file and subsequent invocations of `changeAttributeOrderType()`) in effect for some instance-attribute of the object at the sender must be a time-stamp-ordered policy.
- The receiver must be time-constrained at the point in execution at which the event is received by the LRC and the point in execution at which it delivered to the federate (though not necessary at all subsequent points in-between.)
- A time-stamp argument must be provided to the service invocation resulting in the federation event. The federate will be notified through its `ghla-cb-time-constrained-enabled` callback method when time constraint has taken effect. This callback occurs asynchronously to the `ghla-enable-time-constrained` method invocation. Time-stamp-ordered events will only be delivered to a time-constrained federate when a time-advancement service (e.g., `ghla-time-advance-request` or `ghla-next-event-request`) is in progress. A time-stamp-ordered event will not be delivered until the LRC can guarantee that no events will be received with an earlier time-stamp than the event to be delivered. The `ghla-cb-time-advance-grant` callback method informs the federate that all events with time-stamps less than or equal to (or strictly less than, depending on the time-advancement service used) a specified time have been delivered to the Federate. No time-stamp ordered events will be delivered to a time-constrained federate unless a time-advancement service is in progress. No receive-ordered events will be delivered to a time-constrained federate unless a time-advancement service is in progress or the federate has enabled asynchronous delivery of receive ordered events, using the `ghla-enable-asynchronous-delivery` method.

## Synopsis

ghla-interface::ghla-enable-time-constrained

(*hla-interface*: ghla-interface)

-> *error-code*: symbol, *error-message*: text

Argument	Description
<i>hla-interface</i>	The HLA interface connection to use for the request.

Return Value	Description
<u><i>error-code</i></u>	The error code or NO-ERROR if no error occurred.
<u><i>error-message</i></u>	A message describing the error.

## ghla-interface::ghla-enable-time-regulation

Instructs the Federation to consider the federate's logical time in controlling the advancement of federation time. The specified lookahead represents the length of the logical-time interval extending forward from the federate's logical time at any given point in execution in which the Federate will not generate events. That is, the minimum permissible time-stamp for a time-stamp-ordered event generated by a time regulating Federate is the Federate's current logical time plus its lookahead. The sum of a Federate's logical time and its lookahead may be referred to as the effective logical time of the Federate. The Federate's logical time upon enabling time regulation is the minimum of the specified effective Federate logical time and the current lower-bound time-stamp of the federation. If the specified lookahead is less than the Federate's current lookahead, the specified lookahead will be phased in such that the effective logical time of the federate remains strictly non-decreasing. The new lookahead will completely take effect when the Federate has advanced its logical time from the time at which regulation was enabled by an amount exceeding the difference between the new and old lookahead values. Enabling time regulation allows the Federate to generate events that are subject to time-stamp-ordering. While a Federate is time regulating, events sent by the Federate may be designated as time-stamp ordered, provided:

- A time-stamp ordering policy is in effect for the relevant interaction class or instance-attributes, as defined by the FED file.
- A time-stamp argument is provided to the service invocation generating the event. The federate is notified through its `ghla-cb-time-regulation-enabled` callback when time constraint has taken effect. This callback occurs asynchronously to the `ghla-enable-time-regulation` method invocation.

### Synopsis

`ghla-interface::ghla-enable-time-regulation`

(*hla-interface*: `ghla-interface`, *federate-time*: float , *lookahead-time*: float)

-> *error-code*: symbol, *error-message*: text

Argument	Description
<i>hla-interface</i>	The HLA interface connection to use for the request.
<i>federate-time</i>	The Federate time.
<i>lookahead-time</i>	The Federate lookahead time.

<b>Return Value</b>	<b>Description</b>
<u><i>error-code</i></u>	The error code or NO-ERROR if no error occurred.
<u><i>error-message</i></u>	A message describing the error.

## ghla-interface::ghla-flush-queue-request

Designates all events currently in the Federate's event queue as eligible for presentation to the Federate. Subsequent invocations of tick() will first process any receive-ordered events that have arrived, then will process events in the TSO queue without regard for the federation lower-bound time stamp. For any given invocation of tick() (this call is implemented in the G2-HLA Bridge), the earliest available TSO event is processed; however, RTI may not be able to guarantee that TSO events with a lower time stamp will not arrive in the future. A time advance is granted when the federate has processed all TSO events that were queued at the time of the request. The grant time is the minimum of the minimum-next-event time and the specified cutoff time. Note that this is trivial in the case of a non-time-constrained Federate, which by definition has no events in its TSO queue; in this case, a grant to the specified cutoff time will be made upon the next invocation of tick() (this call is implemented in the G2-HLA Bridge). If time-stamp-ordered events arrive while a flush is in progress, they will be delivered in order with respect to the events remaining in the queue. Late-arriving events may replace events with a later time-stamp in the set of events affected by the flush. That is, a ghla-flush-queue-request invocation will flush a number of time-stamp-ordered events equal to the number of events queued for time-stamp-ordered delivery at the time of the service invocation. However, the actual events delivered as a result of the flush may not be the same events that were in the queue at the time of the ghla-flush-queue-request method invocation.

### Synopsis

ghla-interface::ghla-flush-queue-request  
(*hla-interface*: ghla-interface, *the-time*: float)  
-> *error-code*: symbol, *error-message*: text

Argument	Description
<i>hla-interface</i>	The HLA interface connection to use for the request.
<i>the-time</i>	The time-stamp representing the point on the federation time axis to which to advance the federate's logical time.

  

Return Value	Description
<i>error-code</i>	The error code or NO-ERROR if no error occurred.
<i>error-message</i>	A message describing the error.

## ghla-interface::ghla-get-object-instance-name

Returns the HLA name of an HLA object given its HLA handle.

### Synopsis

ghla-interface::ghla-get-object-instance-name  
 (*hla-interface*: ghla-interface, *hla-object-handle*: float)  
 -> *error-code*: symbol, *error-message*: text, *hla-object-name*: text

Argument	Description
<i>hla-interface</i>	The HLA interface connection to use for the request.
<i>hla-object-handle</i>	The HLA handle of the HLA object.

  

Return Value	Description
<u><i>error-code</i></u>	The error code or NO-ERROR if no error occurred.
<u><i>error-message</i></u>	A message describing the error.
<u><i>hla-object-name</i></u>	The HLA name of the object.

# ghla-interface::ghla-is-attribute-owned-by-federate

Implements the “attribute ownership by federate” service as specified by HLA.

## Synopsis

ghla-interface::ghla-is-attribute-owned-by-federate  
(*hla-interface*: ghla-interface, *hla-object-handle*: float,  
*hla-attribute-name*: symbol)  
-> *error-code*: symbol, *error-message*: text, *result*: truth-value

Argument	Description
<i>hla-interface</i>	The HLA interface connection to use for the request.
<i>hla-object-handle</i>	The HLA handle of the HLA object.
<i>hla-attribute-name</i>	The attribute name to check.

Return Value	Description
<u><i>error-code</i></u>	The error code or NO-ERROR if no error occurred.
<u><i>error-message</i></u>	A message describing the error.
<u><i>result</i></u>	True if the Federate owns the attribute, false otherwise.

## ghla-interface::ghla-modify-lookahead

Specifies a new lookahead value for the local Federate. The lookahead represents the length of the logical-time interval extending forward from the Federate's logical time at any given point in execution in which the Federate will not generate events. That is, the minimum permissible time-stamp for a time-stamp-ordered event generated by a time regulating federate is the Federate's current logical time plus its lookahead. The sum of a federate's logical time and its lookahead may be referred to as the effective logical time of the Federate. If the specified lookahead is less than the Federate's current lookahead, the specified lookahead will be phased in such that the effective logical time of the Federate remains strictly non-decreasing. The new lookahead will completely take effect when the Federate has advanced its logical time from the time at which regulation was enabled by an amount exceeding the difference between the new and old lookahead values. The lookahead setting is only meaningful for a time regulating federate: non-time-regulating federates may always generate events with any time stamp.

### Synopsis

ghla-interface::ghla-modify-lookahead

(*hla-interface*: ghla-interface, *lookahead-time*: float)

-> *error-code*: symbol, *error-message*: text

Argument	Description
<i>hla-interface</i>	The HLA interface connection to use for the request.
<i>lookahead-time</i>	The new lookahead time.

