

DyLP
trunk

Generated by Doxygen 1.8.5

Mon Oct 21 2013 19:04:43

Contents

1	Todo List	1
2	Hierarchical Index	2
2.1	Class Hierarchy	2
3	Class Index	6
3.1	Class List	6
4	File Index	8
4.1	File List	8
5	Class Documentation	9
5.1	attvhdr_struct_tag Struct Reference	9
5.1.1	Detailed Description	9
5.1.2	Member Data Documentation	9
5.2	basis_struct Struct Reference	9
5.2.1	Detailed Description	10
5.2.2	Member Data Documentation	10
5.3	basisel_struct Struct Reference	10
5.3.1	Detailed Description	10
5.3.2	Member Data Documentation	10
5.4	bnfdef_any Union Reference	11
5.4.1	Detailed Description	11
5.4.2	Member Data Documentation	11
5.5	bnfdef_struct Struct Reference	12
5.5.1	Detailed Description	12
5.6	bnfGdef_struct Struct Reference	12
5.6.1	Detailed Description	12
5.6.2	Member Data Documentation	12
5.7	bnfldef_struct Struct Reference	12
5.7.1	Detailed Description	13
5.7.2	Member Data Documentation	13
5.8	bnflBdef_struct Struct Reference	13
5.8.1	Detailed Description	13
5.8.2	Member Data Documentation	13
5.9	bnfldef_struct Struct Reference	14
5.9.1	Detailed Description	14

5.9.2	Member Data Documentation	14
5.10	bnfNPdef_struct Struct Reference	15
5.10.1	Detailed Description	15
5.10.2	Member Data Documentation	15
5.11	bnfref_any Union Reference	15
5.11.1	Detailed Description	15
5.11.2	Member Data Documentation	15
5.12	bnfref_struct_tag Struct Reference	16
5.12.1	Detailed Description	16
5.13	bnfref_type2 Struct Reference	17
5.13.1	Detailed Description	17
5.13.2	Member Data Documentation	17
5.14	bnfref_type3 Struct Reference	17
5.14.1	Detailed Description	17
5.14.2	Member Data Documentation	17
5.15	bnfTdef_struct Struct Reference	18
5.15.1	Detailed Description	18
5.15.2	Member Data Documentation	18
5.16	coeff_struct_tag Struct Reference	18
5.16.1	Detailed Description	19
5.16.2	Member Data Documentation	19
5.17	colhdr_struct_tag Struct Reference	19
5.17.1	Detailed Description	19
5.17.2	Member Data Documentation	20
5.18	conbnd_struct Struct Reference	20
5.18.1	Detailed Description	20
5.18.2	Member Data Documentation	20
5.19	conmtx_struct Struct Reference	21
5.19.1	Detailed Description	21
5.19.2	Member Data Documentation	21
5.20	consys_struct Struct Reference	21
5.20.1	Detailed Description	22
5.20.2	Member Data Documentation	22
5.21	ENV Struct Reference	25
5.21.1	Detailed Description	25
5.21.2	Member Data Documentation	25
5.22	hel_tag Struct Reference	26

5.22.1 Detailed Description	26
5.22.2 Member Data Documentation	26
5.23 INV Struct Reference	26
5.23.1 Detailed Description	27
5.23.2 Member Data Documentation	27
5.24 keytab_entry_internal Struct Reference	28
5.24.1 Detailed Description	28
5.24.2 Member Data Documentation	28
5.25 lex_struct Struct Reference	29
5.25.1 Detailed Description	29
5.25.2 Member Data Documentation	29
5.26 Ink_struct_tag Struct Reference	29
5.26.1 Detailed Description	29
5.26.2 Member Data Documentation	30
5.27 lpopts_struct Struct Reference	30
5.27.1 Detailed Description	32
5.27.2 Member Data Documentation	32
5.28 lpprob_struct Struct Reference	36
5.28.1 Detailed Description	37
5.28.2 Member Data Documentation	37
5.29 lpstats_struct Struct Reference	38
5.29.1 Detailed Description	40
5.29.2 Member Data Documentation	40
5.30 lptols_struct Struct Reference	43
5.30.1 Detailed Description	44
5.30.2 Member Data Documentation	44
5.31 LUF Struct Reference	45
5.31.1 Detailed Description	46
5.31.2 Member Data Documentation	46
5.32 LUF_WA Struct Reference	49
5.32.1 Detailed Description	50
5.32.2 Member Data Documentation	50
5.33 MEM Struct Reference	50
5.33.1 Detailed Description	50
5.33.2 Member Data Documentation	51
5.34 OsiDyIpSolverInterface Class Reference	51
5.34.1 Detailed Description	57

5.34.2	Constructor & Destructor Documentation	57
5.34.3	Member Function Documentation	57
5.34.4	Friends And Related Function Documentation	66
5.34.5	Member Data Documentation	66
5.35	OsiDyLpWarmStartBasis Class Reference	67
5.35.1	Detailed Description	69
5.35.2	Constructor & Destructor Documentation	69
5.35.3	Member Function Documentation	69
5.36	OsiDyLpWarmStartBasisDiff Class Reference	71
5.36.1	Detailed Description	72
5.36.2	Constructor & Destructor Documentation	72
5.36.3	Member Function Documentation	72
5.36.4	Friends And Related Function Documentation	73
5.37	parse_any Union Reference	73
5.37.1	Detailed Description	73
5.37.2	Member Data Documentation	73
5.38	pkcoeff_struct Struct Reference	73
5.38.1	Detailed Description	73
5.38.2	Member Data Documentation	74
5.39	pkvec_struct Struct Reference	74
5.39.1	Detailed Description	74
5.39.2	Member Data Documentation	74
5.40	POOL Struct Reference	75
5.40.1	Detailed Description	75
5.40.2	Member Data Documentation	75
5.41	rowhdr_struct_tag Struct Reference	76
5.41.1	Detailed Description	76
5.41.2	Member Data Documentation	76
6	File Documentation	76
6.1	/home/ted/COIN/trunk/DyLP/src/DyLP/dy_cmdint.h File Reference	76
6.1.1	Macro Definition Documentation	77
6.1.2	Enumeration Type Documentation	77
6.1.3	Function Documentation	77
6.2	/home/ted/COIN/trunk/DyLP/src/DyLP/dy_consys.h File Reference	77
6.2.1	Macro Definition Documentation	80
6.2.2	Typedef Documentation	82

6.2.3	Enumeration Type Documentation	82
6.2.4	Function Documentation	83
6.3	/home/ted/COIN/trunk/DyLP/src/DyLP/dy_vector.h File Reference	85
6.3.1	Function Documentation	85
6.4	/home/ted/COIN/trunk/DyLP/src/DyLP/dydp.h File Reference	86
6.4.1	Macro Definition Documentation	88
6.4.2	Enumeration Type Documentation	91
6.4.3	Function Documentation	94
6.5	/home/ted/COIN/trunk/DyLP/src/DyLP/glpinv.h File Reference	95
6.5.1	Macro Definition Documentation	96
6.5.2	Typedef Documentation	96
6.5.3	Function Documentation	96
6.6	/home/ted/COIN/trunk/DyLP/src/DyLP/glpilib.h File Reference	96
6.6.1	Macro Definition Documentation	98
6.6.2	Typedef Documentation	99
6.6.3	Function Documentation	99
6.7	/home/ted/COIN/trunk/DyLP/src/DyLP/glpluf.h File Reference	100
6.7.1	Macro Definition Documentation	101
6.7.2	Typedef Documentation	102
6.7.3	Function Documentation	102
6.8	/home/ted/COIN/trunk/DyLP/src/DyLPStdLib/config_default.h File Reference	102
6.8.1	Macro Definition Documentation	103
6.9	/home/ted/COIN/trunk/DyLP/src/DyLPStdLib/config_dydp_default.h File Reference	103
6.9.1	Macro Definition Documentation	103
6.10	/home/ted/COIN/trunk/DyLP/src/DyLPStdLib/dylib_bnfdr.h File Reference	104
6.10.1	Macro Definition Documentation	106
6.10.2	Typedef Documentation	111
6.10.3	Enumeration Type Documentation	112
6.10.4	Function Documentation	113
6.11	/home/ted/COIN/trunk/DyLP/src/DyLPStdLib/dylib_errs.h File Reference	113
6.11.1	Function Documentation	113
6.12	/home/ted/COIN/trunk/DyLP/src/DyLPStdLib/dylib_fortran.h File Reference	113
6.12.1	Macro Definition Documentation	114
6.12.2	Typedef Documentation	115
6.12.3	Variable Documentation	115
6.13	/home/ted/COIN/trunk/DyLP/src/DyLPStdLib/dylib_hash.h File Reference	116
6.13.1	Typedef Documentation	116

6.13.2	Function Documentation	116
6.14	/home/ted/COIN/trunk/DyLP/src/DyLPStdLib/dylib_io.h File Reference	116
6.14.1	Macro Definition Documentation	117
6.14.2	Typedef Documentation	118
6.14.3	Enumeration Type Documentation	118
6.14.4	Function Documentation	118
6.15	/home/ted/COIN/trunk/DyLP/src/DyLPStdLib/dylib_keytab.h File Reference	119
6.15.1	Typedef Documentation	119
6.15.2	Function Documentation	119
6.16	/home/ted/COIN/trunk/DyLP/src/DyLPStdLib/dylib_std.h File Reference	119
6.16.1	Macro Definition Documentation	120
6.16.2	Typedef Documentation	121
6.17	/home/ted/COIN/trunk/DyLP/src/DyLPStdLib/dylib_strtrns.h File Reference	122
6.17.1	Macro Definition Documentation	122
6.17.2	Function Documentation	122
6.18	/home/ted/COIN/trunk/DyLP/src/DyLPStdLib/DyLPConfig.h File Reference	122
6.19	/home/ted/COIN/trunk/DyLP/src/OsiDyLP/OsiDyLPMessages.hpp File Reference	123
6.19.1	Enumeration Type Documentation	123
6.20	/home/ted/COIN/trunk/DyLP/src/OsiDyLP/OsiDyLPSolverInterface.hpp File Reference	124
6.20.1	Detailed Description	125
6.20.2	Macro Definition Documentation	125
6.20.3	Enumeration Type Documentation	125
6.21	/home/ted/COIN/trunk/DyLP/src/OsiDyLP/OsiDyLPWarmStartBasis.hpp File Reference	125
6.21.1	Detailed Description	126
6.21.2	Macro Definition Documentation	126

Index

127

1 Todo List

Class **OsiDyLPWarmStartBasisDiff**

This is a pretty generic structure, and vector diff is a pretty generic activity. We should be able to convert this to a template.

Using unsigned int as the data type for the diff vectors might help to contain the damage when this code is inevitably compiled for 64 bit architectures. But the notion of int as 4 bytes is hardwired into CoinWarmStartBasis, so changes are definitely required.

2 Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

std::allocator< T >	
std::array< T >	
attvhdr_struct_tag	9
std::auto_ptr< T >	
std::basic_string< Char >	
std::string	
std::wstring	
std::basic_string< char >	
std::basic_string< wchar_t >	
basis_struct	9
basisel_struct	10
std::bitset< Bits >	
bnfdef_any	11
bnfdef_struct	12
bnfGdef_struct	12
bnfIdef_struct	12
bnfLBdef_struct	13
bnfLdef_struct	14
bnfNPdef_struct	15
bnfref_any	15
bnfref_struct_tag	16
bnfref_type2	17
bnfref_type3	17
bnfTdef_struct	18
coeff_struct_tag	18
CoinWarmStartBasis	
OsiDyIpWarmStartBasis	67
CoinWarmStartBasisDiff	
OsiDyIpWarmStartBasisDiff	71
colhdr_struct_tag	19
std::complex	

conbnd_struct 20

conmtx_struct 21

```

std::list< T >::const_iterator
std::forward_list< T >::const_iterator
std::map< K, T >::const_iterator
std::unordered_map< K, T >::const_iterator
std::basic_string< Char >::const_iterator
std::multimap< K, T >::const_iterator
std::unordered_multimap< K, T >::const_iterator
std::set< K >::const_iterator
std::string::const_iterator
std::unordered_set< K >::const_iterator
std::multiset< K >::const_iterator
std::unordered_multiset< K >::const_iterator
std::wstring::const_iterator
std::vector< T >::const_iterator
std::deque< T >::const_iterator
std::list< T >::const_reverse_iterator
std::forward_list< T >::const_reverse_iterator
std::map< K, T >::const_reverse_iterator
std::unordered_set< K >::const_reverse_iterator
std::unordered_map< K, T >::const_reverse_iterator
std::wstring::const_reverse_iterator
std::multimap< K, T >::const_reverse_iterator
std::set< K >::const_reverse_iterator
std::string::const_reverse_iterator
std::multiset< K >::const_reverse_iterator
std::basic_string< Char >::const_reverse_iterator
std::unordered_multimap< K, T >::const_reverse_iterator
std::unordered_multiset< K >::const_reverse_iterator
std::vector< T >::const_reverse_iterator
std::deque< T >::const_reverse_iterator

```

consys_struct 21

std::deque< T >

ENV 25

```

std::error_category
std::error_code
std::error_condition
std::exception
    std::bad_alloc
    std::bad_cast
    std::bad_exception
    std::bad_typeid
    std::ios_base::failure
    std::logic_error
        std::domain_error
        std::invalid_argument
        std::length_error
        std::out_of_range
    std::runtime_error
        std::overflow_error
        std::range_error

```

std::underflow_error
std::forward_list< T >

hel_tag

26

INV

26

std::ios_base
 basic_ios< char >
 basic_ios< wchar_t >
 std::basic_ios
 basic_istream< char >
 basic_istream< wchar_t >
 basic_ostream< char >
 basic_ostream< wchar_t >
 std::basic_istream
 basic_ifstream< char >
 basic_ifstream< wchar_t >
 basic_iostream< char >
 basic_iostream< wchar_t >
 basic_istreamstream< char >
 basic_istreamstream< wchar_t >
 std::basic_ifstream
 std::ifstream
 std::wifstream
 std::basic_iostream
 basic_fstream< char >
 basic_fstream< wchar_t >
 basic_stringstream< char >
 basic_stringstream< wchar_t >
 std::basic_fstream
 std::fstream
 std::wfstream
 std::basic_stringstream
 std::stringstream
 std::wstringstream
 std::basic_istreamstream
 std::istreamstream
 std::wistreamstream
 std::istream
 std::wistream
 std::basic_ostream
 basic_ostream< char >
 basic_ostream< wchar_t >
 basic_ofstream< char >
 basic_ofstream< wchar_t >
 basic_ostreamstream< char >
 basic_ostreamstream< wchar_t >
 std::basic_ostream
 std::basic_ofstream
 std::ofstream
 std::wofstream
 std::basic_ostreamstream
 std::ostreamstream
 std::wostringstream

std::ostream	
std::wostream	
std::ios	
std::wios	
std::forward_list< T >::iterator	
std::map< K, T >::iterator	
std::wstring::iterator	
std::multiset< K >::iterator	
std::unordered_multiset< K >::iterator	
std::multimap< K, T >::iterator	
std::unordered_multimap< K, T >::iterator	
std::unordered_map< K, T >::iterator	
std::unordered_set< K >::iterator	
std::string::iterator	
std::set< K >::iterator	
std::list< T >::iterator	
std::basic_string< Char >::iterator	
std::vector< T >::iterator	
std::deque< T >::iterator	
keytab_entry_internal	28
lex_struct	29
std::list< T >	
Ink_struct_tag	29
lpopts_struct	30
lpprob_struct	36
lpstats_struct	38
lptols_struct	43
LUF	45
LUF_WA	49
std::map< K, T >	
MEM	50
std::multimap< K, T >	
std::multiset< K >	
OsiSolverInterface	
OsiDyLpSolverInterface	51
parse_any	73
pkcoeff_struct	73
pkvec_struct	74
POOL	75
std::priority_queue< T >	
std::queue< T >	
std::unordered_multimap< K, T >::reverse_iterator	

```

std::multimap< K, T >::reverse_iterator
std::wstring::reverse_iterator
std::forward_list< T >::reverse_iterator
std::unordered_set< K >::reverse_iterator
std::list< T >::reverse_iterator
std::basic_string< Char >::reverse_iterator
std::string::reverse_iterator
std::map< K, T >::reverse_iterator
std::unordered_map< K, T >::reverse_iterator
std::vector< T >::reverse_iterator
std::multiset< K >::reverse_iterator
std::unordered_multiset< K >::reverse_iterator
std::deque< T >::reverse_iterator
std::set< K >::reverse_iterator

```

rowhdr_struct_tag

76

```

std::set< K >
std::smart_ptr< T >
std::stack< T >
std::system_error
std::thread
std::unique_ptr< T >
std::unordered_map< K, T >
std::unordered_multimap< K, T >
std::unordered_multiset< K >
std::unordered_set< K >
std::valarray< T >
std::vector< T >
std::weak_ptr< T >
K
T

