



## C interfaces to GALAHAD HASH

Jari Fowkes and Nick Gould  
STFC Rutherford Appleton Laboratory  
Mon May 1 2023



<b>1 GALAHAD C package gltr</b>	<b>1</b>
1.1 Introduction	1
1.1.1 Purpose	1
1.1.2 Authors	1
1.1.3 Originally released	1
1.1.4 Terminology	2
1.1.5 Method	2
1.1.6 Reference	2
1.1.7 Call order	2
<b>2 File Index</b>	<b>3</b>
2.1 File List	3
<b>3 File Documentation</b>	<b>5</b>
3.1 galahad_gltr.h File Reference	5
3.1.1 Data Structure Documentation	5
3.1.1.1 struct gltr_control_type	5
3.1.1.2 struct gltr_inform_type	6
3.1.2 Function Documentation	7
3.1.2.1 gltr_initialize()	7
3.1.2.2 gltr_read_specfile()	7
3.1.2.3 gltr_import_control()	8
3.1.2.4 gltr_solve_problem()	8
3.1.2.5 gltr_information()	9
3.1.2.6 gltr_terminate()	10
<b>4 Example Documentation</b>	<b>11</b>
4.1 gltrt.c	11



# Chapter 1

## GALAHAD C package hash

### 1.1 Introduction

#### 1.1.1 Purpose

Set up, insert into, remove from and search a chained scatter table (Williams, CACM 2, 21-24, 1959).

Currently, only the control and inform parameters are exposed; these are provided and used by other GALAHAD packages with C interfaces.

#### 1.1.2 Authors

N. I. M. Gould, STFC-Rutherford Appleton Laboratory, England.

C interface, additionally J. Fowkes, STFC-Rutherford Appleton Laboratory.

Julia interface, additionally A. Montoison and D. Orban, Polytechnique Montréal.

#### 1.1.3 Originally released

December 1990, C interface January 2022.



## Chapter 2

# File Index

### 2.1 File List

Here is a list of all files with brief descriptions:

<a href="#">galahad_hash.h</a>	.....	??
--------------------------------	-------	----





## Chapter 3

# File Documentation

### 3.1 galahad\_hash.h File Reference

```
#include <stdbool.h>
#include <stdint.h>
#include "galahad_precision.h"
#include "galahad_cfunctions.h"
```

#### Data Structures

- struct [hash\\_control\\_type](#)
- struct [hash\\_inform\\_type](#)

#### 3.1.1 Data Structure Documentation

##### 3.1.1.1 struct hash\_control\_type

###### Data Fields

int	error	error and warning diagnostics occur on stream error
int	out	general output occurs on stream out
int	print_level	the level of output required. Possible values are: <ul style="list-style-type: none"><li>• <math>\leq 0</math> no output,</li><li>• <math>\geq 1</math> debugging</li></ul>
bool	space_critical	if space_critical true, every effort will be made to use as little space as possible. This may result in longer computation time
bool	deallocate_error_fatal	if deallocate_error_fatal is true, any array/pointer deallocation error will terminate execution. Otherwise, computation will continue
char	prefix[31]	all output lines will be prefixed by prefix(2:LEN(TRIM(prefix))-1) where prefix contains the required string enclosed in quotes, e.g. "string" or 'string'

### 3.1.1.2 struct hash\_inform\_type

#### Data Fields

int	status	return status. Possible values are: <ul style="list-style-type: none"><li>• 0 The insertion or deletion was succesful.</li><li>• -1. An allocation error occurred. A message indicating the offending array is written on unit control.error, and the returned allocation status and a string containing the name of the offending array are held in inform.alloc_status and inform.bad_alloc respectively.</li><li>• -2. A deallocation error occurred. A message indicating the offending array is written on unit control.error and the returned allocation status and a string containing the name of the offending array are held in inform.alloc_status and inform.bad_alloc respectively.</li><li>• -99. The current dictionary is full and should be rebuilt with more space.</li></ul>
int	alloc_status	the status of the last attempted allocation/deallocation.
char	bad_alloc[81]	the name of the array for which an allocation/deallocation error occurred.