Return Value	Description
<i>error-code</i>	The error code or NO-ERROR if no error occurred.
<i>error-message</i>	A message describing the error.

# ghla-interface::ghla-negotiate-attribute-ownership-divestiture

Implements the “negotiate attribute ownership divestiture” service as specified by HLA.

## Synopsis

ghla-interface::ghla-negotiate-attribute-ownership-divestiture  
(*hla-interface*: ghla-interface, *hla-object-handle*: float,  
*hla-attribute-names*: symbol-array, *tag*: text)  
-> *error-code*: symbol, *error-message*: text

Argument	Description
<i>hla-interface</i>	The HLA interface connection to use for the request.
<i>hla-object-handle</i>	The HLA object handle.
<i>hla-attribute-names</i>	The list of attribute names to take ownership with HLA.
<i>tag</i>	The application-specific tag.

Return Value	Description
<u><i>error-code</i></u>	The error code or NO-ERROR if no error occurred.
<u><i>error-message</i></u>	A message describing the error.

## ghla-interface::ghla-next-event-request

Allows the Federate to advance in time to the time stamp of the next time-stamp-ordered event occurring in the federation that matches the Federate's current subscription interests. A `ghla-cb-time-advance-grant` callback method occurs after one or more time-stamp-ordered events have been delivered or the federation lower-bound time stamp (LBTS) advances past the specified cutoff time. In the presence of an event, all relevant events with the same time-stamp are delivered before the grant. The grant time is equal to the time stamp of the event(s) delivered to the Federate. In the absence of a relevant event, the grant time is the specified cutoff time. Any number of receive-ordered events may be delivered while the `ghla-next-event-request` is in progress. Note that if the Federate is not time-constrained, the grant criteria are trivially met (i.e., the effective federation LBTS for a non-constrained Federate is always infinity), so a `ghla-cb-time-advance-grant` to the cutoff time is immediately scheduled for delivery.

### Synopsis

`ghla-interface::ghla-next-event-request`

(*hla-interface*: `ghla-interface`, *the-time*: float)  
 -> *error-code*: symbol, *error-message*: text

Argument	Description
<i>hla-interface</i>	The HLA interface connection to use for the request.
<i>the-time</i>	The time stamp representing the point on the federation time axis to which to advance the Federate's logical time.
Return Value	Description
<i>error-code</i>	The error code or NO-ERROR if no error occurred.
<i>error-message</i>	A message describing the error.

## ghla-interface::ghla-next-event-request-available

Allows the Federate to advance in time to the time stamp of the next time-stamp-ordered event occurring in the federation that matches the federate's current subscription interests. A `ghla-cb-time-advance-grant` callback method occurs after one or more time-stamp-ordered events have been delivered or the federation lower-bound time stamp (LBTS) advances past the specified cutoff time. In the presence of an event, all relevant events with the same time-stamp that have been received by the LRC will be delivered before the grant. The grant time is equal to the time-stamp of the event(s) delivered to the federate. It is possible that other time-stamp-ordered events with a time stamp equal to the grant time will be delivered to the federate during a subsequent time-advancement service. In the absence of a relevant event, the grant time is the specified cutoff time. Any number of receive-ordered events may be delivered while the `ghla-next-event-request-available` is in progress. Note that if the federate is not time-constrained, the grant criteria are trivially met (i.e., the effective federation LBTS for a non-constrained federate is always infinity), so a `ghla-cb-time-advance-grant` to the cutoff time will be immediately scheduled for delivery.

### Synopsis

`ghla-interface::ghla-next-event-request-available`  
(*hla-interface*: `ghla-interface`, *the-time*: float)  
-> *error-code*: symbol, *error-message*: text

Argument	Description
<i>hla-interface</i>	The HLA interface connection to use for the request.
<i>the-time</i>	The time stamp representing the point on the federation time axis to which to advance the federate's logical time.

  

Return Value	Description
<i>error-code</i>	The error code or NO-ERROR if no error occurred.
<i>error-message</i>	A message describing the error.

## ghla-interface::ghla-publish-interaction-class

Publishes an HLA interaction class. The G2-HLA Bridge supports the following simple attribute types: integer, float, truth-value, symbol, and text. The current version does not support arrays, lists, parameters, variables, or embedded objects. While sending and receiving values, the G2-HLA Bridge uses the G2 object attribute type to determine how to format (when sending) or decode (when receiving) a value. It translates data types, as follows:

- A G2 integer attribute value to a 4 byte big endian binary and signed integer value.
- A G2 float attribute value to a 8 byte big endian floating point value using the IEEE 64-bit notation.
- A G2 truth-value attribute value to a 4 byte big endian binary and signed integer value.
- A G2 symbol attribute value to a zero terminated ASCII string value.
- A G2 text attribute value to a zero terminated ASCII string value.

### Synopsis

ghla-interface::ghla-publish-interaction-class

(*hla-interface*: ghla-interface, *hla-class-name*: text, *g2-master-object*: item, *hla-attribute-names*: symbol-array)

-> *error-code*: symbol, *error-message*: text, *class-handle*: float, *published-attributes*: float-array

Argument	Description
<i>hla-interface</i>	The HLA interface connection to use for the request.
<i>hla-class-name</i> :	The HLA name of the class to publish.
<i>g2-master-object</i>	A G2 object corresponding to the published class. Objects of this type should be passed when updating attribute values.
<i>hla-attribute-names</i>	The list of attribute names to publish with HLA. Therefore, your G2 objects may contain both public and private attributes. The HLA attribute name and the G2 attribute name must be identical.

<b>Return Value</b>	<b>Description</b>
<u><i>error-code</i></u>	The error code or NO-ERROR if no error occurred.
<u><i>error-message</i></u>	A message describing the error.
<u><i>class-handle</i></u>	The HLA handle for the published class.
<u><i>published-attributes</i></u>	The HLA handles for each of the published attributes, in the same order as in <i>hla-attribute-names</i> .

## ghla-interface::ghla-publish-object-class

Publishes an HLA object class. Any Federate publishing HLA classes may create objects of this class, register them with the Federation and provide attribute value updates. Different Federates may publish the same HLA class but manage a different set of attributes.

The G2-HLA Bridge supports the following simple attribute types: integer, float, truth-value, symbol, and text. The current version does not support arrays, lists, parameters, variables, or embedded objects. While sending and receiving values, the G2-HLA Bridge uses the G2 object attribute type to determine how to format (when sending) or decode (when receiving) a value. It translates:

- A G2 integer attribute value to a 4 byte big endian binary and signed integer value.
- A G2 float attribute value to a 8 byte big endian floating point value using the IEEE 64-bit notation.
- A G2 truth-value attribute value to a 4 byte big endian binary and signed integer value.
- A G2 symbol attribute value to a zero terminated ASCII string value.
- A G2 text attribute value to a zero terminated ASCII string value.

### Synopsis

ghla-interface::ghla-publish-object-class

(*hla-interface*: ghla-interface, *hla-class-name*: text, *g2-master-object*: item,  
*hla-attribute-names*: symbol-array)

-> *error-code*: symbol, *error-message*: text, *class-handle*: float,  
*published-attributes*: float-array

Argument	Description
<i>hla-interface</i>	The HLA interface connection to use for the request.
<i>hla-class-name</i>	The HLA name of the class to publish.

<b>Argument</b>	<b>Description</b>
<i>g2-master-object</i>	A G2 object corresponding to the published class. Objects of this type should be passed when updating attribute values.
<i>hla-attribute-names</i>	The list of attribute names to publish with HLA. Therefore, your G2 objects may contain both public and private attributes. The HLA attribute name and the G2 attribute name must be identical.

<b>Return Value</b>	<b>Description</b>
<u><i>error-code</i></u>	The error code or NO-ERROR if no error occurred.
<u><i>error-message</i></u>	A message describing the error.
<u><i>class-handle</i></u>	The HLA handle for the published class.
<u><i>published-attributes</i></u>	The HLA handles for each of the published attributes, in the same order as in <i>hla-attribute-names</i> .

# ghla-interface::ghla-query-attribute-ownership

Implements the “query attribute ownership” service as specified by HLA.