```

3 Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

attvhdr_struct_tag	9
basis_struct	9
basisel_struct	10
bnfdef_any	11
bnfdef_struct	12
bnfGdef_struct	12
bnfldef_struct	12
bnfLBdef_struct	13

bnfLdef_struct	14
bnfNPdef_struct	15
bnfref_any	15
bnfref_struct_tag	16
bnfref_type2	17
bnfref_type3	17
bnfTdef_struct	18
coeff_struct_tag	18
colhdr_struct_tag	19
conbnd_struct	20
conmtx_struct	21
consys_struct	21
ENV	25
hel_tag	26
INV	26
keytab_entry_internal	28
lex_struct	29
lnk_struct_tag	29
lpopts_struct	30
lpprob_struct	36
lpstats_struct	38
lptols_struct	43
LUF	45
LUF_WA	49
MEM	50
OsiDylpSolverInterface COIN OSI API for dylp	51
OsiDylpWarmStartBasis The dylp warm start class	67
OsiDylpWarmStartBasisDiff A 'diff' between two OsiDylpWarmStartBasis objects	71

parse_any	73
pkcoeff_struct	73
pkvec_struct	74
POOL	75
rowhdr_struct_tag	76

4 File Index

4.1 File List

Here is a list of all files with brief descriptions:

/home/ted/COIN/trunk/DyLP/src/DyLP/dy_cmdint.h	76
/home/ted/COIN/trunk/DyLP/src/DyLP/dy_consys.h	77
/home/ted/COIN/trunk/DyLP/src/DyLP/dy_vector.h	85
/home/ted/COIN/trunk/DyLP/src/DyLP/dyLP.h	86
/home/ted/COIN/trunk/DyLP/src/DyLP/glpinv.h	95
/home/ted/COIN/trunk/DyLP/src/DyLP/glpLib.h	96
/home/ted/COIN/trunk/DyLP/src/DyLP/glpLuf.h	100
/home/ted/COIN/trunk/DyLP/src/DyLPStdLib/config_default.h	102
/home/ted/COIN/trunk/DyLP/src/DyLPStdLib/config_dyLP_default.h	103
/home/ted/COIN/trunk/DyLP/src/DyLPStdLib/dylib_bnfrdr.h	104
/home/ted/COIN/trunk/DyLP/src/DyLPStdLib/dylib_errs.h	113
/home/ted/COIN/trunk/DyLP/src/DyLPStdLib/dylib_fortran.h	113
/home/ted/COIN/trunk/DyLP/src/DyLPStdLib/dylib_hash.h	116
/home/ted/COIN/trunk/DyLP/src/DyLPStdLib/dylib_io.h	116
/home/ted/COIN/trunk/DyLP/src/DyLPStdLib/dylib_keytab.h	119
/home/ted/COIN/trunk/DyLP/src/DyLPStdLib/dylib_std.h	119
/home/ted/COIN/trunk/DyLP/src/DyLPStdLib/dylib_strtns.h	122
/home/ted/COIN/trunk/DyLP/src/DyLPStdLib/DyLPConfig.h	122
/home/ted/COIN/trunk/DyLP/src/OsiDyLP/OsiDyLPMessages.hpp	123
/home/ted/COIN/trunk/DyLP/src/OsiDyLP/OsiDyLP SolverInterface.hpp	
Declarations of the COIN OSI API for the dyLP solver	124

[/home/ted/COIN/trunk/DyLP/src/OsiDyIp/OsiDyIpWarmStartBasis.hpp](#)

Copyright (C) 2003 – 2007 Lou Hafer, International Business Machines Corporation and others

125

5 Class Documentation

5.1 attvhdr_struct_tag Struct Reference

```
#include <dy_consys.h>
```

Public Attributes

- struct [attvhdr_struct_tag](#) * [nxt](#)
- [flags](#) [what](#)
- int [elsze](#)
- void * [vec](#)
- [lnk_struct](#) * [pveclst](#)

5.1.1 Detailed Description

Definition at line 267 of file [dy_consys.h](#).

5.1.2 Member Data Documentation

5.1.2.1 struct attvhdr_struct_tag* attvhdr_struct_tag::nxt

Definition at line 267 of file [dy_consys.h](#).

5.1.2.2 flags attvhdr_struct_tag::what

Definition at line 268 of file [dy_consys.h](#).

5.1.2.3 int attvhdr_struct_tag::elsze

Definition at line 269 of file [dy_consys.h](#).

5.1.2.4 void* attvhdr_struct_tag::vec

Definition at line 270 of file [dy_consys.h](#).

5.1.2.5 lnk_struct* attvhdr_struct_tag::pveclst

Definition at line 271 of file [dy_consys.h](#).

The documentation for this struct was generated from the following file:

- [/home/ted/COIN/trunk/DyLP/src/DyIp/dy_consys.h](#)

5.2 basis_struct Struct Reference

```
#include <dylp.h>
```

Public Attributes

- int [len](#)
- [basisel_struct](#) * [el](#)

5.2.1 Detailed Description

Definition at line 453 of file [dylp.h](#).

5.2.2 Member Data Documentation

5.2.2.1 int [basis_struct::len](#)

Definition at line 454 of file [dylp.h](#).

5.2.2.2 [basisel_struct](#)* [basis_struct::el](#)

Definition at line 455 of file [dylp.h](#).

The documentation for this struct was generated from the following file:

- [/home/ted/COIN/trunk/DyLP/src/Dylp/dylp.h](#)

5.3 [basisel_struct](#) Struct Reference

```
#include <dylp.h>
```

Public Attributes

- int [cndx](#)
- int [vndx](#)

5.3.1 Detailed Description

Definition at line 451 of file [dylp.h](#).

5.3.2 Member Data Documentation

5.3.2.1 int [basisel_struct::cndx](#)

Definition at line 451 of file [dylp.h](#).

5.3.2.2 int [basisel_struct::vndx](#)

Definition at line 451 of file [dylp.h](#).

The documentation for this struct was generated from the following file:

- [/home/ted/COIN/trunk/DyLP/src/Dylp/dylp.h](#)

5.4 bnfdef_any Union Reference

```
#include <dylib_bnfrdr.h>
```

Public Attributes

- [bnfdef_struct * com](#)
- [bnfGdef_struct * G](#)
- [bnfNPdef_struct * NP](#)
- [bnfPdef_struct * P](#)
- [bnfTdef_struct * T](#)
- [bnfIdef_struct * I](#)
- [bnfLdef_struct * L](#)
- [bnfLBdef_struct * LB](#)

5.4.1 Detailed Description

Definition at line 427 of file dylib_bnfrdr.h.

5.4.2 Member Data Documentation

5.4.2.1 [bnfdef_struct* bnfdef_any::com](#)

Definition at line 427 of file dylib_bnfrdr.h.

5.4.2.2 [bnfGdef_struct* bnfdef_any::G](#)

Definition at line 428 of file dylib_bnfrdr.h.

5.4.2.3 [bnfNPdef_struct* bnfdef_any::NP](#)

Definition at line 429 of file dylib_bnfrdr.h.

5.4.2.4 [bnfPdef_struct* bnfdef_any::P](#)

Definition at line 430 of file dylib_bnfrdr.h.

5.4.2.5 [bnfTdef_struct* bnfdef_any::T](#)

Definition at line 431 of file dylib_bnfrdr.h.

5.4.2.6 [bnfIdef_struct* bnfdef_any::I](#)

Definition at line 432 of file dylib_bnfrdr.h.

5.4.2.7 [bnfLdef_struct* bnfdef_any::L](#)

Definition at line 433 of file dylib_bnfrdr.h.

5.4.2.8 [bnfLBdef_struct* bnfdef_any::LB](#)

Definition at line 434 of file dylib_bnfrdr.h.

The documentation for this union was generated from the following file:

- [/home/ted/COIN/trunk/DyLP/src/DyLPStdLib/dylib_bnfrdr.h](#)

5.5 bnfdef_struct Struct Reference

```
#include <dylib_bnfrdr.h>
```

5.5.1 Detailed Description

Definition at line 266 of file `dylib_bnfrdr.h`.

The documentation for this struct was generated from the following file:

- [/home/ted/COIN/trunk/DyLP/src/DyLPStdLib/dylib_bnfrdr.h](#)

5.6 bnfGdef_struct Struct Reference

```
#include <dylib_bnfrdr.h>
```

Public Attributes

- [bnfdef_common](#) int [size](#)
- int [link](#)
- struct [bnfref_struct_tag](#) ** [comps](#)

5.6.1 Detailed Description

Definition at line 285 of file `dylib_bnfrdr.h`.

5.6.2 Member Data Documentation

5.6.2.1 [bnfdef_common](#) int [bnfGdef_struct::size](#)

Definition at line 286 of file `dylib_bnfrdr.h`.

5.6.2.2 int [bnfGdef_struct::link](#)

Definition at line 287 of file `dylib_bnfrdr.h`.

5.6.2.3 struct [bnfref_struct_tag](#)** [bnfGdef_struct::comps](#)

Definition at line 288 of file `dylib_bnfrdr.h`.

The documentation for this struct was generated from the following file:

- [/home/ted/COIN/trunk/DyLP/src/DyLPStdLib/dylib_bnfrdr.h](#)

5.7 bnfldef_struct Struct Reference

```
#include <dylib_bnfrdr.h>
```

Public Attributes

- [bnfdef_common](#) int [ival](#)

5.7.1 Detailed Description

Definition at line 355 of file [dylib_bnfrdr.h](#).

5.7.2 Member Data Documentation

5.7.2.1 [bnfdef_common](#) int [bnfdef_struct::ival](#)

Definition at line 356 of file [dylib_bnfrdr.h](#).

The documentation for this struct was generated from the following file:

- [/home/ted/COIN/trunk/DyLP/src/DyLPStdLib/dylib_bnfrdr.h](#)

5.8 bnfLBdef_struct Struct Reference

```
#include <dylib_bnfrdr.h>
```

Public Attributes

- [bnfdef_common](#) flags [dflgs](#)
- [bnflblsrc_enum](#) [nmcd](#)
- [bnflblsrc_enum](#) [ndcd](#)
- int [savnm](#)
- struct [bnfref_struct_tag](#) * [nmsrc](#)
- int [savnd](#)
- struct [bnfref_struct_tag](#) * [ndsrc](#)
- int [offset](#)
- int [offset2](#)

5.8.1 Detailed Description

Definition at line 406 of file [dylib_bnfrdr.h](#).

5.8.2 Member Data Documentation

5.8.2.1 [bnfdef_common](#) flags [bnfLBdef_struct::dflgs](#)

Definition at line 407 of file [dylib_bnfrdr.h](#).

5.8.2.2 [bnflblsrc_enum](#) [bnfLBdef_struct::nmcd](#)

Definition at line 408 of file [dylib_bnfrdr.h](#).

5.8.2.3 [bnflblsrc_enum](#) [bnfLBdef_struct::ndcd](#)

Definition at line 409 of file [dylib_bnfrdr.h](#).

5.8.2.4 `int bnfLBdef_struct::savnm`

Definition at line 410 of file `dylib_bnfrdr.h`.

5.8.2.5 `struct bnfref_struct_tag* bnfLBdef_struct::nmsrc`

Definition at line 411 of file `dylib_bnfrdr.h`.

5.8.2.6 `int bnfLBdef_struct::savnd`

Definition at line 412 of file `dylib_bnfrdr.h`.

5.8.2.7 `struct bnfref_struct_tag* bnfLBdef_struct::ndsrc`

Definition at line 413 of file `dylib_bnfrdr.h`.

5.8.2.8 `int bnfLBdef_struct::offset`

Definition at line 414 of file `dylib_bnfrdr.h`.

5.8.2.9 `int bnfLBdef_struct::offset2`

Definition at line 415 of file `dylib_bnfrdr.h`.

The documentation for this struct was generated from the following file:

- [/home/ted/COIN/trunk/DyLP/src/DyLPStdLib/dylib_bnfrdr.h](#)

5.9 `bnfLdef_struct` Struct Reference

```
#include <dylib_bnfrdr.h>
```

Public Attributes

- [bnfdef_common flags dflgs](#)
- `char * txt`

5.9.1 Detailed Description

Definition at line 371 of file `dylib_bnfrdr.h`.

5.9.2 Member Data Documentation

5.9.2.1 `bnfdef_common flags bnfLdef_struct::dflgs`

Definition at line 372 of file `dylib_bnfrdr.h`.

5.9.2.2 `char* bnfLdef_struct::txt`

Definition at line 373 of file `dylib_bnfrdr.h`.

The documentation for this struct was generated from the following file:

- [/home/ted/COIN/trunk/DyLP/src/DyLPStdLib/dylib_bnfrdr.h](#)

5.10 bnfNPdef_struct Struct Reference

```
#include <dylib_bnfdr.h>
```

Public Attributes

- [bnfdef_common](#) struct
[bnfref_struct_tag](#) *** [alts](#)

5.10.1 Detailed Description

Definition at line 301 of file [dylib_bnfdr.h](#).

5.10.2 Member Data Documentation

5.10.2.1 [bnfdef_common](#) struct [bnfref_struct_tag](#)*** [bnfNPdef_struct::alts](#)

Definition at line 302 of file [dylib_bnfdr.h](#).

The documentation for this struct was generated from the following file:

- [/home/ted/COIN/trunk/DyLP/src/DyLPStdLib/dylib_bnfdr.h](#)

5.11 bnfref_any Union Reference

```
#include <dylib_bnfdr.h>
```

Public Attributes

- [bnfref_struct](#) * [com](#)
- struct [bnfref_type1](#) * [t1](#)
- struct [bnfref_type2](#) * [t2](#)
- struct [bnfref_type3](#) * [t3](#)
- [bnfGref_struct](#) * [G](#)
- [bnfNPref_struct](#) * [NP](#)
- [bnfPref_struct](#) * [P](#)
- [bnfTref_struct](#) * [T](#)
- [bnfIref_struct](#) * [I](#)
- [bnfLref_struct](#) * [L](#)
- [bnfLBref_struct](#) * [LB](#)

5.11.1 Detailed Description

Definition at line 522 of file [dylib_bnfdr.h](#).

5.11.2 Member Data Documentation

5.11.2.1 [bnfref_struct](#)* [bnfref_any::com](#)

Definition at line 522 of file [dylib_bnfdr.h](#).

5.11.2.2 `struct bnfref_type1* bnfref_any::t1`

Definition at line 523 of file `dylib_bnfrdr.h`.

5.11.2.3 `struct bnfref_type2* bnfref_any::t2`

Definition at line 524 of file `dylib_bnfrdr.h`.

5.11.2.4 `struct bnfref_type3* bnfref_any::t3`

Definition at line 525 of file `dylib_bnfrdr.h`.

5.11.2.5 `bnfGref_struct* bnfref_any::G`

Definition at line 526 of file `dylib_bnfrdr.h`.

5.11.2.6 `bnfNPref_struct* bnfref_any::NP`

Definition at line 527 of file `dylib_bnfrdr.h`.

5.11.2.7 `bnfPref_struct* bnfref_any::P`

Definition at line 528 of file `dylib_bnfrdr.h`.

5.11.2.8 `bnfTref_struct* bnfref_any::T`

Definition at line 529 of file `dylib_bnfrdr.h`.

5.11.2.9 `bnfIref_struct* bnfref_any::I`

Definition at line 530 of file `dylib_bnfrdr.h`.

5.11.2.10 `bnfLref_struct* bnfref_any::L`

Definition at line 531 of file `dylib_bnfrdr.h`.

5.11.2.11 `bnfLBref_struct* bnfref_any::LB`

Definition at line 532 of file `dylib_bnfrdr.h`.

The documentation for this union was generated from the following file:

- /home/ted/COIN/trunk/DyLP/src/DyLPStdLib/dylib_bnfrdr.h

5.12 `bnfref_struct_tag` Struct Reference

```
#include <dylib_bnfrdr.h>
```

5.12.1 Detailed Description

Definition at line 464 of file `dylib_bnfrdr.h`.

The documentation for this struct was generated from the following file:

- /home/ted/COIN/trunk/DyLP/src/DyLPStdLib/dylib_bnfrdr.h

5.13 bnref_type2 Struct Reference

```
#include <dylib_bnfrdr.h>
```

Public Attributes

- [bnref_common](#) int [offset](#)

5.13.1 Detailed Description

Definition at line 487 of file dylib_bnfrdr.h.

5.13.2 Member Data Documentation

5.13.2.1 bnref_common int bnref_type2::offset

Definition at line 488 of file dylib_bnfrdr.h.

The documentation for this struct was generated from the following file:

- [/home/ted/COIN/trunk/DyLP/src/DyLPStdLib/dylib_bnfrdr.h](#)

5.14 bnref_type3 Struct Reference

```
#include <dylib_bnfrdr.h>
```

Public Attributes

- [bnref_common](#) int [offset](#)
- [bnref_struct](#) * [sep](#)

5.14.1 Detailed Description

Definition at line 508 of file dylib_bnfrdr.h.

5.14.2 Member Data Documentation

5.14.2.1 bnref_common int bnref_type3::offset

Definition at line 509 of file dylib_bnfrdr.h.

5.14.2.2 bnref_struct* bnref_type3::sep

Definition at line 510 of file dylib_bnfrdr.h.

The documentation for this struct was generated from the following file:

- [/home/ted/COIN/trunk/DyLP/src/DyLPStdLib/dylib_bnfrdr.h](#)

5.15 bnfTdef_struct Struct Reference

```
#include <dylib_bnfrdr.h>
```

Public Attributes

- [bnfdef_common bnftype_enum ttype](#)
- char [qschr](#)
- char [qechr](#)
- int [parm1](#)
- const char * [val](#)

5.15.1 Detailed Description

Definition at line 337 of file dylib_bnfrdr.h.

5.15.2 Member Data Documentation

5.15.2.1 [bnfdef_common bnftype_enum bnfTdef_struct::ttype](#)

Definition at line 338 of file dylib_bnfrdr.h.