## Synopsis

```
ghla-interface::ghla-query-attribute-ownership
  (hla-interface: ghla-interface, hla-object-handle: float,
   hla-attribute-name: symbol)
-> error-code: symbol, error-message: text
```

Argument	Description
<i>hla-interface</i>	The HLA interface connection to use for the request.
<i>hla-class-name</i>	The HLA object handle for which to change the attribute ownership.
<i>hla-attribute-name</i>	The attribute name to check.

  

Return Value	Description
<u><i>error-code</i></u>	The error code or NO-ERROR if no error occurred.
<u><i>error-message</i></u>	A message describing the error.

## ghla-interface::ghla-query-federate-time

Queries the current federate time. If time-regulation is enabled, this value (plus the federate's lookahead) represents the minimum time-stamp that may be associated with time-stamp-ordered events generated subsequently by the federate. If a `ghla-time-advance-request` (or `ghla-time-advance-request-available`) method is in progress, the federate's logical time is the federation time subject of the advancement request. If a `ghla-next-event-request` (or `ghla-next-event-request-available`) method is in progress, the federate's logical time may be any value that is less than the current federation lower-bound time-stamp. If a `ghla-flush-queue-request` method is in progress, or if no time-advancement service is in progress, the federate's logical time is the time associated with the last `ghla-cb-time-advance-grant` callback method.

### Synopsis

`ghla-interface::ghla-query-federate-time`

(*hla-interface*: ghla-interface)

-> *error-code*: symbol, *error-message*: text, *federate-time*: float

Argument	Description
<i>hla-interface</i>	The HLA interface connection to use for the request.

Return Value	Description
<i>error-code</i>	The error code or NO-ERROR if no error occurred.
<i>error-message</i>	A message describing the error.
<i>federate-time</i>	The current federate time.

## ghla-interface::ghla-query-lbts

Queries the current federation lower-bound timestamp (LBTS). The LBTS is the greatest time stamp such that it can be guaranteed that no time-stamp-ordered events will be subsequently generated in the federation with a lesser time-stamp. For a time-constrained federate, the effective LBTS is the minimum of the most recently reported effective logical times (i.e., federate logical time plus federate lookahead) for all time-regulating federates in the federation. For a non-time-constrained federate, the effective LBTS is infinity; there are no time-stamp-ordered events in the federation from the perspective of a non-time-constrained federate. Note that events with time stamps earlier than the LBTS may still be queued for time-stamp-ordered delivery to a federate; the LBTS merely indicates that no time-stamp-ordered events will be subsequently generated with an earlier time stamp. A federate may use the `ghla-query-min-next-event-time` method to determine the minimum timestamp of all time-stamp-ordered events that may be subsequently delivered to the federate.

### Synopsis

ghla-interface::ghla-query-lbts

(*hla-interface*: ghla-interface)

-> *error-code*: symbol, *error-message*: text, *lbts*: float

Argument	Description
<i>hla-interface</i>	The HLA interface connection to use for the request.
Return Value	Description
<i>error-code</i>	The error code or NO-ERROR if no error occurred.
<i>error-message</i>	A message describing the error.
<i>lbts</i>	The current LBTS.

# ghla-interface::ghla-query-lookahead

Queries the length lookahead interval in effect for the local federate. The lookahead represents the length of the logical-time interval extending forward from the federate's logical time at any given point in execution in which the federate will not generate events. That is, the minimum permissible time-stamp for a time-stamp-ordered event generated by a time regulating federate is the federate's current logical time plus its lookahead. The sum of a federate's logical time and its lookahead may be referred to as the effective logical time of the federate. If the `ghla-modify-lookahead` method has been used to decrease the length of the lookahead interval, the new lookahead value will not take effect immediately. The specified lookahead will be phased in such that the effective logical time of the federate remains strictly non-decreasing. The new lookahead will completely take effect when the federate has advanced its logical time from the time at which the lookahead was decreased by an amount exceeding the difference between the new and old lookahead values. The lookahead setting is only meaningful for a time regulating federate: non-time-regulating federates may always generate events with any time stamp.

## Synopsis

ghla-interface::ghla-query-lookahead  
(*hla-interface*: ghla-interface)  
-> *error-code*: symbol, *error-message*: text, *lookahead*: float

Argument	Description
<i>hla-interface</i>	The HLA interface connection to use for the request.

  

Return Value	Description
<i>error-code</i>	The error code or NO-ERROR if no error occurred.
<i>error-message</i>	A message describing the error.
<i>lookahead</i>	The lookahead time.

## ghla-interface::ghla-query-min-next-event-time

Queries the minimum time-stamp of all time-stamp-ordered events that may be subsequently delivered to the federate. This service may be used to obtain the largest time stamp such that the LRC can guarantee that no time-stamp-ordered events will be subsequently delivered to the federate with an earlier time stamp. This is computed by taking the minimum of the effective federation lower-bound time stamp (see [ghla-interface::ghla-query-lbts](#)) and the time stamps of all time-stamp-ordered events currently queued for delivery to the federate (if any.) Note that the minimum next-event time for a non-time-constrained federate will generally be infinity. The effective LBTS of a non-time-constrained federate is infinity and no events will be queued for time-stamp-ordered delivery (although there may be some time-stamp-ordered events already queued when time-constraint was disabled.)

### Synopsis

ghla-interface::ghla-query-min-next-event-time

(*hla-interface*: ghla-interface)

-> *error-code*: symbol, *error-message*: text, *lookahead*: float

Argument	Description
<i>hla-interface</i>	The HLA interface connection to use for the request.
Return Value	Description
<i>error-code</i>	The error code or NO-ERROR if no error occurred.
<i>error-message</i>	A message describing the error.
<i>lookahead</i>	The time information.

# ghla-interface::ghla-register-federation-synchronization-point

Registers a synchronization point with the Federation. The Federate calling this method is requesting that a Federate part of the Federation be synchronized at the specified point. Any Federate part of the Federation is informed of this request via a call to `ghla-cb-announce-synchronization-point`. Once informed, each Federate should call the method `ghla-synchronization-point-achieved` once it reached the synchronization point.

## Synopsis

`ghla-interface::ghla-register-federation-synchronization-point`

(*hla-interface*: ghla-interface, *label*: text, *tag*: text)

-> *error-code*: symbol, *error-message*: text

Argument	Description
<i>hla-interface</i>	The HLA interface connection to use for the request.
<i>label</i>	The synchronization point keyword. This label is used to differentiate synchronization points.
<i>tag</i>	The synchronization point tag. The usage of this argument is application specific.

Return Value	Description
<i>error-code</i>	The error code or NO-ERROR if no error occurred.
<i>error-message</i>	A message describing the error.

## ghla-interface::ghla-register-object-instance

Registers a new object with the Federation. Any remote federate that subscribed to the HLA class will be informed via the `ghla-cb-discover-object-instance` callback method.

### Synopsis

`ghla-interface::ghla-register-object-instance`

(*ghla-interface*: `ghla-interface`, *ghla-class-handle*: `float`, *ghla-object-name*: `text`)

-> *error-code*: `symbol`, *error-message*: `text`, *ghla-object-handle*: `float`

Argument	Description
<i>ghla-interface</i>	The HLA interface connection to use for the request.
<i>ghla-class-handle</i>	The HLA class handle of the object to register.
<i>ghla-object-name</i>	The HLA name of the object to register.

  

Return Value	Description
<i>error-code</i>	The error code or NO-ERROR if no error occurred.
<i>error-message</i>	A message describing the error.
<i>ghla-object-handle</i>	The HLA object handle of the register object.

# ghla-interface::ghla-request-class-attribute-value-update

Requests the Federation to provide updated attribute values for all objects of the specified class. The new values are sent to the federate via the `ghla-cb-reflect-attribute-values` callback method.

## Synopsis

```
ghla-interface::ghla-request-class-attribute-value-update  
(hla-interface: ghla-interface, hla-class-handle: float,  
hla-attribute-names: symbol-array)  
-> error-code: symbol, error-message: text
```

Argument	Description
<i>hla-interface</i>	The HLA interface connection to use for the request.
<i>hla-class-handle</i>	The HLA class handle for which attribute value updates is requested for all instances.
<i>hla-attribute-names</i>	The list of attribute names for which a value update is requested.

Return Value	Description
<u><i>error-code</i></u>	The error code or NO-ERROR if no error occurred.
<u><i>error-message</i></u>	A message describing the error.