5.15.2.2 [char bnfTdef_struct::qschr](#)

Definition at line 339 of file dylib_bnfrdr.h.

5.15.2.3 [char bnfTdef_struct::qechr](#)

Definition at line 340 of file dylib_bnfrdr.h.

5.15.2.4 [int bnfTdef_struct::parm1](#)

Definition at line 341 of file dylib_bnfrdr.h.

5.15.2.5 [const char* bnfTdef_struct::val](#)

Definition at line 342 of file dylib_bnfrdr.h.

The documentation for this struct was generated from the following file:

- [/home/ted/COIN/trunk/DyLP/src/DyLPStdLib/dylib_bnfrdr.h](#)

5.16 coeff_struct_tag Struct Reference

```
#include <dy_consys.h>
```

Public Attributes

- struct [rowhdr_struct_tag](#) * [rowhdr](#)
- struct [colhdr_struct_tag](#) * [colhdr](#)
- double [val](#)

- struct [coeff_struct_tag](#) * rownxt
- struct [coeff_struct_tag](#) * colnxt

5.16.1 Detailed Description

Definition at line 102 of file dy_consys.h.

5.16.2 Member Data Documentation

5.16.2.1 struct rowhdr_struct_tag* coeff_struct_tag::rowhdr

Definition at line 103 of file dy_consys.h.

5.16.2.2 struct colhdr_struct_tag* coeff_struct_tag::colhdr

Definition at line 104 of file dy_consys.h.

5.16.2.3 double coeff_struct_tag::val

Definition at line 105 of file dy_consys.h.

5.16.2.4 struct coeff_struct_tag* coeff_struct_tag::rownxt

Definition at line 106 of file dy_consys.h.

5.16.2.5 struct coeff_struct_tag* coeff_struct_tag::colnxt

Definition at line 107 of file dy_consys.h.

The documentation for this struct was generated from the following file:

- [/home/ted/COIN/trunk/DyLP/src/DyLP/dy_consys.h](#)

5.17 colhdr_struct_tag Struct Reference

```
#include <dy_consys.h>
```

Public Attributes

- int [ndx](#)
- int [len](#)
- const char * [nme](#)
- [coeff_struct](#) * [coeffs](#)

5.17.1 Detailed Description

Definition at line 120 of file dy_consys.h.

5.17.2 Member Data Documentation

5.17.2.1 `int colhdr_struct_tag::ndx`

Definition at line 121 of file `dy_consys.h`.

5.17.2.2 `int colhdr_struct_tag::len`

Definition at line 122 of file `dy_consys.h`.

5.17.2.3 `const char* colhdr_struct_tag::nme`

Definition at line 123 of file `dy_consys.h`.

5.17.2.4 `coeff_struct* colhdr_struct_tag::coeffs`

Definition at line 124 of file `dy_consys.h`.

The documentation for this struct was generated from the following file:

- [/home/ted/COIN/trunk/DyLP/src/DyLP/dy_consys.h](#)

5.18 `conbnd_struct` Struct Reference

```
#include <dy_consys.h>
```

Public Attributes

- `int` [revs](#)
- `int` [inf](#)
- `double` [bnd](#)

5.18.1 Detailed Description

Definition at line 308 of file `dy_consys.h`.

5.18.2 Member Data Documentation

5.18.2.1 `int conbnd_struct::revs`

Definition at line 308 of file `dy_consys.h`.

5.18.2.2 `int conbnd_struct::inf`

Definition at line 309 of file `dy_consys.h`.

5.18.2.3 `double conbnd_struct::bnd`

Definition at line 310 of file `dy_consys.h`.

The documentation for this struct was generated from the following file:

- [/home/ted/COIN/trunk/DyLP/src/DyLP/dy_consys.h](#)

5.19 conmtx_struct Struct Reference

```
#include <dy_consys.h>
```

Public Attributes

- int [coeffcnt](#)
- [colhdr_struct](#) ** [cols](#)
- [rowhdr_struct](#) ** [rows](#)

5.19.1 Detailed Description

Definition at line 153 of file [dy_consys.h](#).

5.19.2 Member Data Documentation

5.19.2.1 int conmtx_struct::coeffcnt

Definition at line 154 of file [dy_consys.h](#).

5.19.2.2 colhdr_struct** conmtx_struct::cols

Definition at line 155 of file [dy_consys.h](#).

5.19.2.3 rowhdr_struct** conmtx_struct::rows

Definition at line 156 of file [dy_consys.h](#).

The documentation for this struct was generated from the following file:

- [/home/ted/COIN/trunk/DyLP/src/DyLP/dy_consys.h](#)

5.20 consys_struct Struct Reference

```
#include <dy_consys.h>
```

Public Attributes

- const char * [nme](#)
- [flags](#) [parts](#)
- [flags](#) [opts](#)
- double [inf](#)
- double [tiny](#)
- int [varcnt](#)
- int [archvcnt](#)
- int [logvcnt](#)
- int [intvcnt](#)
- int [binvcnt](#)
- int [maxcollen](#)
- int [maxcolndx](#)

- int [concnt](#)
- int [archccnt](#)
- int [cutccnt](#)
- int [maxrowlen](#)
- int [maxrowndx](#)
- int [colsze](#)
- int [rowsze](#)
- [conmtx_struct](#) [mtx](#)
- double [maxaij](#)
- double [minaij](#)
- double * [rowscale](#)
- double * [colscale](#)
- const char * [objnme](#)
- int [objndx](#)
- int [xzndx](#)
- double * [obj](#)
- [vartyp_enum](#) * [vtyp](#)
- double * [vub](#)
- double * [vlb](#)
- double * [rhs](#)
- double * [rhslow](#)
- [contyp_enum](#) * [ctyp](#)
- [conbnd_struct](#) * [cub](#)
- [conbnd_struct](#) * [clb](#)
- [attvhdr_struct](#) * [attvecs](#)

5.20.1 Detailed Description

Definition at line 460 of file `dy_consys.h`.

5.20.2 Member Data Documentation

5.20.2.1 `const char* consys_struct::nme`

Definition at line 461 of file `dy_consys.h`.

5.20.2.2 `flags consys_struct::parts`

Definition at line 462 of file `dy_consys.h`.

5.20.2.3 `flags consys_struct::opts`

Definition at line 463 of file `dy_consys.h`.

5.20.2.4 `double consys_struct::inf`

Definition at line 464 of file `dy_consys.h`.

5.20.2.5 `double consys_struct::tiny`

Definition at line 465 of file `dy_consys.h`.

5.20.2.6 int consys_struct::varcnt

Definition at line 466 of file dy_consys.h.

5.20.2.7 int consys_struct::archvcnt

Definition at line 467 of file dy_consys.h.

5.20.2.8 int consys_struct::logvcnt

Definition at line 468 of file dy_consys.h.

5.20.2.9 int consys_struct::intvcnt

Definition at line 469 of file dy_consys.h.

5.20.2.10 int consys_struct::binvcnt

Definition at line 470 of file dy_consys.h.

5.20.2.11 int consys_struct::maxcollen

Definition at line 471 of file dy_consys.h.

5.20.2.12 int consys_struct::maxcolndx

Definition at line 472 of file dy_consys.h.

5.20.2.13 int consys_struct::concnt

Definition at line 473 of file dy_consys.h.

5.20.2.14 int consys_struct::archccnt

Definition at line 474 of file dy_consys.h.

5.20.2.15 int consys_struct::cutccnt

Definition at line 475 of file dy_consys.h.

5.20.2.16 int consys_struct::maxrowlen

Definition at line 476 of file dy_consys.h.

5.20.2.17 int consys_struct::maxrowndx

Definition at line 477 of file dy_consys.h.

5.20.2.18 int consys_struct::colsze

Definition at line 478 of file dy_consys.h.

5.20.2.19 int consys_struct::rowsze

Definition at line 479 of file dy_consys.h.

5.20.2.20 conmtx_struct consys_struct::mtx

Definition at line 480 of file dy_consys.h.

5.20.2.21 double consys_struct::maxaij

Definition at line 481 of file dy_consys.h.

5.20.2.22 double consys_struct::minaij

Definition at line 482 of file dy_consys.h.

5.20.2.23 double* consys_struct::rowscale

Definition at line 483 of file dy_consys.h.

5.20.2.24 double* consys_struct::colscale

Definition at line 484 of file dy_consys.h.

5.20.2.25 const char* consys_struct::objnme

Definition at line 485 of file dy_consys.h.

5.20.2.26 int consys_struct::objndx

Definition at line 486 of file dy_consys.h.

5.20.2.27 int consys_struct::xzndx

Definition at line 487 of file dy_consys.h.

5.20.2.28 double* consys_struct::obj

Definition at line 488 of file dy_consys.h.

5.20.2.29 vartyp_enum* consys_struct::vtyp

Definition at line 489 of file dy_consys.h.

5.20.2.30 double* consys_struct::vub

Definition at line 490 of file dy_consys.h.

5.20.2.31 double* consys_struct::vlb

Definition at line 491 of file dy_consys.h.

5.20.2.32 double* consys_struct::rhs

Definition at line 492 of file dy_consys.h.

5.20.2.33 double* consys_struct::rhslow

Definition at line 493 of file dy_consys.h.

5.20.2.34 `contyp_enum*` `consys_struct::ctyp`

Definition at line 494 of file `dy_consys.h`.

5.20.2.35 `conbnd_struct*` `consys_struct::cub`

Definition at line 495 of file `dy_consys.h`.

5.20.2.36 `conbnd_struct*` `consys_struct::clb`

Definition at line 496 of file `dy_consys.h`.

5.20.2.37 `attvhdr_struct*` `consys_struct::attvecs`

Definition at line 497 of file `dy_consys.h`.

The documentation for this struct was generated from the following file:

- `/home/ted/COIN/trunk/DyLP/src/DyLP/dy_consys.h`

5.21 ENV Struct Reference

```
#include <glplib.h>
```

Public Attributes

- `MEM *` `mem_ptr`
- `int` `mem_limit`
- `int` `mem_total`
- `int` `mem_tpeak`
- `int` `mem_count`
- `int` `mem_cpeak`

5.21.1 Detailed Description

Definition at line 53 of file `glplib.h`.

5.21.2 Member Data Documentation

5.21.2.1 `MEM*` `ENV::mem_ptr`

Definition at line 55 of file `glplib.h`.

5.21.2.2 `int` `ENV::mem_limit`

Definition at line 57 of file `glplib.h`.

5.21.2.3 `int` `ENV::mem_total`

Definition at line 60 of file `glplib.h`.

5.21.2.4 `int` `ENV::mem_tpeak`

Definition at line 63 of file `glplib.h`.

5.21.2.5 int ENV::mem_count

Definition at line 65 of file gplib.h.

5.21.2.6 int ENV::mem_cpeak

Definition at line 67 of file gplib.h.

The documentation for this struct was generated from the following file:

- [/home/ted/COIN/trunk/DyLP/src/DyLP/gplib.h](#)

5.22 hel_tag Struct Reference

```
#include <dylib_hash.h>
```

Public Attributes

- struct [hel_tag](#) * [next](#)
- const char * [key](#)
- void * [ent](#)

5.22.1 Detailed Description

Definition at line 37 of file dylib_hash.h.

5.22.2 Member Data Documentation

5.22.2.1 struct hel_tag* hel_tag::next

Definition at line 37 of file dylib_hash.h.

5.22.2.2 const char* hel_tag::key

Definition at line 38 of file dylib_hash.h.

5.22.2.3 void* hel_tag::ent

Definition at line 39 of file dylib_hash.h.

The documentation for this struct was generated from the following file:

- [/home/ted/COIN/trunk/DyLP/src/DyLPStdLib/dylib_hash.h](#)

5.23 INV Struct Reference

```
#include <glpinv.h>
```

Public Attributes

- int [m](#)

- int `valid`
- `LUF * luf`
- int `hh_max`
- int `hh_nfs`
- int * `hh_ndx`
- int * `hh_ptr`
- int * `hh_len`
- int * `p0_row`
- int * `p0_col`
- int `cc_len`
- int * `cc_ndx`
- double * `cc_val`
- double `upd_tol`
- int `nnz_h`
- double `min_vrratio`

5.23.1 Detailed Description

Definition at line 78 of file `glpinv.h`.

5.23.2 Member Data Documentation

5.23.2.1 int INV::m

Definition at line 80 of file `glpinv.h`.

5.23.2.2 int INV::valid

Definition at line 82 of file `glpinv.h`.

5.23.2.3 LUF* INV::luf

Definition at line 85 of file `glpinv.h`.

5.23.2.4 int INV::hh_max

Definition at line 89 of file `glpinv.h`.

5.23.2.5 int INV::hh_nfs

Definition at line 92 of file `glpinv.h`.

5.23.2.6 int* INV::hh_ndx

Definition at line 94 of file `glpinv.h`.

5.23.2.7 int* INV::hh_ptr

Definition at line 98 of file `glpinv.h`.

5.23.2.8 int* INV::hh_len

Definition at line 103 of file `glpinv.h`.

5.23.2.9 int* INV::p0_row

Definition at line 109 of file glpinv.h.

5.23.2.10 int* INV::p0_col

Definition at line 111 of file glpinv.h.

5.23.2.11 int INV::cc_len

Definition at line 120 of file glpinv.h.

5.23.2.12 int* INV::cc_ndx

Definition at line 123 of file glpinv.h.

5.23.2.13 double* INV::cc_val

Definition at line 127 of file glpinv.h.

5.23.2.14 double INV::upd_tol

Definition at line 133 of file glpinv.h.

5.23.2.15 int INV::nnz_h

Definition at line 146 of file glpinv.h.

5.23.2.16 double INV::min_vrratio

Definition at line 148 of file glpinv.h.

The documentation for this struct was generated from the following file:

- [/home/ted/COIN/trunk/DyLP/src/DyLP/glpinv.h](#)

5.24 keytab_entry_internal Struct Reference

```
#include <dylib_keytab.h>
```

Public Attributes

- const char * [keyword](#)
- int [min](#)
- int [token](#)

5.24.1 Detailed Description

Definition at line 33 of file dylib_keytab.h.

5.24.2 Member Data Documentation

5.24.2.1 const char* keytab_entry_internal::keyword

Definition at line 33 of file dylib_keytab.h.

5.24.2.2 int keytab_entry_internal::min

Definition at line 34 of file dylib_keytab.h.

5.24.2.3 int keytab_entry_internal::token

Definition at line 35 of file dylib_keytab.h.

The documentation for this struct was generated from the following file:

- [/home/ted/COIN/trunk/DyLP/src/DyLPStdLib/dylib_keytab.h](#)

5.25 lex_struct Struct Reference

```
#include <dylib_io.h>
```

Public Attributes

- char * [string](#)

5.25.1 Detailed Description

Definition at line 74 of file dylib_io.h.

5.25.2 Member Data Documentation

5.25.2.1 char* lex_struct::string

Definition at line 74 of file dylib_io.h.

The documentation for this struct was generated from the following file:

- [/home/ted/COIN/trunk/DyLP/src/DyLPStdLib/dylib_io.h](#)

5.26 Ink_struct_tag Struct Reference

```
#include <dylib_std.h>
```

Public Attributes

- struct [Ink_struct_tag](#) * [llnxt](#)
- void * [llval](#)

5.26.1 Detailed Description

Definition at line 115 of file dylib_std.h.

5.26.2 Member Data Documentation

5.26.2.1 struct `Ink_struct_tag*` `Ink_struct_tag::lInxt`

Definition at line 116 of file `dylib_std.h`.

5.26.2.2 `void*` `Ink_struct_tag::lIval`

Definition at line 117 of file `dylib_std.h`.

The documentation for this struct was generated from the following file:

- /home/ted/COIN/trunk/DyLP/src/DyLPStdLib/dylib_std.h

5.27 `lpopts_struct` Struct Reference

```
#include <dylp.h>
```

Public Attributes

- [cxtpe_enum](#) `context`
- `int` [scan](#)
- `int` [iterlim](#)
- `int` [idlelim](#)
- `struct` {
 - `int` [strat](#)
 - `bool` [flex](#)
 - `bool` [allownopiv](#)
- `dpsel`
- `struct` {
 - `int` [strat](#)
- `ppsel`
- `int` [factor](#)
- `int` [check](#)
- `int` [groom](#)
- `struct` {
 - `int` [actlvl](#)
 - `int` [actlim](#)
 - `int` [deactlvl](#)
- `con`
- `int` [addvar](#)
- `int` [dualadd](#)
- `int` [coldvars](#)
- `bool` [forcecold](#)
- `bool` [forcewarm](#)
- `bool` [usedual](#)
- `bool` [degen](#)
- `int` [degenpivlim](#)
- `int` [degenlite](#)

- [bool patch](#)
- [bool fullsys](#)
- [bool copyorigsys](#)
- [int scaling](#)
- [struct {
 \[float vars\]\(#\)
 \[float cons\]\(#\)
} active](#)
- [struct {
 \[double frac\]\(#\)
 \[bool i1lopen\]\(#\)
 \[double i1l\]\(#\)
 \[bool i1uopen\]\(#\)
 \[double i1u\]\(#\)
 \[bool i2valid\]\(#\)
 \[bool i2lopen\]\(#\)
 \[double i2l\]\(#\)
 \[bool i2uopen\]\(#\)
 \[double i2u\]\(#\)
} initcons](#)
- [ibtype_enum coldbasis](#)
- [struct {
 \[bool cons\]\(#\)
 \[bool vars\]\(#\)
} finpurge](#)
- [struct {
 \[bool d2p\]\(#\)
 \[bool p2d\]\(#\)
 \[bool flips\]\(#\)
} heroics](#)
- [struct {
 \[int major\]\(#\)
 \[int scaling\]\(#\)
 \[int setup\]\(#\)
 \[int crash\]\(#\)
 \[int pricing\]\(#\)
 \[int pivoting\]\(#\)
 \[int pivreject\]\(#\)
 \[int degen\]\(#\)
 \[int phase1\]\(#\)
 \[int phase2\]\(#\)
 \[int dual\]\(#\)
 \[int basis\]\(#\)
 \[int conmgmt\]\(#\)
 \[int varmgmt\]\(#\)
 \[int force\]\(#\)
 \[int tableau\]\(#\)
 \[int rays\]\(#\)
 \[int soln\]\(#\)
} print](#)

5.27.1 Detailed Description

Definition at line 1114 of file dylp.h.