## ghla-interface::ghla-request-object-attribute-value-update

Request the Federation to provide updated attribute values for a HLA object. The new values will be sent to the federate via the `ghla-cb-reflect-attribute-values` callback method.

### Synopsis

```
ghla-interface::ghla-request-object-attribute-value-update
(hla-interface: ghla-interface, hla-object-handle: float,
 hla-attribute-names: symbol-array)
-> error-code: symbol, error-message: text
```

Argument	Description
<i>hla-interface</i>	The HLA interface connection to use for the request.
<i>hla-class-handle</i>	The HLA object handle for which an attribute value update is requested.
<i>hla-attribute-names</i>	The list of attribute names for which a value update is requested.
Return Value	Description
<u><i>error-code</i></u>	The error code or NO-ERROR if no error occurred.
<u><i>error-message</i></u>	A message describing the error.

# ghla-interface::ghla-resign-and-destroy-federation

Enables the G2-HLA Bridge application (Federate) to resign from the Federation. Once resigned, no other APIs should be called with the exception of `ghla-create-and-join-federation`. In addition, a call to this method optionally destroys the Federate Executive (Fedex).

## Synopsis

`ghla-interface::ghla-resign-and-destroy-federation`  
(*hla-interface*: ghla-interface, *destroy*: truth-value)  
-> *error-code*: symbol, *error-message*: text

Argument	Description
<i>hla-interface</i>	The HLA interface connection to use for the request.
<i>destroy</i>	If true, the call destroys the Federate executive (Fedex).

Return Value	Description
<i>error-code</i>	The error code or NO-ERROR if no error occurred.
<i>error-message</i>	A message describing the error.

## ghla-interface::ghla-send-interaction

Sends an interaction. The values of the parameters are extracted from the specified object by the G2-HLA Bridge. For information on the supported data types and the mechanism used when sending the values, see [ghla-interface::ghla-publish-interaction-class](#).

### Synopsis

ghla-interface::ghla-send-interaction

(*hla-interface*: ghla-interface, *hla-interaction-handle*: float, *hla-time*: float, *object*: item, *tag*: text)

-> *error-code*: symbol, *error-message*: text, *retraction-handle*: float

Argument	Description
<i>hla-interface</i>	The HLA interface connection to use for the request.
<i>hla-interaction-handle</i>	The HLA handle of the HLA interaction class.
<i>hla-time</i>	The federate time.
<i>object</i>	The G2 object from which to extract the attribute values.
<i>tag</i>	An application-specific tag that is passed to all federates receiving the interaction.
Return Value	Description
<i>error-code</i>	The error code or NO-ERROR if no error occurred.
<i>error-message</i>	A message describing the error.
<i>retraction-handle</i>	The retraction handle for this call.

# ghla-interface::ghla-subscribe-interaction-class

Subscribes to an HLA interaction class. If the *active* argument is set to *false*, the federation will not be notified of the subscription. Once a Federate has subscribed to an HLA interaction class, it will be informed of changes via the `ghla-cb-receive-interaction` callback method. The G2-HLA Bridge supports the following simple attribute types: integer, float, truth-value, symbol, and text. The current version does not support arrays, lists, parameters, variables or embedded objects. While sending and receiving values, the G2-HLA Bridge uses the G2 object attribute type to determine how to format (when sending) or decode (when receiving) a value. For information on the supported data types, see [ghla-interface::ghla-publish-interaction-class](#).

## Synopsis

`ghla-interface::ghla-subscribe-interaction-class`

(*hla-interface*: ghla-interface, *hla-class-name*: text, *g2-master-object*: item, *hla-attribute-names*: symbol-array, *active*: truth-value)

-> *error-code*: symbol, *error-message*: text, *hla-class-handle*: float, *hla-attribute-handles*: float-array

Argument	Description
<i>hla-interface</i>	The HLA interface connection to use for the request.
<i>hla-class-name</i>	The HLA name of the interaction class to which to subscribe.
<i>g2-master-object</i>	A G2 object corresponding to the published class. Objects of this type should be passed when updating attribute values.
<i>hla-attribute-names</i>	The list of HLA public attribute names. Therefore, your G2 objects may contain either public or private attributes. The HLA attribute name and the G2 attribute name must be identical.
<i>active</i>	Whether the subscription should be taken into account when advising publishing federates of object-registration relevance. When <i>false</i> , the federation will not be notified of the subscription.

<b>Return Value</b>	<b>Description</b>
<u><i>error-code</i></u>	The error code or NO-ERROR if no error occurred.
<u><i>error-message</i></u>	A message describing the error.
<u><i>hla-class-handle</i></u>	The HLA handle for the subscribed class.
<u><i>hla-attribute-handles</i></u>	The HLA handles for each of the published HLA attributes in the same order as <i>hla-attribute-names</i> .

# ghla-interface::ghla-subscribe-object-class-attributes

Subscribes to an HLA object class. If the *active* argument is set to *false*, the federation will not be notified of the subscription, in which case, no *ghla-cb-start-registration-for-object-class* callback will be made as a result to the subscription. A federate will discover object instances whenever instances are registered and updated after the subscription. Once a Federate has subscribed to an HLA class, it will be informed of changes via the following callback methods:

- *ghla-cb-discover-object-instance* informs the Federate that objects of the specified class have been registered. The discover callback method is always called once for each instance before the first attribute reflection callback method is called.
- *ghla-cb-remove-object-instance* informs the Federate that objects of the specified class have been deleted.
- *ghla-cb-reflect-attribute-values* informs the Federate of new attribute values of HLA objects.

For information on the supported data types, see [ghla-interface::ghla-publish-interaction-class](#).

## Synopsis

`ghla-interface::ghla-subscribe-object-class-attributes`

(*hla-interface*: `ghla-interface`, *hla-class-name*: `text`, *g2-master-object*: `item`,  
*hla-attribute-names*: `symbol-array`, *active*: `truth-value`)

-> *error-code*: `symbol`, *error-message*: `text`, *hla-class-handle*: `float`,  
*hla-attribute-handles*: `float-array`

Argument	Description
<i>hla-interface</i>	The HLA interface connection to use for the request.
<i>hla-class-name</i>	The HLA name of the interaction class to which to subscribe.
<i>g2-master-object</i>	A G2 object corresponding to the published class. Objects of this type should be passed when updating attribute values.

<b>Argument</b>	<b>Description</b>
<i>hla-attribute-names</i>	The list of attribute names to which to subscribe. Therefore, your G2 objects may contain both private and public attributes. In addition, it may only subscribe to the public HLA attributes. The HLA attribute name and the G2 attribute name must be identical.
<i>active</i>	Whether the subscription should be taken into account when advising publishing federates of object-registration relevance. When <b>false</b> , the federation will not be notified of the subscription.
<b>Return Value</b>	<b>Description</b>
<u><i>error-code</i></u>	The error code or NO-ERROR if no error occurred.
<u><i>error-message</i></u>	A message describing the error.
<u><i>hla-class-handle</i></u>	The HLA handle for the subscribed class.
<u><i>hla-attribute-handles</i></u>	The HLA handles for each of the published HLA attributes in the same order as <i>hla-attribute-names</i> .

# ghla-interface::ghla-synchronization-point-achieved

Subscribes to an HLA object class. If the *active* argument is set to *false*, the federation will not be notified of the subscription, in which case, no *ghla-cb-start-registration-for-object-class* callback will be made as a result to the subscription. A federate will discover object instances whenever instances are registered and updated after the subscription. Once a Federate has subscribed to an HLA class, it will be informed of changes via the following callback methods:

- *ghla-cb-discover-object-instance* informs the Federate that objects of the specified class have been registered. The discover callback method is always called once for each instance before the first attribute reflection callback method is called.
- *ghla-cb-remove-object-instance* informs the Federate that objects of the specified class have been deleted.
- *ghla-cb-reflect-attribute-values* informs the Federate of new attribute values of HLA objects.

For information on the supported data types, see [ghla-interface::ghla-publish-interaction-class](#).

## Synopsis

ghla-interface::ghla-synchronization-point-achieved

(*hla-interface*: ghla-interface, *label*: text)

-> *error-code*: symbol, *error-message*: text

Argument	Description
<i>hla-interface</i>	The HLA interface connection to use for the request.
<i>label</i>	The synchronization point keyword. This label is used to differentiate synchronization points.