5.27.2 Member Data Documentation

5.27.2.1 `cxtypes_enum` `lpopts_struct::context`

Definition at line 1115 of file dylp.h.

5.27.2.2 `int` `lpopts_struct::scan`

Definition at line 1116 of file dylp.h.

5.27.2.3 `int` `lpopts_struct::iterlim`

Definition at line 1117 of file dylp.h.

5.27.2.4 `int` `lpopts_struct::idlelim`

Definition at line 1118 of file dylp.h.

5.27.2.5 `int` `lpopts_struct::strat`

Definition at line 1119 of file dylp.h.

5.27.2.6 `bool` `lpopts_struct::flex`

Definition at line 1120 of file dylp.h.

5.27.2.7 `bool` `lpopts_struct::allownopiv`

Definition at line 1121 of file dylp.h.

5.27.2.8 `struct { ... }` `lpopts_struct::dpsel`

5.27.2.9 `struct { ... }` `lpopts_struct::ppsel`

5.27.2.10 `int` `lpopts_struct::factor`

Definition at line 1123 of file dylp.h.

5.27.2.11 `int` `lpopts_struct::check`

Definition at line 1124 of file dylp.h.

5.27.2.12 `int` `lpopts_struct::groom`

Definition at line 1125 of file dylp.h.

5.27.2.13 `int` `lpopts_struct::activl`

Definition at line 1126 of file dylp.h.

5.27.2.14 `int lpopts_struct::actlim`

Definition at line 1127 of file dylp.h.

5.27.2.15 `int lpopts_struct::deactivl`

Definition at line 1128 of file dylp.h.

5.27.2.16 `struct { ... } lpopts_struct::con`**5.27.2.17** `int lpopts_struct::addvar`

Definition at line 1129 of file dylp.h.

5.27.2.18 `int lpopts_struct::dualadd`

Definition at line 1130 of file dylp.h.

5.27.2.19 `int lpopts_struct::coldvars`

Definition at line 1131 of file dylp.h.

5.27.2.20 `bool lpopts_struct::forcecold`

Definition at line 1132 of file dylp.h.

5.27.2.21 `bool lpopts_struct::forcewarm`

Definition at line 1133 of file dylp.h.

5.27.2.22 `bool lpopts_struct::usedual`

Definition at line 1134 of file dylp.h.

5.27.2.23 `bool lpopts_struct::degen`

Definition at line 1135 of file dylp.h.

5.27.2.24 `int lpopts_struct::degenpivlim`

Definition at line 1136 of file dylp.h.

5.27.2.25 `int lpopts_struct::degenlite`

Definition at line 1137 of file dylp.h.

5.27.2.26 `bool lpopts_struct::patch`

Definition at line 1138 of file dylp.h.

5.27.2.27 `bool lpopts_struct::fullsys`

Definition at line 1139 of file dylp.h.

5.27.2.28 `bool lpopts_struct::copyorigsys`

Definition at line 1140 of file dylp.h.

5.27.2.29 int lpopts_struct::scaling

Definition at line 1141 of file dylp.h.

5.27.2.30 float lpopts_struct::vars

Definition at line 1142 of file dylp.h.

5.27.2.31 float lpopts_struct::cons

Definition at line 1143 of file dylp.h.

5.27.2.32 struct { ... } lpopts_struct::active**5.27.2.33 double lpopts_struct::frac**

Definition at line 1144 of file dylp.h.

5.27.2.34 bool lpopts_struct::i1lopen

Definition at line 1145 of file dylp.h.

5.27.2.35 double lpopts_struct::i1l

Definition at line 1146 of file dylp.h.

5.27.2.36 bool lpopts_struct::i1uopen

Definition at line 1147 of file dylp.h.

5.27.2.37 double lpopts_struct::i1u

Definition at line 1148 of file dylp.h.

5.27.2.38 bool lpopts_struct::i2valid

Definition at line 1149 of file dylp.h.

5.27.2.39 bool lpopts_struct::i2lopen

Definition at line 1150 of file dylp.h.

5.27.2.40 double lpopts_struct::i2l

Definition at line 1151 of file dylp.h.

5.27.2.41 bool lpopts_struct::i2uopen

Definition at line 1152 of file dylp.h.

5.27.2.42 double lpopts_struct::i2u

Definition at line 1153 of file dylp.h.

5.27.2.43 struct { ... } lpopts_struct::initcons

5.27.2.44 `ibtype_enum lpopts_struct::coldbasis`

Definition at line 1154 of file dylp.h.

5.27.2.45 `bool lpopts_struct::cons`

Definition at line 1155 of file dylp.h.

5.27.2.46 `bool lpopts_struct::vars`

Definition at line 1156 of file dylp.h.

5.27.2.47 `struct { ... } lpopts_struct::finpurge`**5.27.2.48** `bool lpopts_struct::d2p`

Definition at line 1157 of file dylp.h.

5.27.2.49 `bool lpopts_struct::p2d`

Definition at line 1158 of file dylp.h.

5.27.2.50 `bool lpopts_struct::flips`

Definition at line 1159 of file dylp.h.

5.27.2.51 `struct { ... } lpopts_struct::heroics`**5.27.2.52** `int lpopts_struct::major`

Definition at line 1160 of file dylp.h.

5.27.2.53 `int lpopts_struct::setup`

Definition at line 1162 of file dylp.h.

5.27.2.54 `int lpopts_struct::crash`

Definition at line 1163 of file dylp.h.

5.27.2.55 `int lpopts_struct::pricing`

Definition at line 1164 of file dylp.h.

5.27.2.56 `int lpopts_struct::pivoting`

Definition at line 1165 of file dylp.h.

5.27.2.57 `int lpopts_struct::pivreject`

Definition at line 1166 of file dylp.h.

5.27.2.58 `int lpopts_struct::degen`

Definition at line 1167 of file dylp.h.

5.27.2.59 `int lpopts_struct::phase1`

Definition at line 1168 of file `dylp.h`.

5.27.2.60 `int lpopts_struct::phase2`

Definition at line 1169 of file `dylp.h`.

5.27.2.61 `int lpopts_struct::dual`

Definition at line 1170 of file `dylp.h`.

5.27.2.62 `int lpopts_struct::basis`

Definition at line 1171 of file `dylp.h`.

5.27.2.63 `int lpopts_struct::conmgmt`

Definition at line 1172 of file `dylp.h`.

5.27.2.64 `int lpopts_struct::varmgmt`

Definition at line 1173 of file `dylp.h`.

5.27.2.65 `int lpopts_struct::force`

Definition at line 1174 of file `dylp.h`.

5.27.2.66 `int lpopts_struct::tableau`

Definition at line 1175 of file `dylp.h`.

5.27.2.67 `int lpopts_struct::rays`

Definition at line 1176 of file `dylp.h`.

5.27.2.68 `int lpopts_struct::soln`

Definition at line 1177 of file `dylp.h`.

5.27.2.69 `struct { ... } lpopts_struct::print`

The documentation for this struct was generated from the following file:

- `/home/ted/COIN/trunk/DyLP/src/Dylp/dylp.h`

5.28 `lpprob_struct` Struct Reference

```
#include <dylp.h>
```

Public Attributes

- `void *` `owner`
- `flags` `ctlopts`
- `dyphase_enum` `phase`

- [lpret_enum lpret](#)
- double [obj](#)
- int [iters](#)
- [consys_struct](#) * [consys](#)
- bool [fullsys](#)
- [basis_struct](#) * [basis](#)
- [flags](#) * [status](#)
- double * [x](#)
- double * [y](#)
- bool * [actvars](#)
- int [colsze](#)
- int [rowsze](#)

5.28.1 Detailed Description

Definition at line 586 of file dylp.h.

5.28.2 Member Data Documentation

5.28.2.1 void* lpprob_struct::owner

Definition at line 587 of file dylp.h.

5.28.2.2 flags lpprob_struct::ctlopts

Definition at line 588 of file dylp.h.

5.28.2.3 dyphase_enum lpprob_struct::phase

Definition at line 589 of file dylp.h.

5.28.2.4 lpret_enum lpprob_struct::lpret

Definition at line 590 of file dylp.h.

5.28.2.5 double lpprob_struct::obj

Definition at line 591 of file dylp.h.

5.28.2.6 int lpprob_struct::iters

Definition at line 592 of file dylp.h.

5.28.2.7 consys_struct* lpprob_struct::consys

Definition at line 593 of file dylp.h.

5.28.2.8 bool lpprob_struct::fullsys

Definition at line 594 of file dylp.h.

5.28.2.9 basis_struct* lpprob_struct::basis

Definition at line 595 of file dylp.h.

5.28.2.10 flags* lpprob_struct::status

Definition at line 596 of file dylp.h.

5.28.2.11 double* lpprob_struct::x

Definition at line 597 of file dylp.h.

5.28.2.12 double* lpprob_struct::y

Definition at line 598 of file dylp.h.

5.28.2.13 bool* lpprob_struct::actvars

Definition at line 599 of file dylp.h.

5.28.2.14 int lpprob_struct::colsize

Definition at line 600 of file dylp.h.

5.28.2.15 int lpprob_struct::rowse

Definition at line 601 of file dylp.h.

The documentation for this struct was generated from the following file:

- </home/ted/COIN/trunk/DyLP/src/Dylp/dylp.h>

5.29 lpstats_struct Struct Reference

```
#include <dylp.h>
```

Public Attributes

- int [phasecnts](#) [[dyDONE](#)+1]
- [dyphase_enum](#) [ini_simplex](#)
- struct {
 - int [sze](#)
 - double * [angle](#)
 - int * [actcnt](#)
 - int * [deactcnt](#)
 - bool * [init](#)
 - bool * [fin](#)
} [cons](#)
- struct {
 - int [sze](#)
 - int * [actcnt](#)
 - int * [deactcnt](#)
} [vars](#)
- struct {
 - float [max](#)
 - float [min](#)

- ```
 int hist [DYSTATS_HISTBINS]
} angle
```
- struct {  
 int cnt  
 int prevpiv  
 float avgpivs  
 int maxpivs  
} factor
  - struct {  
 int max  
 int mad  
 int sing  
 int pivtol\_red  
 double min\_pivtol  
 int puntcall  
 int puntret  
} pivrej
  - struct {  
 int flippable  
 int cnt  
 int cands  
 int promote  
 int nontrivial  
 int evals  
 int flips  
 int pivrnks  
 int maxrnk  
} dmulti
  - struct {  
 int cnt  
 int cands  
 int nontrivial  
 int promote  
} pmulti
  - struct {  
 int prevpiv  
 int maxcnt  
 int totpivs  
 int maxpivs  
 int chgcnt1  
 int chgcnt2  
} infeas
  - struct {  
 int cnt  
 float avgsiz  
 int maxsiz  
 int totpivs  
 float avgpivs

```

 int maxpivs
} pdegen [DYSTATS_MAXDEGEN]

• struct {
 int cnt
 float avgsiz
 int maxsiz
 int totpivs
 float avgpivs
 int maxpivs
} ddegen [DYSTATS_MAXDEGEN]

• struct {
 int iters
 int pivs
} tot

• struct {
 int iters
 int pivs
} p1

• struct {
 int iters
 int pivs
} p2

• struct {
 int iters
 int pivs
} d2

```

### 5.29.1 Detailed Description

Definition at line 1303 of file dylp.h.

### 5.29.2 Member Data Documentation

#### 5.29.2.1 int lpstats\_struct::phasecnts[dyDONE+1]

Definition at line 1304 of file dylp.h.

#### 5.29.2.2 dyphase\_enum lpstats\_struct::ini\_simplex

Definition at line 1305 of file dylp.h.

#### 5.29.2.3 int lpstats\_struct::size

Definition at line 1306 of file dylp.h.

#### 5.29.2.4 double\* lpstats\_struct::angle

Definition at line 1307 of file dylp.h.

**5.29.2.5 int\* lpstats\_struct::actcnt**

Definition at line 1308 of file dylp.h.

**5.29.2.6 int\* lpstats\_struct::deactcnt**

Definition at line 1309 of file dylp.h.

**5.29.2.7 bool\* lpstats\_struct::init**

Definition at line 1310 of file dylp.h.

**5.29.2.8 bool\* lpstats\_struct::fin**

Definition at line 1311 of file dylp.h.

**5.29.2.9 struct { ... } lpstats\_struct::cons****5.29.2.10 struct { ... } lpstats\_struct::vars****5.29.2.11 float lpstats\_struct::max**

Definition at line 1315 of file dylp.h.

**5.29.2.12 float lpstats\_struct::min**

Definition at line 1316 of file dylp.h.

**5.29.2.13 int lpstats\_struct::hist[DYSTATS\_HISTBINS]**

Definition at line 1317 of file dylp.h.

**5.29.2.14 struct { ... } lpstats\_struct::angle****5.29.2.15 int lpstats\_struct::cnt**

Definition at line 1318 of file dylp.h.

**5.29.2.16 int lpstats\_struct::prevpiv**

Definition at line 1319 of file dylp.h.

**5.29.2.17 float lpstats\_struct::avgpivs**

Definition at line 1320 of file dylp.h.

**5.29.2.18 int lpstats\_struct::maxpivs**

Definition at line 1321 of file dylp.h.

**5.29.2.19 struct { ... } lpstats\_struct::factor****5.29.2.20 int lpstats\_struct::max**

Definition at line 1322 of file dylp.h.

#### 5.29.2.21 `int lpstats_struct::mad`

Definition at line 1323 of file dylp.h.

#### 5.29.2.22 `int lpstats_struct::sing`

Definition at line 1324 of file dylp.h.

#### 5.29.2.23 `int lpstats_struct::pivotl_red`

Definition at line 1325 of file dylp.h.

#### 5.29.2.24 `double lpstats_struct::min_pivotl`

Definition at line 1326 of file dylp.h.

#### 5.29.2.25 `int lpstats_struct::puntcall`

Definition at line 1327 of file dylp.h.

#### 5.29.2.26 `int lpstats_struct::puntret`

Definition at line 1328 of file dylp.h.

#### 5.29.2.27 `struct { ... } lpstats_struct::pivrej`

#### 5.29.2.28 `int lpstats_struct::flippable`

Definition at line 1329 of file dylp.h.

#### 5.29.2.29 `int lpstats_struct::cands`

Definition at line 1331 of file dylp.h.

#### 5.29.2.30 `int lpstats_struct::promote`

Definition at line 1332 of file dylp.h.

#### 5.29.2.31 `int lpstats_struct::nontrivial`

Definition at line 1333 of file dylp.h.

#### 5.29.2.32 `int lpstats_struct::evals`

Definition at line 1334 of file dylp.h.

#### 5.29.2.33 `int lpstats_struct::flips`

Definition at line 1335 of file dylp.h.

#### 5.29.2.34 `int lpstats_struct::pivrnks`

Definition at line 1336 of file dylp.h.

#### 5.29.2.35 `int lpstats_struct::maxrnk`

Definition at line 1337 of file dylp.h.



5.29.2.36 struct { ... } lpstats\_struct::dmulti

5.29.2.37 struct { ... } lpstats\_struct::pmulti

5.29.2.38 int lpstats\_struct::maxcnt

Definition at line 1343 of file dylp.h.

5.29.2.39 int lpstats\_struct::totpivs

Definition at line 1344 of file dylp.h.

5.29.2.40 int lpstats\_struct::chgcnt1

Definition at line 1346 of file dylp.h.

5.29.2.41 int lpstats\_struct::chgcnt2

Definition at line 1347 of file dylp.h.

5.29.2.42 struct { ... } lpstats\_struct::infeas

5.29.2.43 float lpstats\_struct::avgsiz

Definition at line 1349 of file dylp.h.

5.29.2.44 int lpstats\_struct::maxsiz

Definition at line 1350 of file dylp.h.

5.29.2.45 struct { ... } lpstats\_struct::pdegen[DYSTATS\_MAXDEGEN]

5.29.2.46 struct { ... } lpstats\_struct::ddegen[DYSTATS\_MAXDEGEN]

5.29.2.47 int lpstats\_struct::iters

Definition at line 1360 of file dylp.h.

5.29.2.48 int lpstats\_struct::pivs

Definition at line 1361 of file dylp.h.

5.29.2.49 struct { ... } lpstats\_struct::tot

5.29.2.50 struct { ... } lpstats\_struct::p1

5.29.2.51 struct { ... } lpstats\_struct::p2

5.29.2.52 struct { ... } lpstats\_struct::d2

The documentation for this struct was generated from the following file:

- [/home/ted/COIN/trunk/DyLP/src/DyLP/dylp.h](#)

## 5.30 Iptols\_struct Struct Reference

```
#include <dylp.h>
```

## Public Attributes

- double [inf](#)
- double [zero](#)
- double [pchk](#)
- double [pfeas](#)
- double [pfeas\\_scale](#)
- double [cost](#)
- double [dchk](#)
- double [dfeas](#)
- double [dfeas\\_scale](#)
- double [pivot](#)
- double [bogus](#)
- double [swing](#)
- double [toobig](#)
- double [purge](#)
- double [purgevar](#)
- double [reframe](#)

### 5.30.1 Detailed Description

Definition at line 666 of file dylp.h.

### 5.30.2 Member Data Documentation

#### 5.30.2.1 double `lptols_struct::inf`

Definition at line 667 of file dylp.h.

#### 5.30.2.2 double `lptols_struct::zero`

Definition at line 668 of file dylp.h.

#### 5.30.2.3 double `lptols_struct::pchk`

Definition at line 669 of file dylp.h.

#### 5.30.2.4 double `lptols_struct::pfeas`

Definition at line 670 of file dylp.h.

#### 5.30.2.5 double `lptols_struct::pfeas_scale`

Definition at line 671 of file dylp.h.

#### 5.30.2.6 double `lptols_struct::cost`

Definition at line 672 of file dylp.h.

#### 5.30.2.7 double `lptols_struct::dchk`

Definition at line 673 of file dylp.h.

**5.30.2.8 double lptols\_struct::dfeas**

Definition at line 674 of file dylp.h.

**5.30.2.9 double lptols\_struct::dfeas\_scale**

Definition at line 675 of file dylp.h.

**5.30.2.10 double lptols\_struct::pivot**

Definition at line 676 of file dylp.h.

**5.30.2.11 double lptols\_struct::bogus**

Definition at line 677 of file dylp.h.

**5.30.2.12 double lptols\_struct::swing**

Definition at line 678 of file dylp.h.

**5.30.2.13 double lptols\_struct::toobig**

Definition at line 679 of file dylp.h.

**5.30.2.14 double lptols\_struct::purge**

Definition at line 680 of file dylp.h.

**5.30.2.15 double lptols\_struct::purgevar**

Definition at line 681 of file dylp.h.

**5.30.2.16 double lptols\_struct::reframe**

Definition at line 682 of file dylp.h.

The documentation for this struct was generated from the following file:

- </home/ted/COIN/trunk/DyLP/src/Dylp/dylp.h>

**5.31 LUF Struct Reference**

```
#include <glpluf.h>
```

**Public Attributes**

- int [n](#)
- int [valid](#)
- int \* [fr\\_ptr](#)
- int \* [fr\\_len](#)
- int \* [fc\\_ptr](#)
- int \* [fc\\_len](#)
- int \* [vr\\_ptr](#)
- int \* [vr\\_len](#)
- int \* [vr\\_cap](#)

- double \* [vr\\_piv](#)
- int \* [vc\\_ptr](#)
- int \* [vc\\_len](#)
- int \* [vc\\_cap](#)
- int \* [pp\\_row](#)
- int \* [pp\\_col](#)
- int \* [qq\\_row](#)
- int \* [qq\\_col](#)
- int [sv\\_size](#)
- int [sv\\_beg](#)
- int [sv\\_end](#)
- int \* [sv\\_ndx](#)
- double \* [sv\\_val](#)
- int [sv\\_head](#)
- int [sv\\_tail](#)
- int \* [sv\\_prev](#)
- int \* [sv\\_next](#)
- int \* [flag](#)
- double \* [work](#)
- int [new\\_sva](#)
- double [piv\\_tol](#)
- int [piv\\_lim](#)
- int [suhl](#)
- double [eps\\_tol](#)
- double [max\\_gro](#)
- int [nnz\\_a](#)
- int [nnz\\_f](#)
- int [nnz\\_v](#)
- double [max\\_a](#)
- double [big\\_v](#)
- int [rank](#)

#### 5.31.1 Detailed Description

Definition at line 83 of file `glpluf.h`.

#### 5.31.2 Member Data Documentation

##### 5.31.2.1 `int LUF::n`

Definition at line 85 of file `glpluf.h`.

##### 5.31.2.2 `int LUF::valid`

Definition at line 87 of file `glpluf.h`.

##### 5.31.2.3 `int* LUF::fr_ptr`

Definition at line 91 of file `glpluf.h`.

**5.31.2.4 int\* LUF::fr\_len**

Definition at line 95 of file glpluf.h.

**5.31.2.5 int\* LUF::fc\_ptr**

Definition at line 101 of file glpluf.h.

**5.31.2.6 int\* LUF::fc\_len**

Definition at line 105 of file glpluf.h.

**5.31.2.7 int\* LUF::vr\_ptr**

Definition at line 111 of file glpluf.h.

**5.31.2.8 int\* LUF::vr\_len**

Definition at line 115 of file glpluf.h.

**5.31.2.9 int\* LUF::vr\_cap**

Definition at line 119 of file glpluf.h.

**5.31.2.10 double\* LUF::vr\_piv**

Definition at line 124 of file glpluf.h.

**5.31.2.11 int\* LUF::vc\_ptr**

Definition at line 130 of file glpluf.h.

**5.31.2.12 int\* LUF::vc\_len**

Definition at line 134 of file glpluf.h.

**5.31.2.13 int\* LUF::vc\_cap**

Definition at line 138 of file glpluf.h.

**5.31.2.14 int\* LUF::pp\_row**

Definition at line 145 of file glpluf.h.

**5.31.2.15 int\* LUF::pp\_col**

Definition at line 147 of file glpluf.h.

**5.31.2.16 int\* LUF::qq\_row**

Definition at line 155 of file glpluf.h.

**5.31.2.17 int\* LUF::qq\_col**

Definition at line 157 of file glpluf.h.

**5.31.2.18 int LUF::sv\_size**

Definition at line 174 of file glpluf.h.

**5.31.2.19 int LUF::sv\_beg**

Definition at line 179 of file glpluf.h.

**5.31.2.20 int LUF::sv\_end**

Definition at line 179 of file glpluf.h.

**5.31.2.21 int\* LUF::sv\_ndx**

Definition at line 186 of file glpluf.h.

**5.31.2.22 double\* LUF::sv\_val**

Definition at line 190 of file glpluf.h.

**5.31.2.23 int LUF::sv\_head**

Definition at line 200 of file glpluf.h.

**5.31.2.24 int LUF::sv\_tail**

Definition at line 202 of file glpluf.h.

**5.31.2.25 int\* LUF::sv\_prev**

Definition at line 204 of file glpluf.h.

**5.31.2.26 int\* LUF::sv\_next**

Definition at line 207 of file glpluf.h.

**5.31.2.27 int\* LUF::flag**

Definition at line 212 of file glpluf.h.

**5.31.2.28 double\* LUF::work**

Definition at line 214 of file glpluf.h.

**5.31.2.29 int LUF::new\_sva**

Definition at line 218 of file glpluf.h.

**5.31.2.30 double LUF::piv\_tol**

Definition at line 221 of file glpluf.h.

**5.31.2.31 int LUF::piv\_lim**

Definition at line 229 of file glpluf.h.

#### 5.31.2.32 int LUF::suhl

Definition at line 233 of file `glpluf.h`.

#### 5.31.2.33 double LUF::eps\_tol

Definition at line 241 of file `glpluf.h`.

#### 5.31.2.34 double LUF::max\_gro

Definition at line 244 of file `glpluf.h`.

#### 5.31.2.35 int LUF::nnz\_a

Definition at line 252 of file `glpluf.h`.

#### 5.31.2.36 int LUF::nnz\_f

Definition at line 254 of file `glpluf.h`.

#### 5.31.2.37 int LUF::nnz\_v

Definition at line 257 of file `glpluf.h`.

#### 5.31.2.38 double LUF::max\_a

Definition at line 261 of file `glpluf.h`.

#### 5.31.2.39 double LUF::big\_v

Definition at line 263 of file `glpluf.h`.

#### 5.31.2.40 int LUF::rank

Definition at line 266 of file `glpluf.h`.

The documentation for this struct was generated from the following file:

- `/home/ted/COIN/trunk/DyLP/src/DyLP/glpluf.h`

## 5.32 LUF\_WA Struct Reference

```
#include <glpluf.h>
```

### Public Attributes

- double \* `rs_max`
- int \* `rs_head`
- int \* `rs_prev`
- int \* `rs_next`
- int \* `cs_head`
- int \* `cs_prev`
- int \* `cs_next`

### 5.32.1 Detailed Description

Definition at line 270 of file glpluf.h.

### 5.32.2 Member Data Documentation

#### 5.32.2.1 double\* LUF\_WA::rs\_max

Definition at line 272 of file glpluf.h.

#### 5.32.2.2 int\* LUF\_WA::rs\_head

Definition at line 286 of file glpluf.h.

#### 5.32.2.3 int\* LUF\_WA::rs\_prev

Definition at line 289 of file glpluf.h.

#### 5.32.2.4 int\* LUF\_WA::rs\_next

Definition at line 293 of file glpluf.h.

#### 5.32.2.5 int\* LUF\_WA::cs\_head

Definition at line 297 of file glpluf.h.

#### 5.32.2.6 int\* LUF\_WA::cs\_prev

Definition at line 300 of file glpluf.h.

#### 5.32.2.7 int\* LUF\_WA::cs\_next

Definition at line 304 of file glpluf.h.

The documentation for this struct was generated from the following file:

- [/home/ted/COIN/trunk/DyLP/src/DyLP/glpluf.h](#)

## 5.33 MEM Struct Reference

```
#include <glplib.h>
```

### Public Attributes

- int [size](#)
- int [flag](#)
- MEM \* [prev](#)
- MEM \* [next](#)

### 5.33.1 Detailed Description

Definition at line 105 of file glplib.h.



## 5.33.2 Member Data Documentation

## 5.33.2.1 int MEM::size

Definition at line 107 of file glplib.h.

## 5.33.2.2 int MEM::flag

Definition at line 109 of file glplib.h.

## 5.33.2.3 MEM\* MEM::prev

Definition at line 111 of file glplib.h.

## 5.33.2.4 MEM\* MEM::next

Definition at line 113 of file glplib.h.

The documentation for this struct was generated from the following file:

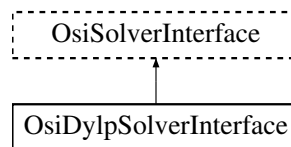
- [/home/ted/COIN/trunk/DyLP/src/DyLP/glplib.h](#)

## 5.34 OsiDyLpSolverInterface Class Reference

COIN OSI API for dylp.

```
#include <OsiDyLpSolverInterface.hpp>
```

Inheritance diagram for OsiDyLpSolverInterface:



## Public Member Functions

## Constructors and Destructors

- [OsiDyLpSolverInterface](#) ()  
*Default constructor.*
- [OsiDyLpSolverInterface](#) (const [OsiDyLpSolverInterface](#) &src)  
*Copy constructor.*
- [OsiSolverInterface](#) \* [clone](#) (bool copyData=true) const  
*Clone the solver object.*
- [OsiDyLpSolverInterface](#) & [operator=](#) (const [OsiDyLpSolverInterface](#) &rhs)  
*Assignment.*
- [~OsiDyLpSolverInterface](#) ()  
*Destructor.*
- void [reset](#) ()  
*Reset the solver object to the state produced by the default constructor.*

## Methods to load a problem

- int `readMps` (const char \*filename, const char \*extension="mps")  
*Read a problem description in MPS format from a file.*
- int `readMps` (const char \*filename, const char \*extension, int &numberSets, CoinSet \*\*&sets)  
*Read a problem description in MPS format from a file, including SOS information.*
- void `writeMps` (const char \*basename, const char \*extension="mps", double objsense=0.0) const  
*Write the problem into the specified file in MPS format.*
- void `loadProblem` (const CoinPackedMatrix &matrix, const double \*collb, const double \*colub, const double \*obj, const char \*rowSen, const double \*rowrhs, const double \*rowrng)  
*Load a problem description (OSI packed matrix, row sense, parameters unaffected).*
- void `loadProblem` (const CoinPackedMatrix &matrix, const double \*collb, const double \*colub, const double \*obj, const double \*rowlb, const double \*rowub)  
*Load a problem description (OSI packed matrix, row bounds, parameters unaffected).*
- void `loadProblem` (const int colcnt, const int rowcnt, const int \*start, const int \*index, const double \*value, const double \*collb, const double \*colub, const double \*obj, const char \*sense, const double \*rhsin, const double \*range)  
*Load a problem description (standard column-major packed matrix, row sense, parameters unaffected)*
- void `loadProblem` (const int colcnt, const int rowcnt, const int \*start, const int \*index, const double \*value, const double \*collb, const double \*colub, const double \*obj, const double \*row\_lower, const double \*row\_upper)  
*Load a problem description (standard column-major packed matrix, row bounds, parameters unaffected)*
- void `assignProblem` (CoinPackedMatrix \*&matrix, double \*&collb, double \*&colub, double \*&obj, char \*&rowSen, double \*&rowrhs, double \*&rowrng)  
*Load a problem description (OSI packed matrix, row sense, parameters destroyed).*
- void `assignProblem` (CoinPackedMatrix \*&matrix, double \*&collb, double \*&colub, double \*&obj, double \*&rowlb, double \*&rowub)  
*Load a problem description (OSI packed matrix, row bounds, parameters destroyed).*

### Methods to obtain problem information

- int `getNumCols` () const  
*Get the number of columns (variables)*
- int `getNumRows` () const  
*Get the number of rows (constraints)*
- int `getNumElements` () const  
*Get the number of non-zero coefficients.*
- int `getNumIntegers` () const  
*Get the number of integer variables.*
- const double \* `getColLower` () const  
*Get the column (variable) lower bound vector.*
- const double \* `getColUpper` () const  
*Get the column (variable) upper bound vector.*
- bool `isContinuous` (int colIndex) const  
*Return true if the variable is continuous.*
- bool `isBinary` (int colIndex) const  
*Return true if the variable is binary.*
- bool `isIntegerNonBinary` (int colIndex) const  
*Return true if the variable is general integer.*
- bool `isInteger` (int colIndex) const  
*Return true if the variable is integer (general or binary)*
- const char \* `getRowSense` () const  
*Get the row sense (constraint type) vector.*
- const double \* `getRightHandSide` () const  
*Get the row (constraint) right-hand-side vector.*
- const double \* `getRowRange` () const  
*Get the row (constraint) range vector.*
- const double \* `getRowLower` () const

- *Get the row (constraint) lower bound vector.*  
const double \* [getRowUpper](#) () const
- *Get the row (constraint) upper bound vector.*  
const double \* [getObjCoefficients](#) () const
- *Get the objective function coefficient vector.*  
double [getObjSense](#) () const
- *Get the objective function sense (min/max)*  
const CoinPackedMatrix \* [getMatrixByRow](#) () const
- *Get a pointer to a row-major copy of the constraint matrix.*  
const CoinPackedMatrix \* [getMatrixByCol](#) () const
- *Get a pointer to a column-major copy of the constraint matrix.*

### Methods for row and column names.

Only the set methods need to be overridden to ensure consistent names between OsiDyIp and the OSI base class.