  

Return Value	Description
<i>error-code</i>	The error code or NO-ERROR if no error occurred.
<i>error-message</i>	A message describing the error.

## ghla-interface::ghla-time-advance-request

Releases all time-stamp-ordered (TSO) events between the federate's current logical time and the requested time, inclusive, for delivery to the federate. Each relevant TSO event occurring in this interval will be delivered to the federate as soon as it can be guaranteed that all relevant TSO events with earlier time stamps have already been delivered. That is, an event will be delivered if and only if:

- All events currently queued for TSO delivery to the federate have a time stamp that is not greater than the event's time stamp.
- The federation LBTS is not less than the time stamp of the event (i.e., the LRC can guarantee that no TSO events with an earlier time stamp will arrive subsequently)

When all relevant TSO events with time stamps less than or equal to the requested time have been delivered to the federate (i.e., the minimum next-event time is greater than the requested time), the federate will receive the `ghla-cb-time-advance-grant` callback method indicating that the time-advancement has completed. Only after receiving such a callback may the federate proceed to initiate another time-advancement service. Subsequent to initiating a `ghla-time-advance-request`, the federate may not generate TSO events whose time stamps are less than the requested time plus the federate's lookahead. That is, the logical time of the federate is immediately set equal to the requested time upon a `ghla-time-advance-request` service invocation. For non-time-constrained federates, time advances are trivial; by definition such federates do not receive any TSO events, so a time-advance grant is immediately scheduled for delivery.

### Synopsis

```
ghla-interface::ghla-time-advance-request
(hla-interface: ghla-interface, the-time: float)
-> error-code: symbol, error-message: text
```

Argument	Description
<i>hla-interface</i>	The HLA interface connection to use for the request.
<i>the-time</i>	The time-stamp representing the point on the federation time axis to which to advance the federate's logical time.

<b>Return Value</b>	<b>Description</b>
<u><i>error-code</i></u>	The error code or NO-ERROR if no error occurred.
<u><i>error-message</i></u>	A message describing the error.

## ghla-interface::ghla-time-advance-request-available

Releases all time-stamp-ordered (TSO) events between the federate's current logical time and the requested time, inclusive, for delivery to the federate. Each relevant TSO event occurring in this interval will be delivered to the federate as soon as it can be guaranteed that all relevant TSO events with earlier time stamps have already been delivered. That is, an event will be delivered if and only if:

- All events currently queued for TSO delivery to the federate have a time stamp that is not greater than the event's time stamp.
- The federation LBTS is not less than the time stamp of the event (i.e., the LRC can guarantee that no TSO events with an earlier time stamp will arrive subsequently).

When all relevant TSO events with time stamps less than the requested time have been delivered to the federate (i.e., the minimum next-event time is greater than the requested time), the federate will receive a `ghla-cb-time-advance-grant` callback method indicating that the time-advancement has completed. Only after receiving such a callback may the federate proceed to initiate another time-advancement service. The available variant of this service does not necessarily deliver all TSO events occurring exactly at the requested time before making a `ghla-cb-time-advance-grant`. Subsequent to initiating `ghla-time-advance-request-available`, the federate may not generate TSO events whose time stamps are less than the requested time plus the federate's lookahead. That is, the logical time of the federate is immediately set equal to the requested time upon a `ghla-time-advance-request-available` service invocation. For non-time-constrained federates, time advances are trivial; by definition such federates do not receive any TSO events, so a time-advance grant is immediately scheduled for delivery.

### Synopsis

`ghla-interface::ghla-time-advance-request-available`

(*hla-interface*: `ghla-interface`, *the-time*: float)

-> *error-code*: symbol, *error-message*: text

Argument	Description
<i>hla-interface</i>	The HLA interface connection to use for the request.
<i>the-time</i>	The time-stamp representing the point on the federation time axis to which to advance the federate's logical time.

<b>Return Value</b>	<b>Description</b>
<u><i>error-code</i></u>	The error code or NO-ERROR if no error occurred.
<u><i>error-message</i></u>	A message describing the error.

# ghla-interface::ghla-unconditional-attribute-ownership-divestiture

Implements the “unconditional attribute ownership divestiture” service as specified by HLA.

## Synopsis

ghla-interface::ghla-unconditional-attribute-ownership-divestiture

(*hla-interface*: ghla-interface, *hla-object-handle*: float,

*hla-attribute-names*: symbol-array)

-> *error-code*: symbol, *error-message*: text

Argument	Description
<i>hla-interface</i>	The HLA interface connection to use for the request.
<i>hla-object-handle</i>	The HLA object handle for which to change the attribute ownership.
<i>hla-attribute-names</i>	The list of attribute names to take ownership with HLA.
Return Value	Description
<i>error-code</i>	The error code or NO-ERROR if no error occurred.
<i>error-message</i>	A message describing the error.

# ghla-interface::ghla-unpublish-interaction-class

Unpublishes an HLA interaction class previously published via a call to ghla-publish-interaction-class.

## Synopsis

ghla-interface::ghla-unpublish-interaction-class  
(*hla-interface*: ghla-interface, *hla-class-handle*: float)  
-> *error-code*: symbol, *error-message*: text

Argument	Description
<i>hla-interface</i>	The HLA interface connection to use for the request.
<i>hla-class-handle</i>	The HLA handle for the published interaction class.

Return Value	Description
<u><i>error-code</i></u>	The error code or NO-ERROR if no error occurred.
<u><i>error-message</i></u>	A message describing the error.

## ghla-interface::ghla-unpublish-object-class

Unpublishes an HLA class previously published via a call to `ghla-publish-object-class`.

### Synopsis

```
ghla-interface::ghla-unpublish-object-class
(hla-interface: ghla-interface, hla-class-handle: float)
-> error-code: symbol, error-message: text
```

Argument	Description
<i>hla-interface</i>	The HLA interface connection to use for the request.
<i>hla-class-handle</i>	The HLA handle for the published interaction class.

Return Value	Description
<u><i>error-code</i></u>	The error code or NO-ERROR if no error occurred.
<u><i>error-message</i></u>	A message describing the error.

# ghla-interface::ghla-unsubscribe-interaction-class

Unsubscribes from an HLA interaction class previously subscribed to via a call to ghla-subscribe-interaction-class.

## Synopsis

ghla-interface::ghla-unsubscribe-interaction-class  
(*hla-interface*: ghla-interface, *hla-class-handle*: float)  
-> *error-code*: symbol, *error-message*: text

Argument	Description
<i>hla-interface</i>	The HLA interface connection to use for the request.
<i>hla-class-handle</i>	The HLA handle for the subscribed interaction class.

Return Value	Description
<i>error-code</i>	The error code or NO-ERROR if no error occurred.
<i>error-message</i>	A message describing the error.

## ghla-interface::ghla-unsubscribe-object-class

Unsubscribes from an HLA class previously subscribed to via a call to `ghla-subscribe-object-class-attributes`.

### Synopsis

```
ghla-interface::ghla-unsubscribe-object-class
(ghla-interface: ghla-interface, ghla-class-handle: float)
-> error-code: symbol, error-message: text
```

Argument	Description
<i>ghla-interface</i>	The HLA interface connection to use for the request.
<i>ghla-class-handle</i>	The HLA handle for the subscribed class.

  

Return Value	Description
<u><i>error-code</i></u>	The error code or NO-ERROR if no error occurred.
<u><i>error-message</i></u>	A message describing the error.

# ghla-interface::ghla-update-attribute-values

Publishes the new attribute values of an HLA object. The values of the attributes are extracted from object by the G2-HLA Bridge. For a description of the data types and the mechanism used when sending the values, see [ghla-interface::ghla-publish-object-class](#).

## Synopsis

ghla-interface::ghla-update-attribute-values

(*hla-interface*: ghla-interface, *hla-object-handle*: float, *hla-time*: float, *object*: item, *hla-attribute-names*: symbol-array, *tag*: text)  
-> *error-code*: symbol, *error-message*: text, ???????: float

Argument	Description
<i>hla-interface</i>	The HLA interface connection to use for the request.
<i>hla-object-handle</i>	The HLA handle of the HLA object to update.
<i>hla-time</i>	The federate time.
<i>object</i>	The G2 object from which to extract the attribute values.
<i>hla-attribute-names</i>	The list of attribute names to update.
<i>tag</i>	An application-specific tag that is passed to all federates receiving the updated values.