- void [setObjName](#) (std::string name)  
*Set the objective function name.*
- void [setRowName](#) (int ndx, std::string name)  
*Set a row name.*
- void [setColName](#) (int ndx, std::string name)  
*Set a column name.*

### Methods to modify the problem

- void [setContinuous](#) (int index)  
*Set a single variable to be continuous.*
- void [setInteger](#) (int index)  
*Set a single variable to be integer.*
- void [setColLower](#) (int index, double value)  
*Set the lower bound on a column (variable)*
- void [setColUpper](#) (int index, double value)  
*Set the upper bound on a column (variable)*
- void [setRowLower](#) (int index, double value)  
*Set the lower bound on a row (constraint)*
- void [setRowUpper](#) (int index, double value)  
*Set the upper bound on a row (constraint)*
- void [setRowType](#) (int index, char rowsen, double rowrhs, double rowrng)  
*Set the type of a row (constraint)*
- void [setObjCoeff](#) (int index, double value)  
*Set an objective function coefficient.*
- void [setObjective](#) (const double \*array)  
*Set the objective coefficients for all columns.*
- void [setObjSense](#) (double sense)  
*Set the sense (min/max) of the objective.*
- void [setColSolution](#) (const double \*colsol)  
*Set the value of the primal variables in the problem solution.*
- void [setRowPrice](#) (const double \*)  
*Set the value of the dual variables in the problem solution.*
- void [addCol](#) (const CoinPackedVectorBase &vec, const double collb, const double colub, const double obj)  
*Add a column (variable) to the problem.*
- void [deleteCols](#) (const int num, const int \*colIndices)  
*Remove column(s) (variable(s)) from the problem.*
- void [addRow](#) (const CoinPackedVectorBase &row, const double rowlb, const double rowub)  
*Add a row (constraint) to the problem.*

- void `addRow` (const CoinPackedVectorBase &row, const char rowsen, const double rowrhs, const double rowrng)  
*Add a row (constraint) to the problem.*
- void `deleteRows` (const int num, const int \*rowIndices)  
*Delete row(s) (constraint(s)) from the problem.*
- void `applyRowCut` (const OsiRowCut &cut)  
*Apply a row (constraint) cut (add one constraint)*
- void `applyColCut` (const OsiColCut &cut)  
*Apply a column (variable) cut (adjust one or more bounds)*

### Solve methods

- void `initialSolve` ()  
*Solve an lp from scratch.*
- CoinWarmStart \* `getEmptyWarmStart` () const  
*Get an empty `OsiDylpWarmStartBasis` object.*
- CoinWarmStart \* `getWarmStart` () const  
*Build a warm start object for the current lp solution.*
- bool `setWarmStart` (const CoinWarmStart \*warmStart)  
*Apply a warm start object.*
- void `resolve` ()  
*Call dylp to reoptimize (warm start).*
- void `markHotStart` ()  
*Create a hot start snapshot.*
- void `solveFromHotStart` ()  
*Call dylp to reoptimize (hot start).*
- void `unmarkHotStart` ()  
*Delete the hot start snapshot.*

### Methods returning solver termination status

- bool `isAbandoned` () const  
*True if dylp abandoned the problem.*
- bool `isProvenOptimal` () const  
*True if dylp reported an optimal solution.*
- bool `isProvenPrimalInfeasible` () const  
*True if dylp reported the problem to be primal infeasible.*
- bool `isProvenDualInfeasible` () const  
*True if dylp reported the problem to be dual infeasible (primal unbounded)*
- bool `isIterationLimitReached` () const  
*True if dylp reached the iteration limit.*
- int `getIterationCount` () const  
*Get the number of iterations for the last lp.*
- bool `isPrimalObjectiveLimitReached` () const  
*Is the primal objective limit reached?*
- bool `isDualObjectiveLimitReached` () const  
*Is the dual objective limit reached?*

### Methods to set/get solver parameters

- double `getInfinity` () const  
*Get dylp's value for infinity.*
- bool `setIntParam` (OsiIntParam key, int value)  
*Set an OSI integer parameter.*

- [bool setDbiParam](#) (OsiDbiParam key, double value)  
*Set an OSI double parameter.*
- [bool setStrParam](#) (OsiStrParam key, const std::string &value)  
*Set an OSI string parameter.*
- [bool setHintParam](#) (OsiHintParam key, [bool](#) sense=true, OsiHintStrength strength=OsiHintTry, void \*info=0)  
*Set an OSI hint.*
- [bool getIntParam](#) (OsiIntParam key, int &value) const  
*Get an OSI integer parameter.*
- [bool getDbiParam](#) (OsiDbiParam key, double &value) const  
*Get an OSI double parameter.*
- [bool getStrParam](#) (OsiStrParam key, std::string &value) const  
*Get an OSI string parameter.*
- [bool getHintParam](#) (OsiHintParam key, [bool](#) &sense, OsiHintStrength &strength, void \*&info) const  
*Get an OSI hint.*
- void [newLanguage](#) (CoinMessages::Language language)  
*Change the language for OsiDyIpsolver messages.*
- void [setLanguage](#) (CoinMessages::Language language)  
*An alias for [OsiDyIpsolverInterface::newLanguage](#).*

#### Methods to obtain solution information

- double [getObjValue](#) () const  
*Get the objective function value for the solution.*
- const double \* [getColSolution](#) () const  
*Return the vector of primal variables for the solution.*
- const double \* [getRowPrice](#) () const  
*Return the vector of dual variables for the solution.*
- const double \* [getReducedCost](#) () const  
*Return the vector of reduced costs for the solution.*
- const double \* [getRowActivity](#) () const  
*Return the vector of row activity for the solution.*
- std::vector< double \* > [getDualRays](#) (int maxNumRays, [bool](#) fullRay) const  
*Get as many dual rays as the solver can provide.*
- std::vector< double \* > [getPrimalRays](#) (int maxNumRays) const  
*Get as many primal rays as the solver can provide.*

#### Simplex API methods

- int [canDoSimplexInterface](#) () const  
*Return the simplex implementation level.*
- void [enableFactorization](#) () const  
*Prepare the solver for the use of tableau access methods.*
- void [disableFactorization](#) () const  
*Undo the effects of [enableFactorization](#).*
- [bool](#) [basisIsAvailable](#) () const  
*Check if an optimal basis is available.*
- void [getBasisStatus](#) (int \*archStatus, int \*logStatus) const  
*Retrieve status information for architectural and logical variables.*
- int [setBasisStatus](#) (const int \*archStatus, const int \*logStatus)  
*Set a basis and update the factorization and solution.*
- virtual void [getReducedGradient](#) (double \*columnReducedCosts, double \*duals, const double \*c) const  
*Calculate duals and reduced costs for the given objective coefficients.*
- virtual void [getBasics](#) (int \*index) const  
*Get indices of basic variables.*

- virtual void `getBlvCol` (int col, double \*betak) const  
*Get a column of the basis inverse.*
- virtual void `getBlvACol` (int col, double \*abarj) const  
*Get a column of the tableau.*
- virtual void `getBlvRow` (int row, double \*betai) const  
*Get a row of the basis inverse.*
- virtual void `getBlvARow` (int row, double \*abari, double \*betai=0) const  
*Get a row of the tableau.*

### Debugging Methods

- void `activateRowCutDebugger` (const char \*modelName)  
*Activate the row cut debugger.*
- void `activateRowCutDebugger` (const double \*solution, bool keepContinuous=false)  
*Activate the row cut debugger.*

### DyLP-specific methods

- void `dylp_controlfile` (const char \*name, const bool silent, const bool mustexist=true)  
*Process an options (.spc) file.*
- void `dylp_logfile` (const char \*name, bool echo=false)  
*Establish a log file.*
- void `dylp_outfile` (const char \*name)  
*Establish an output (solution and/or statistics) file.*
- void `dylp_printsoln` (bool wantSoln, bool wantStats)  
*Print the solution and/or statistics to the output file.*
- void `setOsiDyLPMessages` (CoinMessages::Language local\_language)  
*Set the language for messages.*

### Unsupported functions

- void `branchAndBound` ()  
*Invoke the solver's built-in branch-and-bound algorithm.*

### Friends

- void `OsiDyLPsolverInterfaceUnitTest` (const std::string &mpsDir, const std::string &netLibDir)  
*Unit test for `OsiDyLPsolverInterface`.*

### DyLP data structures

These fields hold pointers to the data structures which are used to pass an lp problem to dylp.

- `lpopts_struct * initialSolveOptions`  
*Solver options for an initial solve.*
- `lpopts_struct * resolveOptions`  
*Solver options for a resolve.*
- `lptols_struct * tolerances`  
*Solver numeric tolerances.*

### 5.34.1 Detailed Description

COIN OSI API for dylp.

The class [OsiDyIpSolverInterface](#) (ODSI) implements the public functions defined for the COIN OsiSolverInterface (OSI) API.

#### [OsiDyIpSolverInterface](#) Principles for Users

In addition to the principles outlined for the OsiSolverInterface class, ODSI maintains the following:

**Construction of a Constraint System:** A constraint system can be batch loaded from a file (MPS format) or from a data structure, or it can be built incrementally. When building a constraint system incrementally, keep in mind that you must create a row or column (addRow or addCol, respectively) before you can adjust other properties (row or column bounds, objective, variable values, etc.)

**Existence of a Solution:** For proper operation, OSI requires that a SI maintain a basic primal solution at all times after a problem has been loaded.

When a problem is loaded, ODSI generates a basic primal solution (primal variable values and a matching basis). The solution is not necessarily primal or dual feasible. In terms of the objective function, this solution is pessimistic, but not necessarily worst-case. ODSI does not generate matching values for the dual variables (row prices).

Any successful call to dylp (*i.e.*, a call that results in an optimal, infeasible, or unbounded result, or that terminates on iteration limit) will replace the existing solution with the result of the call to dylp.

It is possible to specify initial values for the primal and dual variables using [setColSolution\(\)](#) and [setRowPrice\(\)](#). To specify an initial basis, see the documentation for the CoinWarmStartBasis and [OsiDyIpWarmStartBasis](#) classes. When these functions are used, it is the responsibility of the client to ensure validity and consistency.

**Maintenance of an LP Basis** Skirting the edges of the principle that changing the problem invalidates the solution, OsiDyIp will maintain a valid basis across two common operations used in branch-and-cut: deletion of a loose constraint and deletion of a nonbasic variable. Arguably the set of allowable modifications could be increased.

**Assignment** Assignment ([operator=\(\)](#)) works pretty much as you'd expect, with one exception. Only one ODSI object can control the dylp solver at a time, so hot start information is not copied on assignment.

Detailed implementation comments are contained in OsiDyIpSolverInterface.cpp, which is not normally scanned when generating COIN OSI API documentation.

Definition at line 107 of file OsiDyIpSolverInterface.hpp.

### 5.34.2 Constructor & Destructor Documentation

#### 5.34.2.1 [OsiDyIpSolverInterface::OsiDyIpSolverInterface \( \)](#)

Default constructor.

#### 5.34.2.2 [OsiDyIpSolverInterface::OsiDyIpSolverInterface \( const \[OsiDyIpSolverInterface\]\(#\) & src \)](#)

Copy constructor.

#### 5.34.2.3 [OsiDyIpSolverInterface::~~OsiDyIpSolverInterface \( \)](#)

Destructor.

### 5.34.3 Member Function Documentation

5.34.3.1 `OsiSolverInterface* OsiDyLpSolverInterface::clone ( bool copyData = true ) const`

Clone the solver object.

5.34.3.2 `OsiDyLpSolverInterface& OsiDyLpSolverInterface::operator= ( const OsiDyLpSolverInterface & rhs )`

Assignment.

5.34.3.3 `void OsiDyLpSolverInterface::reset ( )`

Reset the solver object to the state produced by the default constructor.

5.34.3.4 `int OsiDyLpSolverInterface::readMps ( const char * filename, const char * extension = "mps" )`

Read a problem description in MPS format from a file.

5.34.3.5 `int OsiDyLpSolverInterface::readMps ( const char * filename, const char * extension, int & numberSets, CoinSet **& sets )`

Read a problem description in MPS format from a file, including SOS information.

5.34.3.6 `void OsiDyLpSolverInterface::writeMps ( const char * basename, const char * extension = "mps", double objsense = 0.0 ) const`

Write the problem into the specified file in MPS format.

*objsense* == 1 forces the file to be written as a maximisation problem, while -1 forces a minimisation problem. The default of 0 writes the file as maximisation or minimisation using the solver's current setting.

5.34.3.7 `void OsiDyLpSolverInterface::loadProblem ( const CoinPackedMatrix & matrix, const double * collb, const double * colub, const double * obj, const char * rowsen, const double * rowrhs, const double * rowrng )`

Load a problem description (OSI packed matrix, row sense, parameters unaffected).

5.34.3.8 `void OsiDyLpSolverInterface::loadProblem ( const CoinPackedMatrix & matrix, const double * collb, const double * colub, const double * obj, const double * rowlb, const double * rowub )`

Load a problem description (OSI packed matrix, row bounds, parameters unaffected).

5.34.3.9 `void OsiDyLpSolverInterface::loadProblem ( const int colcnt, const int rowcnt, const int * start, const int * index, const double * value, const double * collb, const double * colub, const double * obj, const char * sense, const double * rhsin, const double * range )`

Load a problem description (standard column-major packed matrix, row sense, parameters unaffected)

5.34.3.10 `void OsiDyLpSolverInterface::loadProblem ( const int colcnt, const int rowcnt, const int * start, const int * index, const double * value, const double * collb, const double * colub, const double * obj, const double * row_lower, const double * row_upper )`

Load a problem description (standard column-major packed matrix, row bounds, parameters unaffected)

5.34.3.11 `void OsiDyLpSolverInterface::assignProblem ( CoinPackedMatrix *& matrix, double *& collb, double *& colub, double *& obj, char *& rowsen, double *& rowrhs, double *& rowrng )`

Load a problem description (OSI packed matrix, row sense, parameters destroyed).



5.34.3.12 void OsiDyIpSolverInterface::assignProblem ( CoinPackedMatrix \*& *matrix*, double \*& *collb*, double \*& *colub*, double \*& *obj*, double \*& *rowlb*, double \*& *rowub* )

Load a problem description (OSI packed matrix, row bounds, parameters destroyed).

5.34.3.13 int OsiDyIpSolverInterface::getNumCols ( ) const

Get the number of columns (variables)

5.34.3.14 int OsiDyIpSolverInterface::getNumRows ( ) const

Get the number of rows (constraints)

5.34.3.15 int OsiDyIpSolverInterface::getNumElements ( ) const

Get the number of non-zero coefficients.

5.34.3.16 int OsiDyIpSolverInterface::getNumIntegers ( ) const

Get the number of integer variables.

Counts both binary and general integer variables.

5.34.3.17 const double\* OsiDyIpSolverInterface::getColLower ( ) const

Get the column (variable) lower bound vector.

5.34.3.18 const double\* OsiDyIpSolverInterface::getColUpper ( ) const

Get the column (variable) upper bound vector.

5.34.3.19 bool OsiDyIpSolverInterface::isContinuous ( int *colIndex* ) const

Return true if the variable is continuous.

5.34.3.20 bool OsiDyIpSolverInterface::isBinary ( int *colIndex* ) const

Return true if the variable is binary.

5.34.3.21 bool OsiDyIpSolverInterface::isIntegerNonBinary ( int *colIndex* ) const

Return true if the variable is general integer.

5.34.3.22 bool OsiDyIpSolverInterface::isInteger ( int *colIndex* ) const

Return true if the variable is integer (general or binary)

5.34.3.23 const char\* OsiDyIpSolverInterface::getRowSense ( ) const

Get the row sense (constraint type) vector.

5.34.3.24 const double\* OsiDyIpSolverInterface::getRightHandSide ( ) const

Get the row (constraint) right-hand-side vector.

5.34.3.25 const double\* OsiDyIpSolverInterface::getRowRange ( ) const

Get the row (constraint) range vector.

**5.34.3.26** `const double* OsiDyLpSolverInterface::getRowLower ( ) const`

Get the row (constraint) lower bound vector.

**5.34.3.27** `const double* OsiDyLpSolverInterface::getRowUpper ( ) const`

Get the row (constraint) upper bound vector.

**5.34.3.28** `const double* OsiDyLpSolverInterface::getObjCoefficients ( ) const`

Get the objective function coefficient vector.

**5.34.3.29** `double OsiDyLpSolverInterface::getObjSense ( ) const`

Get the objective function sense (min/max)

A value of 1 indicates minimisation; -1 indicates maximisation.

**5.34.3.30** `const CoinPackedMatrix* OsiDyLpSolverInterface::getMatrixByRow ( ) const`

Get a pointer to a row-major copy of the constraint matrix.

**5.34.3.31** `const CoinPackedMatrix* OsiDyLpSolverInterface::getMatrixByCol ( ) const`

Get a pointer to a column-major copy of the constraint matrix.

**5.34.3.32** `void OsiDyLpSolverInterface::setObjName ( std::string name )`

Set the objective function name.

**5.34.3.33** `void OsiDyLpSolverInterface::setRowName ( int ndx, std::string name )`

Set a row name.

Quietly does nothing if the name discipline (#OsiNameDiscipline) is auto. Quietly fails if the row index is invalid.

**5.34.3.34** `void OsiDyLpSolverInterface::setColName ( int ndx, std::string name )`

Set a column name.

Quietly does nothing if the name discipline (#OsiNameDiscipline) is auto. Quietly fails if the column index is invalid.

**5.34.3.35** `void OsiDyLpSolverInterface::setContinuous ( int index )`

Set a single variable to be continuous.

**5.34.3.36** `void OsiDyLpSolverInterface::setInteger ( int index )`

Set a single variable to be integer.

**5.34.3.37** `void OsiDyLpSolverInterface::setColLower ( int index, double value )`

Set the lower bound on a column (variable)

**5.34.3.38** `void OsiDyLpSolverInterface::setColUpper ( int index, double value )`

Set the upper bound on a column (variable)

5.34.3.39 void OsiDyIpSolverInterface::setRowLower ( int *index*, double *value* )

Set the lower bound on a row (constraint)

5.34.3.40 void OsiDyIpSolverInterface::setRowUpper ( int *index*, double *value* )

Set the upper bound on a row (constraint)

5.34.3.41 void OsiDyIpSolverInterface::setRowType ( int *index*, char *rowsen*, double *rowrhs*, double *rowrng* )

Set the type of a row (constraint)

5.34.3.42 void OsiDyIpSolverInterface::setObjCoeff ( int *index*, double *value* )

Set an objective function coefficient.

5.34.3.43 void OsiDyIpSolverInterface::setObjective ( const double \* *array* )

Set the objective coefficients for all columns.

5.34.3.44 void OsiDyIpSolverInterface::setObjSense ( double *sense* )

Set the sense (min/max) of the objective.

Use 1 for minimisation, -1 for maximisation. (The default is minimisation; the objective is multiplied by -1 to maximise.)

5.34.3.45 void OsiDyIpSolverInterface::setColSolution ( const double \* *colsol* )

Set the value of the primal variables in the problem solution.

5.34.3.46 void OsiDyIpSolverInterface::setRowPrice ( const double \* )

Set the value of the dual variables in the problem solution.

5.34.3.47 void OsiDyIpSolverInterface::addCol ( const CoinPackedVectorBase & *vec*, const double *collb*, const double *colub*, const double *obj* )

Add a column (variable) to the problem.

5.34.3.48 void OsiDyIpSolverInterface::deleteCols ( const int *num*, const int \* *colIndices* )

Remove column(s) (variable(s)) from the problem.

5.34.3.49 void OsiDyIpSolverInterface::addRow ( const CoinPackedVectorBase & *row*, const double *rowlb*, const double *rowub* )

Add a row (constraint) to the problem.

5.34.3.50 void OsiDyIpSolverInterface::addRow ( const CoinPackedVectorBase & *row*, const char *rowsen*, const double *rowrhs*, const double *rowrng* )

Add a row (constraint) to the problem.

5.34.3.51 void OsiDyIpSolverInterface::deleteRows ( const int *num*, const int \* *rowIndices* )

Delete row(s) (constraint(s)) from the problem.

5.34.3.52 `void OsiDyIpSolverInterface::applyRowCut ( const OsiRowCut & cut )`

Apply a row (constraint) cut (add one constraint)

5.34.3.53 `void OsiDyIpSolverInterface::applyColCut ( const OsiColCut & cut )`

Apply a column (variable) cut (adjust one or more bounds)

5.34.3.54 `void OsiDyIpSolverInterface::initialSolve ( )`

Solve an lp from scratch.

5.34.3.55 `CoinWarmStart* OsiDyIpSolverInterface::getEmptyWarmStart ( ) const`

Get an empty [OsiDyIpWarmStartBasis](#) object.

5.34.3.56 `CoinWarmStart* OsiDyIpSolverInterface::getWarmStart ( ) const`

Build a warm start object for the current lp solution.

5.34.3.57 `bool OsiDyIpSolverInterface::setWarmStart ( const CoinWarmStart * warmStart )`

Apply a warm start object.

By definition, a null parameter is a request to synch the warm start basis with the solver. ODSI interprets a 0x0 basis as a request to remove warm start information.

5.34.3.58 `void OsiDyIpSolverInterface::resolve ( )`

Call dylp to reoptimize (warm start).

5.34.3.59 `void OsiDyIpSolverInterface::markHotStart ( )`

Create a hot start snapshot.

5.34.3.60 `void OsiDyIpSolverInterface::solveFromHotStart ( )`

Call dylp to reoptimize (hot start).

5.34.3.61 `void OsiDyIpSolverInterface::unmarkHotStart ( )`

Delete the hot start snapshot.

5.34.3.62 `bool OsiDyIpSolverInterface::isAbandoned ( ) const`

True if dylp abandoned the problem.

5.34.3.63 `bool OsiDyIpSolverInterface::isProvenOptimal ( ) const`

True if dylp reported an optimal solution.

5.34.3.64 `bool OsiDyIpSolverInterface::isProvenPrimalInfeasible ( ) const`

True if dylp reported the problem to be primal infeasible.

5.34.3.65 `bool OsiDyIpSolverInterface::isProvenDualInfeasible ( ) const`

True if dylp reported the problem to be dual infeasible (primal unbounded)

**5.34.3.66** `bool OsiDyIpSolverInterface::isIterationLimitReached ( ) const`

True if dylp reached the iteration limit.

**5.34.3.67** `int OsiDyIpSolverInterface::getIterationCount ( ) const`

Get the number of iterations for the last lp.

**5.34.3.68** `bool OsiDyIpSolverInterface::isPrimalObjectiveLimitReached ( ) const`

Is the primal objective limit reached?

Put in different terms, quit when the objective value becomes better than the given limit for an acceptable value.

**5.34.3.69** `bool OsiDyIpSolverInterface::isDualObjectiveLimitReached ( ) const`

Is the dual objective limit reached?

Put in different terms, quit when the objective value becomes worse than the given limit for an acceptable value.

**5.34.3.70** `double OsiDyIpSolverInterface::getInfinity ( ) const`

Get dylp's value for infinity.

**5.34.3.71** `bool OsiDyIpSolverInterface::setIntParam ( OsiIntParam key, int value )`

Set an OSI integer parameter.

**5.34.3.72** `bool OsiDyIpSolverInterface::setDbiParam ( OsiDbiParam key, double value )`

Set an OSI double parameter.

**5.34.3.73** `bool OsiDyIpSolverInterface::setStrParam ( OsiStrParam key, const std::string & value )`

Set an OSI string parameter.

**5.34.3.74** `bool OsiDyIpSolverInterface::setHintParam ( OsiHintParam key, bool sense = true, OsiHintStrength strength = OsiHintTry, void * info = 0 )`

Set an OSI hint.

**5.34.3.75** `bool OsiDyIpSolverInterface::getIntParam ( OsiIntParam key, int & value ) const`

Get an OSI integer parameter.

**5.34.3.76** `bool OsiDyIpSolverInterface::getDbiParam ( OsiDbiParam key, double & value ) const`

Get an OSI double parameter.

**5.34.3.77** `bool OsiDyIpSolverInterface::getStrParam ( OsiStrParam key, std::string & value ) const`

Get an OSI string parameter.

**5.34.3.78** `bool OsiDyIpSolverInterface::getHintParam ( OsiHintParam key, bool & sense, OsiHintStrength & strength, void *& info ) const`

Get an OSI hint.

**5.34.3.79** `void OsiDyIpSolverInterface::newLanguage ( CoinMessages::Language language ) [inline]`

Change the language for OsiDyIp messages.

Definition at line 569 of file OsiDyIpSolverInterface.hpp.

**5.34.3.80** `void OsiDyIpSolverInterface::setLanguage ( CoinMessages::Language language ) [inline]`

An alias for [OsiDyIpSolverInterface::newLanguage](#).

Definition at line 574 of file OsiDyIpSolverInterface.hpp.

**5.34.3.81** `double OsiDyIpSolverInterface::getObjValue ( ) const`

Get the objective function value for the solution.

**5.34.3.82** `const double* OsiDyIpSolverInterface::getColSolution ( ) const`

Return the vector of primal variables for the solution.

**5.34.3.83** `const double* OsiDyIpSolverInterface::getRowPrice ( ) const`

Return the vector of dual variables for the solution.

**5.34.3.84** `const double* OsiDyIpSolverInterface::getReducedCost ( ) const`

Return the vector of reduced costs for the solution.

**5.34.3.85** `const double* OsiDyIpSolverInterface::getRowActivity ( ) const`

Return the vector of row activity for the solution.

**5.34.3.86** `std::vector<double*> OsiDyIpSolverInterface::getDualRays ( int maxNumRays, bool fullRay ) const`

Get as many dual rays as the solver can provide.

If `fullRay` is false (the default), the ray will contain only the components associated with the row duals. If `fullRay` is set to `true`, the ray will also contain the components associated with nonbasic variables.

**5.34.3.87** `std::vector<double*> OsiDyIpSolverInterface::getPrimalRays ( int maxNumRays ) const`

Get as many primal rays as the solver can provide.

**5.34.3.88** `int OsiDyIpSolverInterface::canDoSimplexInterface ( ) const`

Return the simplex implementation level.

**5.34.3.89** `void OsiDyIpSolverInterface::enableFactorization ( ) const`

Prepare the solver for the use of tableau access methods.

In order for the tableau methods to work, the ODSI object invoking them must own the solver; the most recent call to optimise the problem must have resulted in an optimal solution; and the solver must be holding retained data structures for that optimal solution. It's much more efficient if the solver is using the full system, but it's not mandatory.

Because this is a const method, we can't force any of this; we can only check.

**5.34.3.90** `void OsiDyIpSolverInterface::disableFactorization ( ) const`

Undo the effects of [enableFactorization](#).

Even if `resolve` was invoked by `enableFactorization`, little needs to be done here. Ownership of the solver is transferred by invocation, so there's no need to explicitly give it back.

**5.34.3.91** `bool OsiDyIpSolverInterface::basisIsAvailable ( ) const`

Check if an optimal basis is available.

**5.34.3.92** `void OsiDyIpSolverInterface::getBasisStatus ( int * archStatus, int * logStatus ) const`

Retrieve status information for architectural and logical variables.

Retrieve status vectors for architectural (also called structural or column) and logical (also called artificial or row) variables. Returns the same information as `getWarmStart`, but in a different format.

**5.34.3.93** `int OsiDyIpSolverInterface::setBasisStatus ( const int * archStatus, const int * logStatus )`

Set a basis and update the factorization and solution.

Provides the combined functionality of `setWarmStart` followed by `resolve`. As with `getBasisStatus`, the status vectors are coded as integers.

**5.34.3.94** `virtual void OsiDyIpSolverInterface::getReducedGradient ( double * columnReducedCosts, double * duals, const double * c ) const [virtual]`

Calculate duals and reduced costs for the given objective coefficients.

The solver's objective coefficient vector is not changed (cf. `#setObjectiveAndRefresh`)

**5.34.3.95** `virtual void OsiDyIpSolverInterface::getBasics ( int * index ) const [virtual]`

Get indices of basic variables.

**5.34.3.96** `virtual void OsiDyIpSolverInterface::getBlvCol ( int col, double * betak ) const [virtual]`

Get a column of the basis inverse.

**5.34.3.97** `virtual void OsiDyIpSolverInterface::getBlvACol ( int col, double * abarj ) const [virtual]`

Get a column of the tableau.

**5.34.3.98** `virtual void OsiDyIpSolverInterface::getBlvRow ( int row, double * betai ) const [virtual]`

Get a row of the basis inverse.

**5.34.3.99** `virtual void OsiDyIpSolverInterface::getBlvARow ( int row, double * abari, double * betai = 0 ) const [virtual]`

Get a row of the tableau.

**5.34.3.100** `void OsiDyIpSolverInterface::activateRowCutDebugger ( const char * modelName )`

Activate the row cut debugger.

Activate the debugger for a model known to the debugger. The debugger will consult an internal database for an optimal solution vector.

**5.34.3.101** `void OsiDyIpSolverInterface::activateRowCutDebugger ( const double * solution, bool keepContinuous = false )`

Activate the row cut debugger.

Activate the debugger for a model not included in the debugger's internal database. `solution` must be a full solution

vector, but only the integer variables need to be correct. The debugger will fill in the continuous variables by solving an lp relaxation with the integer variables fixed as specified. If the given values for the continuous variables should be preserved, set `keepContinuous` to true.

**5.34.3.102** `void OsiDyLpSolverInterface::dylp_controlfile ( const char * name, const bool silent, const bool mustexist = true )`

Process an options (.spc) file.

**5.34.3.103** `void OsiDyLpSolverInterface::dylp_logfile ( const char * name, bool echo = false )`

Establish a log file.

**5.34.3.104** `void OsiDyLpSolverInterface::dylp_outfile ( const char * name )`

Establish an output (solution and/or statistics) file.

**5.34.3.105** `void OsiDyLpSolverInterface::dylp_printsoln ( bool wantSoln, bool wantStats )`

Print the solution and/or statistics to the output file.

**5.34.3.106** `void OsiDyLpSolverInterface::setOsiDyLpMessages ( CoinMessages::Language local_language )`

Set the language for messages.

**5.34.3.107** `void OsiDyLpSolverInterface::branchAndBound ( )`

Invoke the solver's built-in branch-and-bound algorithm.

#### 5.34.4 Friends And Related Function Documentation

**5.34.4.1** `void OsiDyLpSolverInterfaceUnitTest ( const std::string & mpsDir, const std::string & netLibDir ) [friend]`

Unit test for [OsiDyLpSolverInterface](#).

Performs various tests to see if ODSI is functioning correctly. Not an exhaustive test, but it'll (usually) catch gross problems.

#### 5.34.5 Member Data Documentation

**5.34.5.1** `Ipopts_struct* OsiDyLpSolverInterface::initialSolveOptions`

Solver options for an initial solve.

Definition at line 778 of file `OsiDyLpSolverInterface.hpp`.

**5.34.5.2** `Ipopts_struct* OsiDyLpSolverInterface::resolveOptions`

Solver options for a resolve.

Definition at line 781 of file `OsiDyLpSolverInterface.hpp`.

**5.34.5.3** `Iptols_struct* OsiDyLpSolverInterface::tolerances`

Solver numeric tolerances.

Definition at line 784 of file `OsiDyLpSolverInterface.hpp`.



## 5.34.5.4 CoinWarmStart\* OsiDyIpSolverInterface::basis

Definition at line 939 of file OsiDyIpSolverInterface.hpp.

## 5.34.5.5 basisCondition OsiDyIpSolverInterface::condition

Definition at line 940 of file OsiDyIpSolverInterface.hpp.

## 5.34.5.6 int OsiDyIpSolverInterface::balance

Definition at line 941 of file OsiDyIpSolverInterface.hpp.

## 5.34.5.7 int OsiDyIpSolverInterface::simplex

Definition at line 957 of file OsiDyIpSolverInterface.hpp.

## 5.34.5.8 bool OsiDyIpSolverInterface::saved\_fullsys

Definition at line 958 of file OsiDyIpSolverInterface.hpp.

The documentation for this class was generated from the following file:

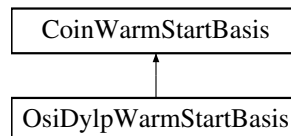
- /home/ted/COIN/trunk/DyLP/src/OsiDyIp/[OsiDyIpSolverInterface.hpp](#)

## 5.35 OsiDyIpWarmStartBasis Class Reference

The dylp warm start class.

```
#include <OsiDyIpWarmStartBasis.hpp>
```

Inheritance diagram for OsiDyIpWarmStartBasis:



## Public Member Functions

**Methods to get and set basis information.**

*Methods for structural and artificial variables are inherited from CoinWarmStartBasis.*

*Constraint status is coded using the CoinWarmStartBasis::Status codes. Active constraints are coded as atLower-Bound, inactive as isFree.*

- int [numberActiveConstraints](#) () const  
*Return the number of active constraints.*
- Status [getConStatus](#) (int i) const  
*Return the status of the specified constraint.*
- void [setConStatus](#) (int i, Status st)  
*Set the status of the specified constraint.*
- char \* [getConstraintStatus](#) ()  
*Return the status array for constraints.*
- const char \* [getConstraintStatus](#) () const

- *const overload for [getConstraintStatus\(\)](#)*
- void [setPhase](#) ([dyphase\\_enum](#) phase)  
*Set the lp phase for this basis.*
- [dyphase\\_enum](#) [getPhase](#) () const  
*Get the lp phase for this basis.*

### Basis 'diff' methods

- CoinWarmStartDiff \* [generateDiff](#) (const CoinWarmStart \*const oldCWS) const  
*Generate a 'diff' that can convert oldBasis to this basis.*
- void [applyDiff](#) (const CoinWarmStartDiff \*const cwsdDiff)  
*Apply diff to this basis.*

### Methods to modify the warm start object

- void [setSize](#) (int ns, int na)  
*Set basis capacity; existing basis is discarded.*
- void [resize](#) (int numRows, int numCols)  
*Set basis capacity; existing basis is maintained.*
- void [compressRows](#) (int tgtCnt, const int \*tgts)  
*Delete a set of rows from the basis.*
- void [deleteRows](#) (int number, const int \*which)  
*Delete a set of rows from the basis.*
- virtual void [mergeBasis](#) (const CoinWarmStartBasis \*src, const XferVec \*xferRows, const XferVec \*xferCols)  
*Merge entries from a source basis into this basis.*

### Constructors, destructors, and related functions

- [OsiDyLpWarmStartBasis](#) ()  
*Default constructor (empty object)*
- [OsiDyLpWarmStartBasis](#) (int ns, int na, const char \*sStat, const char \*aStat, const char \*cStat=0)  
*Constructs a warm start object with the specified status arrays.*
- [OsiDyLpWarmStartBasis](#) (const CoinWarmStartBasis &cwsb)  
*Construct an [OsiDyLpWarmStartBasis](#) from a [CoinWarmStartBasis](#).*
- [OsiDyLpWarmStartBasis](#) (const [OsiDyLpWarmStartBasis](#) &ws)  
*Copy constructor.*
- CoinWarmStart \* [clone](#) () const  
*'Virtual constructor'*
- [~OsiDyLpWarmStartBasis](#) ()  
*Destructor.*
- [OsiDyLpWarmStartBasis](#) & [operator=](#) (const [OsiDyLpWarmStartBasis](#) &rhs)  
*Assignment.*
- void [assignBasisStatus](#) (int ns, int na, char \*&sStat, char \*&aStat, char \*&cStat)  
*Assign the status vectors to be the warm start information.*
- void [assignBasisStatus](#) (int ns, int na, char \*&sStat, char \*&aStat)  
*Assign the status vectors to be the warm start information.*

### Miscellaneous methods

- void [print](#) () const  
*Prints in readable format (for debug)*
- void [checkBasis](#) (CoinMessageHandler \*msghandler=NULL) const  
*Performs basis consistency checks (for debug)*

## 5.35.1 Detailed Description

The dylp warm start class.

This derived class is necessary because dylp by default works with a subset of the full constraint system. The warm start object needs to contain a list of the active constraints in addition to the status information included in CoinWarmStartBasis. It is also convenient to include the solver phase in the warm start object.

Definition at line 44 of file OsiDyIpWarmStartBasis.hpp.

## 5.35.2 Constructor &amp; Destructor Documentation

## 5.35.2.1 OsiDyIpWarmStartBasis::OsiDyIpWarmStartBasis ( )

Default constructor (empty object)

5.35.2.2 OsiDyIpWarmStartBasis::OsiDyIpWarmStartBasis ( int *ns*, int *na*, const char \* *sStat*, const char \* *aStat*, const char \* *cStat* = 0 )

Constructs a warm start object with the specified status arrays.

5.35.2.3 OsiDyIpWarmStartBasis::OsiDyIpWarmStartBasis ( const CoinWarmStartBasis & *cwsb* )

Construct an [OsiDyIpWarmStartBasis](#) from a CoinWarmStartBasis.