Return Value	Description
<u><i>error-code</i></u>	The error code or NO-ERROR if no error occurred.
<u><i>error-message</i></u>	A message describing the error.
<u>???????</u>	????????????????????????????????

## Callback Methods

ghla-interface::ghla-cb-announce-synchronization-point  
ghla-interface::ghla-cb-attribute-is-not-owned  
ghla-interface::ghla-cb-attribute-owned-by-rti  
ghla-interface::ghla-cb-attribute-ownership-acquisition-notification  
ghla-interface::ghla-cb-attribute-ownership-divestiture-notification  
ghla-interface::ghla-cb-attribute-ownership-unavailable  
ghla-interface::ghla-cb-confirm-attribute-ownership-acquisition-cancellation  
ghla-interface::ghla-cb-discover-object-instance  
ghla-interface::ghla-cb-error-handler  
ghla-interface::ghla-cb-federation-synchronized  
ghla-interface::ghla-cb-inform-attribute-ownership  
ghla-interface::ghla-cb-provide-attribute-value-update  
ghla-interface::ghla-cb-receive-interaction  
ghla-interface::ghla-cb-reflect-attribute-values  
ghla-interface::ghla-cb-remove-object-instance  
ghla-interface::ghla-request-attribute-ownership-assumption  
ghla-interface::ghla-request-attribute-ownership-release  
ghla-interface::ghla-start-registration-for-object-class  
ghla-interface::ghla-stop-registration-for-object-class  
ghla-interface::ghla-synchronization-point-registration-failed  
ghla-interface::ghla-synchronization-point-registration-succeeded  
ghla-interface::ghla-time-advance-grant  
ghla-interface::ghla-time-constrained-enabled  
ghla-interface::ghla-time-regulation-enabled  
ghla-interface::ghla-turn-interactions-off  
ghla-interface::ghla-turn-interactions-on  
ghla-interface::ghla-turn-updates-off-for-object-instance  
ghla-interface::ghla-turn-updates-on-for-object-instance

# ghla-interface::ghla-cb-announce-synchronization-point

This callback method is called by the G2-HLA Bridge on behalf of the RTI to inform the G2 application that a Federate registered a synchronization point by calling `ghla-register-federation-synchronization-point`. The G2 application should perform any required operations to achieve the requested synchronization and then make a call to `ghla-synchronization-point-achieved`.

## Synopsis

`ghla-interface::ghla-cb-announce-synchronization-point`  
(*hla-interface*: ghla-interface, *label*: text, *tag*: text)

Argument	Description
<i>hla-interface</i>	The HLA interface that started this callback method.
<i>label</i>	The synchronization point keyword. This label is used to differentiate different synchronization points.
<i>tag</i>	The synchronization point tag. The usage of this argument is application specific.

## ghla-interface::ghla-cb-attribute-is-not-owned

This callback method is called by the G2-HLA Bridge on behalf of the RTI.

### Synopsis

ghla-interface::ghla-cb-attribute-is-not-owned  
(*hla-interface*: ghla-interface, *hla-object-handle*: float,  
*hla-attribute-name*: symbol)

Argument	Description
<i>hla-interface</i>	The HLA interface that started this callback method.
<i>hla-object-handle</i>	The object handle affected by the method.
<i>hla-attribute-name</i>	The attribute of the object affected by the method.

# ghla-interface::ghla-cb-attribute-owned-by-rti

This callback method is called by the G2-HLA Bridge on behalf of the RTI.

## Synopsis

ghla-interface::ghla-cb-attribute-owned-by-rti  
(*hla-interface*: ghla-interface, *hla-object-handle*: float,  
*hla-attribute-name*: symbol)

Argument	Description
<i>hla-interface</i>	The HLA interface that started this callback method.
<i>hla-object-handle</i>	The object handle affected by the method.
<i>hla-attribute-name</i>	The attribute of the object affected by the method.

## ghla-interface::ghla-cb-attribute-ownership-acquisition-notification

This callback method is called by the G2-HLA Bridge on behalf of the RTI.

### Synopsis

```
ghla-interface::ghla-cb-attribute-ownership-acquisition-notification
(hla-interface: ghla-interface, hla-object-handle: float,
 hla-attribute-name: symbol)
```

Argument	Description
<i>hla-interface</i>	The HLA interface that started this callback method.
<i>hla-object-handle</i>	The object handle affected by the method.
<i>hla-attribute-name</i>	The attribute of the object affected by the method.

# ghla-interface::ghla-cb-attribute-ownership-divestiture-notification

This callback method is called by the G2-HLA Bridge on behalf of the RTI.

## Synopsis

ghla-interface::ghla-cb-attribute-ownership-divestiture-notification  
(*hla-interface*: ghla-interface, *hla-object-handle*: float,  
*hla-attribute-name*: symbol)

Argument	Description
<i>hla-interface</i>	The HLA interface that started this callback method.
<i>hla-object-handle</i>	The object handle affected by the method.
<i>hla-attribute-name</i>	The attribute of the object affected by the method.

## ghla-interface::ghla-cb-attribute-ownership-unavailable

This callback method is called by the G2-HLA Bridge on behalf of the RTI.

### Synopsis

ghla-interface::ghla-cb-attribute-ownership-unavailable  
 (*hla-interface*: ghla-interface, *hla-object-handle*: float,  
*hla-attribute-name*: symbol)

Argument	Description
<i>hla-interface</i>	The HLA interface that started this callback method.
<i>hla-object-handle</i>	The object handle affected by the method.
<i>hla-attribute-name</i>	The attribute of the object affected by the method.

# ghla-interface::ghla-cb-confirm-attribute-ownership-acquisition-cancellation

This callback method is called by the G2-HLA Bridge on behalf of the RTI.

## Synopsis

ghla-interface::ghla-cb-confirm-attribute-ownership-acquisition-cancellation  
(*hla-interface*: ghla-interface, *hla-object-handle*: float,  
*hla-attribute-name*: symbol, *tag*: text)

Argument	Description
<i>hla-interface</i>	The HLA interface that started this callback method.
<i>hla-object-handle</i>	The object handle affected by the method.
<i>hla-attribute-name</i>	The attribute of the object affected by the method.
<i>tag</i>	The synchronization point tag. The usage of this argument is application specific.

## ghla-interface::ghla-cb-discover-object-instance

This callback method is called by the G2-HLA Bridge on behalf of the RTI to inform the G2 application that a new HLA object has been registered. The Federate may use the information to keep track of available HLA objects. Attribute value changes for these objects are forwarded to the G2 application via the `ghla-cb-reflect-attribute-values` callback method.

### Synopsis

`ghla-interface::ghla-cb-discover-object-instance`

(*hla-interface*: `ghla-interface`, *hla-class-handle*: `float`, *hla-object-handle*: `float`, *hla-object-name*: `text`)

Argument	Description
<i>hla-interface</i>	The HLA interface that started this callback method.
<i>hla-class-handle</i>	The class handle of the HLA object.
<i>hla-object-handle</i>	The HLA object handle of the instance.
<i>hla-object-name</i>	The HLA instance name.

## ghla-interface::ghla-cb-error-handler

This callback method is called by the G2-HLA Bridge to report errors that occurred in the bridge.

### Synopsis

ghla-interface::ghla-cb-error-handler

(*hla-interface*: ghla-interface, *error-symbol*: symbol, *error-text*: text)

Argument	Description
<i>hla-interface</i>	The HLA interface that generated the error.
<i>error-symbol</i>	The error code.
<i>error-text</i>	The error message.

## ghla-interface::ghla-cb-federation-synchronized

This callback method is called by the G2-HLA Bridge on behalf of the RTI to inform the G2 application that the Federation is synchronized at the specified synchronization point. The RTI informs each Federate part of a Federation that the synchronization point has been reached only after each Federate makes a call to `ghla-synchronization-point-achieved`.

### Synopsis

`ghla-interface::ghla-cb-federation-synchronized`  
*(hla-interface: ghla-interface, label: text)*

Argument	Description
<i>hla-interface</i>	The HLA interface that started this callback method.
<i>label</i>	The synchronization point keyword. This label is used to differentiate different synchronization points.

# ghla-interface::ghla-cb-inform-attribute-ownership

This callback method is called by the G2-HLA Bridge on behalf of the RTI.

## Synopsis

ghla-interface::ghla-cb-inform-attribute-ownership  
(*hla-interface*: ghla-interface, *hla-object-handle*: float,  
*hla-attribute-name*: symbol, *federate-handle*: float)

Argument	Description
<i>hla-interface</i>	The HLA interface that started this callback method.
<i>hla-object-handle</i>	The object handle affected by the method.
<i>hla-attribute-name</i>	The attribute of the object affected by the method.
<i>federate-name</i>	The Federate handle.

## ghla-interface::ghla-cb-provide-attribute-value-update

This callback method is called by the G2-HLA Bridge on behalf of the RTI to inform the G2 application that another Federate is requesting attribute value updates for an HLA object that was registered by the G2 Federate. The G2 application should compute the attribute values for the specified object and call `ghla-update-attribute-values` to update the values. The G2 application does not need to provide new values for all public attributes but only for the ones specified in the *hla-attribute-names* argument.

### Synopsis

```
ghla-interface::ghla-cb-provide-attribute-value-update
(hla-interface: ghla-interface, hla-object-handle: float,
 hla-attribute-names: symbol-array)
```

Argument	Description
<i>hla-interface</i>	The HLA interface that started this callback method.
<i>hla-object-handle</i>	The HLA object handle for which attribute values are requested.
<i>hla-attribute-names</i>	The list of attribute values to provide.

# ghla-interface::ghla-cb-receive-interaction

This callback method is called by the G2-HLA Bridge on behalf of the RTI to inform the G2 application that another Federate sent an interaction. Only Federates that subscribed to the interaction class are informed of changes. For a description of the supported data types and the mechanism used when sending the values, see [ghla-interface::ghla-send-interaction](#).

## Synopsis

ghla-interface::ghla-cb-receive-interaction

(*hla-interface*: ghla-interface, *hla-time*: float, *hla-interaction-handle*: float, *hla-attribute-names*: symbol-array, *object*: item, *tag*: text )

Argument	Description
<i>hla-interface</i>	The HLA interface that started this callback method.
<i>hla-time</i>	The time corresponding to this update.
<i>hla-interaction-handle</i>	The HLA interaction handle for which new attribute values are provided.
<i>hla-attribute-names</i>	The list of attribute names that contain new values.
<i>object</i>	The G2 object containing the attribute values. The G2-HLA Bridge stores the master object passed during the <code>ghla-subscribe-interaction-class</code> method call and makes copies of it for each received interaction. Before calling this callback method, it updates the parameter values listed in <i>hla-attribute-names</i> .
<i>tag</i>	An application-specific tag name that was passed to the <code>ghla-send-interaction</code> method.

## ghla-interface::ghla-cb-reflect-attribute-values

This callback method is called by the G2-HLA Bridge on behalf of the RTI to inform the G2 application that another Federate provided new attribute values for an HLA object. Only Federates that subscribed to the attributes that changed are informed of changes. For a description of the supported data types and the mechanism used when sending the values, see the documentation for the `hla-subscribe-object-class` method

### Synopsis

ghla-interface::ghla-cb-reflect-attribute-values

(*hla-interface*: ghla-interface, *hla-time*: float, *hla-object-handle*: float, *hla-attribute-names*: symbol-array, *object*: item, *tag*: text )

Argument	Description
<i>hla-interface</i>	The HLA interface that started this callback method.
<i>hla-time</i>	The time corresponding to this update.
<i>hla-interaction-handle</i>	The HLA interaction handle for which new attribute values are provided.
<i>hla-attribute-names</i>	The list of attribute names that contain new values.
<i>object</i>	The G2 object containing the attribute values. The G2-HLA Bridge stores the master object passed during the <code>ghla-subscribe-object-class-attributes</code> method call and makes copies of it for each attribute reflection. Before calling this callback method, it updates the attribute values listed in <i>hla-attribute-names</i> .
<i>tag</i>	An application-specific tag name that was passed to the <code>ghla-update-attribute-values</code> method.

## ghla-interface::ghla-cb-remove-object-instance

This callback method is called by the G2-HLA Bridge on behalf of the RTI to inform the G2 application that an HLA object should be removed. The G2 application should remove all references to this object.

### Synopsis

ghla-interface::ghla-cb-remove-object-instance

(*hla-interface*: ghla-interface, *hla-object-handle*: float, *hla-time*: float, *tag*: text )

Argument	Description
<i>hla-interface</i>	The HLA interface that started this callback method.
<i>hla-object-handle</i>	The HLA handle of the object to remove.
<i>hla-time</i>	The time corresponding to this update.
<i>tag</i>	An application-specific tag.

## ghla-interface::ghla-request-attribute-ownership-assumption

This callback method is called by the G2-HLA Bridge on behalf of the RTI.

### Synopsis

ghla-interface::ghla-cb-request-attribute-ownership-assumption

(*hla-interface*: ghla-interface, *hla-object-handle*: float,

*hla-attribute-names*: symbol-array, *tag*: text )

Argument	Description
<i>hla-interface</i>	The HLA interface that started this callback method.
<i>hla-object-handle</i>	The object handle affected by this method.
<i>hla-attribute-names</i>	The affected attribute names of the object.
<i>tag</i>	The user-defined tag name.

# ghla-interface::ghla-request-attribute-ownership-release

This callback method is called by the G2-HLA Bridge on behalf of the RTI.

## Synopsis

ghla-interface::ghla-cb-request-attribute-ownership-release

(*hla-interface*: ghla-interface, *hla-object-handle*: float,

*hla-attribute-names*: symbol-array, *tag*: text )

Argument	Description
<i>hla-interface</i>	The HLA interface that started this callback method.
<i>hla-object-handle</i>	The object handle affected by this method.
<i>hla-attribute-names</i>	The affected attribute names of the object.
<i>tag</i>	The user-defined tag name.

## ghla-interface::ghla-start-registration-for-object-class

This callback method is called by the G2-HLA Bridge on behalf of the RTI to inform the G2 application that the Federate may start registering objects of the specified class. The RTI informs the Federate of this state in case another Federate has interest (subscribed) in at least one attribute of objects of the class published by your G2 Federate application.

### Synopsis

ghla-interface::ghla-cb-start-registration-for-object-class  
 (*hla-interface*: ghla-interface, *hla-class-handle*: float)

Argument	Description
<i>hla-interface</i>	The HLA interface that started this callback method.
<i>hla-class-handle</i>	The HLA object class handle that the Federate published.

## ghla-interface::ghla-stop-registration-for-object-class

This callback method is called by the G2-HLA Bridge on behalf of the RTI to inform the G2 application that the Federate may stop registering objects of the specified class. The RTI informs the Federate of this state in case no other Federate has interest in (i.e., subscribed to) attributes of objects of the class published by your G2 Federate application.

### Synopsis

ghla-interface::ghla-cb-stop-registration-for-object-class  
(*hla-interface*: ghla-interface, *hla-class-handle*: float)

Argument	Description
<i>hla-interface</i>	The HLA interface that started this callback method.
<i>hla-class-handle</i>	The HLA object class handle that the Federate published.

## ghla-interface::ghla-synchronization-point-registration-failed

This callback method is called by the G2-HLA Bridge on behalf of the RTI to inform the G2 application that the `ghla-register-federation-synchronization-point` method call failed. This method is only called for the Federate that issued the request by calling `ghla-register-federation-synchronization-point`.

### Synopsis

`ghla-interface::ghla-cb-synchronization-point-registration-failed`  
*(hla-interface: ghla-interface, label: text)*

Argument	Description
<i>hla-interface</i>	The HLA interface that started this callback method.
<i>label</i>	The synchronization point keyword. This label is used to differentiate synchronization points.

# ghla-interface::ghla-synchronization-point-registration-succeeded

This callback method is called by the G2-HLA Bridge on behalf of the RTI to inform the G2 application that the `ghla-register-federation-synchronization-point` method call succeeded. This method is only called for the Federate that issued the request by calling `ghla-register-federation-synchronization-point`.

## Synopsis

`ghla-interface::ghla-cb-synchronization-point-registration-succeeded`  
(*hla-interface*: `ghla-interface`, *label*: `text`)

Argument	Description
<i>hla-interface</i>	The HLA interface that started this callback method.
<i>label</i>	The synchronization point keyword. This label is used to differentiate synchronization points.

## ghla-interface::ghla-time-advance-grant

This callback method is called by the G2-HLA Bridge on behalf of the RTI.

### Synopsis

ghla-interface::ghla-cb-time-advance-grant  
(*hla-interface*: ghla-interface, *the-time*: float)

Argument	Description
<i>hla-interface</i>	The HLA interface that started this callback method.
<i>the-time</i>	The time.

# ghla-interface::ghla-time-constrained-enabled

This callback method is called by the G2-HLA Bridge on behalf of the RTI.

## Synopsis

ghla-interface::ghla-cb-time-constrained-enabled  
(*hla-interface*: ghla-interface, *federate-time*: float)

Argument	Description
<i>hla-interface</i>	The HLA interface that started this callback method.
<i>federate-time</i>	The Federate time.

## ghla-interface::ghla-time-regulation-enabled

This callback method is called by the G2-HLA Bridge on behalf of the RTI.

### Synopsis

ghla-interface::ghla-cb-time-regulation-enabled  
(*hla-interface*: ghla-interface, *federate-time*: float)

Argument	Description
<i>hla-interface</i>	The HLA interface that started this callback method.
<i>federate-time</i>	The Federate time.

## ghla-interface::ghla-turn-interactions-off

This callback method is called by the G2-HLA Bridge on behalf of the RTI to inform the G2 application that the Federate may turn the interactions off for the specified class. The RTI informs the Federate of this state in case no other Federate has interest in (i.e., subscribed to) the specified interaction class.

### Synopsis

ghla-interface::ghla-cb-turn-interactions-off

(*hla-interface*: ghla-interface, *hla-interaction-handle*: float)

Argument	Description
<i>hla-interface</i>	The HLA interface that started this callback method.
<i>hla-interaction-handle</i>	The target registered HLA object.

## ghla-interface::ghla-turn-interactions-on

This callback method is called by the G2-HLA Bridge on behalf of the RTI to inform the G2 application that the Federate may turn the interactions on for the specified class. The RTI informs the Federate of this state in case another Federate has interest in (i.e., subscribed to) the specified interaction class.

### Synopsis

ghla-interface::ghla-cb-turn-interactions-on

(*hla-interface*: ghla-interface, *hla-interaction-handle*: float)

Argument	Description
<i>hla-interface</i>	The HLA interface that started this callback method.
<i>hla-interaction-handle</i>	The target registered HLA object.

# ghla-interface::ghla-turn-updates-off-for-object-instance

This callback method is called by the G2-HLA Bridge on behalf of the RTI to inform the G2 application that no another Federate has interest in the specified attributes of the specified object. Therefore, the Federate does not need to send attribute value updates for the specified attributes of the specified object. This callback may be used to improve the efficiency of the application. If no other Federate has interest in attributes of some objects, there is no need to send value updates as values change.

## Synopsis

ghla-interface::ghla-cb-turn-updates-off-for-object-instance  
(*hla-interface*: ghla-interface, *hla-object-handle*: float,  
*hla-attribute-names*: symbol-array)

Argument	Description
<i>hla-interface</i>	The HLA interface that started this callback method.
<i>hla-object-handle</i>	The target registered HLA object.
<i>hla-attribute-names</i>	The list of attribute names to which another Federate subscribed.

## ghla-interface::ghla-turn-updates-on-for-object-instance

This callback method is called by the G2-HLA Bridge on behalf of the RTI to inform the G2 application that at least another Federate has interest in the specified attributes of the specified object. Therefore, the Federate should send attribute value updates for the specified attributes of the specified object. This callback may be used to improve the efficiency of the application. If no other federate has interest in attributes of some objects. There is no need to send value updates as values change.

### Synopsis

```
ghla-interface::ghla-cb-turn-updates-on-for-object-instance
(hla-interface: ghla-interface, hla-object-handle: float,
 hla-attribute-names: symbol-array)
```

Argument	Description
<i>hla-interface</i>	The HLA interface that started this callback method.
<i>hla-object-handle</i>	The target registered HLA object.
<i>hla-attribute-names</i>	The list of attribute names to which another Federate subscribed.



@ 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**

callback methods  
customer support services

**G**

## G2-HLA Bridge (GHLA)

application programmers? interface  
callback methods  
examples  
    Federate  
    Hello World Federate  
introduction to  
loading  
request methods  
using  
ghla-interface attributes  
ghla-interface::ghla-attribute-ownership-acquisition  
ghla-interface::ghla-cb-announce-synchronization-point  
ghla-interface::ghla-cb-attribute-is-not-owned  
ghla-interface::ghla-cb-attribute-owned-by-rti  
ghla-interface::ghla-cb-attribute-ownership-acquisition-notification  
ghla-interface::ghla-cb-attribute-ownership-divestiture-notification  
ghla-interface::ghla-cb-attribute-ownership-unavailable  
ghla-interface::ghla-cb-confirm-attribute-ownership-acquisition-cancellation  
ghla-interface::ghla-cb-discover-object-instance  
ghla-interface::ghla-cb-error-handler  
ghla-interface::ghla-cb-federation-synchronized  
ghla-interface::ghla-cb-inform-attribute-ownership  
ghla-interface::ghla-cb-provide-attribute-value-update  
ghla-interface::ghla-cb-receive-interaction  
ghla-interface::ghla-cb-reflect-attribute-values

ghla-interface::ghla-cb-remove-object-instance  
ghla-interface::ghla-enable-class-relevance-advisory-switch  
ghla-interface::ghla-enable-interaction-relevance-advisory-switch  
ghla-interface::ghla-enable-time-constrained  
ghla-interface::ghla-enable-time-regulation  
ghla-interface::ghla-flush-queue-request  
ghla-interface::ghla-get-object-instance-name  
ghla-interface::ghla-is-attribute-owned-by-federate  
ghla-interface::ghla-modify-lookahead  
ghla-interface::ghla-negotiate-attribute-ownership-divestiture  
ghla-interface::ghla-next-event-request  
ghla-interface::ghla-next-event-request-available  
ghla-interface::ghla-publish-interaction-class  
ghla-interface::ghla-publish-object-class  
ghla-interface::ghla-query-attribute-ownership  
ghla-interface::ghla-query-federate-time  
ghla-interface::ghla-query-lbts  
ghla-interface::ghla-query-lookahead  
ghla-interface::ghla-query-min-next-event-time  
ghla-interface::ghla-register-federation-synchronization-point  
ghla-interface::ghla-register-object-instance  
ghla-interface::ghla-request-attribute-ownership-assumption  
ghla-interface::ghla-request-attribute-ownership-release  
ghla-interface::ghla-request-class-attribute-value-update  
ghla-interface::ghla-request-object-attribute-value-update  
ghla-interface::ghla-resign-and-destroy-federation  
ghla-interface::ghla-send-interaction  
ghla-interface::ghla-start-registration-for-object-class  
ghla-interface::ghla-stop-registration-for-object-class  
ghla-interface::ghla-subscribe-interaction-class

ghla-interface::ghla-subscribe-object-class-attributes  
ghla-interface::ghla-synchronization-point-achieved  
ghla-interface::ghla-synchronization-point-registration-failed  
ghla-interface::ghla-synchronization-point-registration-succeeded  
ghla-interface::ghla-time-advance-grant  
ghla-interface::ghla-time-advance-request  
ghla-interface::ghla-time-advance-request-available  
ghla-interface::ghla-time-constrained-enabled  
ghla-interface::ghla-time-regulation-enabled  
ghla-interface::ghla-turn-interactions-off  
ghla-interface::ghla-turn-interactions-on  
ghla-interface::ghla-turn-updates-off-for-object-instance  
ghla-interface::ghla-turn-updates-on-for-object-instance  
ghla-interface::ghla-unconditional-attribute-ownership-divestiture  
ghla-interface::ghla-unpublish-interaction-class  
ghla-interface::ghla-unpublish-object-class  
ghla-interface::ghla-unsubscribe-interaction-class  
ghla-interface::ghla-unsubscribe-object-class  
ghla-interface::ghla-update-attribute-values

## M

methods  
    callback  
    request

## R

request methods  
`RTI_HOME` environment variable