5.35.2.4 OsiDyIpWarmStartBasis::OsiDyIpWarmStartBasis ( const OsiDyIpWarmStartBasis & *ws* )

Copy constructor.

## 5.35.2.5 OsiDyIpWarmStartBasis::~~OsiDyIpWarmStartBasis ( )

Destructor.

## 5.35.3 Member Function Documentation

## 5.35.3.1 int OsiDyIpWarmStartBasis::numberActiveConstraints ( ) const

Return the number of active constraints.

5.35.3.2 Status OsiDyIpWarmStartBasis::getConStatus ( int *i* ) const [inline]

Return the status of the specified constraint.

Definition at line 64 of file OsiDyIpWarmStartBasis.hpp.

5.35.3.3 void OsiDyIpWarmStartBasis::setConStatus ( int *i*, Status *st* ) [inline]

Set the status of the specified constraint.

Definition at line 72 of file OsiDyIpWarmStartBasis.hpp.

## 5.35.3.4 char\* OsiDyIpWarmStartBasis::getConstraintStatus ( ) [inline]

Return the status array for constraints.

Definition at line 81 of file OsiDyIpWarmStartBasis.hpp.

**5.35.3.5** `const char* OsiDyIpWarmStartBasis::getConstraintStatus ( ) const` `[inline]`

`const` overload for [getConstraintStatus\(\)](#)

Definition at line 89 of file `OsiDyIpWarmStartBasis.hpp`.

**5.35.3.6** `void OsiDyIpWarmStartBasis::setPhase ( dyphase_enum phase )` `[inline]`

Set the lp phase for this basis.

Definition at line 96 of file `OsiDyIpWarmStartBasis.hpp`.

**5.35.3.7** `dyphase_enum OsiDyIpWarmStartBasis::getPhase ( ) const` `[inline]`

Get the lp phase for this basis.

Definition at line 100 of file `OsiDyIpWarmStartBasis.hpp`.

**5.35.3.8** `CoinWarmStartDiff* OsiDyIpWarmStartBasis::generateDiff ( const CoinWarmStart *const oldCWS ) const`

Generate a 'diff' that can convert oldBasis to this basis.

**5.35.3.9** `void OsiDyIpWarmStartBasis::applyDiff ( const CoinWarmStartDiff *const cwsdDiff )`

Apply `diff` to this basis.

**5.35.3.10** `void OsiDyIpWarmStartBasis::setSize ( int ns, int na )`

Set basis capacity; existing basis is discarded.

**5.35.3.11** `void OsiDyIpWarmStartBasis::resize ( int numRows, int numCols )`

Set basis capacity; existing basis is maintained.

**5.35.3.12** `void OsiDyIpWarmStartBasis::compressRows ( int tgtCnt, const int * tgts )`

Delete a set of rows from the basis.

#### Warning

This routine assumes that the set of indices to be deleted is sorted in ascending order and is free from duplicates. Use `deleteRows` if this is not guaranteed.

The resulting basis is guaranteed valid only if all deleted constraints are slack (hence the associated logicals are basic).

**5.35.3.13** `void OsiDyIpWarmStartBasis::deleteRows ( int number, const int * which )`

Delete a set of rows from the basis.

#### Warning

The resulting basis is guaranteed valid only if all deleted constraints are slack (hence the associated logicals are basic).

**5.35.3.14** `virtual void OsiDyIpWarmStartBasis::mergeBasis ( const CoinWarmStartBasis * src, const XferVec * xferRows, const XferVec * xferCols )` `[virtual]`

Merge entries from a source basis into this basis.

**Warning**

It's the client's responsibility to ensure validity of the merged basis, if that's important to the application.

The vector `xferCols` (`xferRows`) specifies runs of entries to be taken from the source basis and placed in this basis. Each entry is a `CoinTriple`, with first specifying the starting source index of a run, second specifying the starting destination index, and third specifying the run length.

**5.35.3.15** `CoinWarmStart* OsiDyIpWarmStartBasis::clone ( ) const`

'Virtual constructor'

**5.35.3.16** `OsiDyIpWarmStartBasis& OsiDyIpWarmStartBasis::operator= ( const OsiDyIpWarmStartBasis & rhs )`

Assignment.

**5.35.3.17** `void OsiDyIpWarmStartBasis::assignBasisStatus ( int ns, int na, char *& sStat, char *& aStat, char *& cStat )`

Assign the status vectors to be the warm start information.

**5.35.3.18** `void OsiDyIpWarmStartBasis::assignBasisStatus ( int ns, int na, char *& sStat, char *& aStat )`

Assign the status vectors to be the warm start information.

**5.35.3.19** `void OsiDyIpWarmStartBasis::print ( ) const`

Prints in readable format (for debug)

**5.35.3.20** `void OsiDyIpWarmStartBasis::checkBasis ( CoinMessageHandler * msghandler = NULL ) const`

Performs basis consistency checks (for debug)

The documentation for this class was generated from the following file:

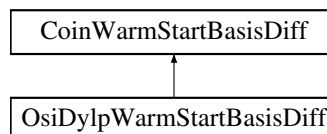
- </home/ted/COIN/trunk/DyLP/src/OsiDyIp/OsiDyIpWarmStartBasis.hpp>

**5.36 OsiDyIpWarmStartBasisDiff Class Reference**

A 'diff' between two [OsiDyIpWarmStartBasis](#) objects.

```
#include <OsiDyIpWarmStartBasis.hpp>
```

Inheritance diagram for `OsiDyIpWarmStartBasisDiff`:

**Public Member Functions**

- virtual `CoinWarmStartDiff * clone ( ) const`  
'Virtual constructor'
- virtual `OsiDyIpWarmStartBasisDiff & operator= (const OsiDyIpWarmStartBasisDiff &rhs)`

*Assignment.*

- virtual `~OsiDyIpWarmStartBasisDiff()`

*Destructor.*

## Friends

- `CoinWarmStartDiff * OsiDyIpWarmStartBasis::generateDiff (const CoinWarmStart *const oldCWS) const`
- void `OsiDyIpWarmStartBasis::applyDiff (const CoinWarmStartDiff *const diff)`

### 5.36.1 Detailed Description

A ‘diff’ between two `OsiDyIpWarmStartBasis` objects.

This class exists in order to hide from the world the details of calculating and representing a ‘diff’ between two `OsiDyIpWarmStartBasis` objects. For convenience, assignment, cloning, and deletion are visible to the world, and default and copy constructors are visible to derived classes. Knowledge of the rest of this structure, and of generating and applying diffs, is restricted to the functions `OsiDyIpWarmStartBasis::generateDiff()` and `OsiDyIpWarmStartBasis::applyDiff()`.

The actual data structure is a pair of unsigned int vectors, `#diffNdxs_` and `#diffVals_`, and a `CoinWarmStartBasisDiff` object.

**Todo** This is a pretty generic structure, and vector diff is a pretty generic activity. We should be able to convert this to a template.

**Todo** Using unsigned int as the data type for the diff vectors might help to contain the damage when this code is inevitably compiled for 64 bit architectures. But the notion of int as 4 bytes is hardwired into `CoinWarmStartBasis`, so changes are definitely required.

Definition at line 266 of file `OsiDyIpWarmStartBasis.hpp`.

### 5.36.2 Constructor & Destructor Documentation

5.36.2.1 `virtual OsiDyIpWarmStartBasisDiff::~~OsiDyIpWarmStartBasisDiff ( ) [inline],[virtual]`

Destructor.

Definition at line 279 of file `OsiDyIpWarmStartBasis.hpp`.

### 5.36.3 Member Function Documentation

5.36.3.1 `virtual CoinWarmStartDiff* OsiDyIpWarmStartBasisDiff::clone ( ) const [inline],[virtual]`

‘Virtual constructor’

Definition at line 270 of file `OsiDyIpWarmStartBasis.hpp`.

5.36.3.2 `virtual OsiDyIpWarmStartBasisDiff& OsiDyIpWarmStartBasisDiff::operator= ( const OsiDyIpWarmStartBasisDiff & rhs ) [virtual]`

Assignment.

#### 5.36.4 Friends And Related Function Documentation

5.36.4.1 `CoinWarmStartDiff* OsiDyIpWarmStartBasis::generateDiff ( const CoinWarmStart *const oldCWS ) const`  
[friend]

5.36.4.2 `void OsiDyIpWarmStartBasis::applyDiff ( const CoinWarmStartDiff *const diff )` [friend]

The documentation for this class was generated from the following file:

- [/home/ted/COIN/trunk/DyLP/src/OsiDyIp/OsiDyIpWarmStartBasis.hpp](#)

## 5.37 parse\_any Union Reference

```
#include <dylib_bnfrdr.h>
```

### Public Attributes

- void \* [g](#)
- char \* [c](#)

#### 5.37.1 Detailed Description

Definition at line 718 of file `dylib_bnfrdr.h`.

#### 5.37.2 Member Data Documentation

5.37.2.1 `void* parse_any::g`

Definition at line 718 of file `dylib_bnfrdr.h`.

5.37.2.2 `char* parse_any::c`

Definition at line 719 of file `dylib_bnfrdr.h`.

The documentation for this union was generated from the following file:

- [/home/ted/COIN/trunk/DyLP/src/DyIpStdLib/dylib\\_bnfrdr.h](#)

## 5.38 pkcoeff\_struct Struct Reference

```
#include <dy_vector.h>
```

### Public Attributes

- int [ndx](#)
- double [val](#)

#### 5.38.1 Detailed Description

Definition at line 238 of file `dy_vector.h`.

### 5.38.2 Member Data Documentation

#### 5.38.2.1 int pkcoeff\_struct::ndx

Definition at line 238 of file dy\_vector.h.

#### 5.38.2.2 double pkcoeff\_struct::val

Definition at line 239 of file dy\_vector.h.

The documentation for this struct was generated from the following file:

- /home/ted/COIN/trunk/DyLP/src/DyLP/dy\_vector.h

## 5.39 pkvec\_struct Struct Reference

```
#include <dy_vector.h>
```

### Public Attributes

- int [ndx](#)
- const char \* [nme](#)
- int [dim](#)
- double [dflt](#)
- int [cnt](#)
- int [size](#)
- [pkcoeff\\_struct](#) \* [coeffs](#)

### 5.39.1 Detailed Description

Definition at line 241 of file dy\_vector.h.

### 5.39.2 Member Data Documentation

#### 5.39.2.1 int pkvec\_struct::ndx

Definition at line 241 of file dy\_vector.h.

#### 5.39.2.2 const char\* pkvec\_struct::nme

Definition at line 242 of file dy\_vector.h.

#### 5.39.2.3 int pkvec\_struct::dim

Definition at line 243 of file dy\_vector.h.

#### 5.39.2.4 double pkvec\_struct::dflt

Definition at line 244 of file dy\_vector.h.

#### 5.39.2.5 int pkvec\_struct::cnt

Definition at line 245 of file dy\_vector.h.



#### 5.39.2.6 int pkvec\_struct::size

Definition at line 246 of file dy\_vector.h.

#### 5.39.2.7 pkcoeff\_struct\* pkvec\_struct::coeffs

Definition at line 247 of file dy\_vector.h.

The documentation for this struct was generated from the following file:

- [/home/ted/COIN/trunk/DyLP/src/DyLP/dy\\_vector.h](#)

## 5.40 POOL Struct Reference

```
#include <glplib.h>
```

### Public Attributes

- int [size](#)
- void \* [avail](#)
- void \* [link](#)
- int [used](#)
- void \* [stock](#)
- int [count](#)

#### 5.40.1 Detailed Description

Definition at line 130 of file glplib.h.

#### 5.40.2 Member Data Documentation

##### 5.40.2.1 int POOL::size

Definition at line 132 of file glplib.h.

##### 5.40.2.2 void\* POOL::avail

Definition at line 135 of file glplib.h.

##### 5.40.2.3 void\* POOL::link

Definition at line 137 of file glplib.h.

##### 5.40.2.4 int POOL::used

Definition at line 140 of file glplib.h.

##### 5.40.2.5 void\* POOL::stock

Definition at line 142 of file glplib.h.

#### 5.40.2.6 int POOL::count

Definition at line 144 of file `gplib.h`.

The documentation for this struct was generated from the following file:

- [/home/ted/COIN/trunk/DyLP/src/DyLP/gplib.h](#)

### 5.41 rowhdr\_struct\_tag Struct Reference

```
#include <dy_consys.h>
```

#### Public Attributes

- int [ndx](#)
- int [len](#)
- const char \* [nme](#)
- [coeff\\_struct](#) \* [coeffs](#)

#### 5.41.1 Detailed Description

Definition at line 137 of file `dy_consys.h`.

#### 5.41.2 Member Data Documentation

##### 5.41.2.1 int rowhdr\_struct\_tag::ndx

Definition at line 138 of file `dy_consys.h`.

##### 5.41.2.2 int rowhdr\_struct\_tag::len

Definition at line 139 of file `dy_consys.h`.

##### 5.41.2.3 const char\* rowhdr\_struct\_tag::nme

Definition at line 140 of file `dy_consys.h`.

##### 5.41.2.4 coeff\_struct\* rowhdr\_struct\_tag::coeffs

Definition at line 141 of file `dy_consys.h`.

The documentation for this struct was generated from the following file:

- [/home/ted/COIN/trunk/DyLP/src/DyLP/dy\\_consys.h](#)

## 6 File Documentation

### 6.1 /home/ted/COIN/trunk/DyLP/src/DyLP/dy\_cmdint.h File Reference

```
#include "dylib_std.h"
```

```
#include "dylib_io.h"
#include "dylib_errs.h"
#include "dylp.h"
```

## Macros

- `#define DYLP_INTERNAL`

## Enumerations

- enum `cmd_retval` { `cmdOK`, `cmdHALTERROR`, `cmdHALTNOERROR` }

## Functions

- `cmd_retval dy_processcmds` ( `ioid cmdchn`, `bool silent`, `lpopts_struct *lpopts`, `lptols_struct *lptols` )

### 6.1.1 Macro Definition Documentation

#### 6.1.1.1 `#define DYLP_INTERNAL`

Definition at line 32 of file `dy_cmdint.h`.

### 6.1.2 Enumeration Type Documentation

#### 6.1.2.1 enum `cmd_retval`

## Enumerator

**`cmdOK`**  
**`cmdHALTERROR`**  
**`cmdHALTNOERROR`**

Definition at line 59 of file `dy_cmdint.h`.

### 6.1.3 Function Documentation

#### 6.1.3.1 `cmd_retval dy_processcmds` ( `ioid cmdchn`, `bool silent`, `lpopts_struct *lpopts`, `lptols_struct *lptols` )

## 6.2 /home/ted/COIN/trunk/DyLP/src/DyLP/dy\_consys.h File Reference

```
#include "dy_vector.h"
#include "dylib_io.h"
#include "dylib_std.h"
```

## Classes

- struct [coeff\\_struct\\_tag](#)
- struct [colhdr\\_struct\\_tag](#)
- struct [rowhdr\\_struct\\_tag](#)
- struct [conmtx\\_struct](#)
- struct [attvhdr\\_struct\\_tag](#)
- struct [conbnd\\_struct](#)
- struct [consys\\_struct](#)

## Macros

- #define [CONSYS\\_MTX](#) ((flags) 1<<0)
- #define [CONSYS\\_ROW](#) ((flags) 1<<1)
- #define [CONSYS\\_COL](#) ((flags) 1<<2)
- #define [CONSYS\\_OBJ](#) ((flags) 1<<3)
- #define [CONSYS\\_VUB](#) ((flags) 1<<4)
- #define [CONSYS\\_VLB](#) ((flags) 1<<5)
- #define [CONSYS\\_RHS](#) ((flags) 1<<6)
- #define [CONSYS\\_CUB](#) ((flags) 1<<7)
- #define [CONSYS\\_CLB](#) ((flags) 1<<8)
- #define [CONSYS\\_RHSLOW](#) ((flags) 1<<9)
- #define [CONSYS\\_VTYP](#) ((flags) 1<<10)
- #define [CONSYS\\_CTYP](#) ((flags) 1<<11)
- #define [CONSYS\\_COLHDR](#) ((flags) 1<<12)
- #define [CONSYS\\_ROWHDR](#) ((flags) 1<<13)
- #define [CONSYS\\_RSCALE](#) ((flags) 1<<14)
- #define [CONSYS\\_CSCALE](#) ((flags) 1<<15)
- #define [CONSYS\\_ROWVEC](#)
- #define [CONSYS\\_COLVEC](#)
- #define [VALID\\_ATTVTYP](#)(zz\_vectype\_zz)
- #define [VALID\\_CONTYPE](#)(zz\_ctyp\_zz)
- #define [VALID\\_VARTYPE](#)(zz\_vtyp\_zz)
- #define [INT\\_VARTYPE](#)(zz\_vtyp\_zz)
- #define [CONSYS\\_LVARS](#) ((flags) 1<<0)
- #define [CONSYS\\_WRNZERO](#) ((flags) 1<<1)
- #define [CONSYS\\_WRNATT](#) ((flags) 1<<2)
- #define [CONSYS\\_FININF](#) ((flags) 1<<3)
- #define [CONSYS\\_CORRUPT](#) ((flags) 1<<4)
- #define [CONSYS\\_MAXBUFLN](#) 32

## Typedefs

- typedef struct [coeff\\_struct\\_tag](#) [coeff\\_struct](#)
- typedef struct [colhdr\\_struct\\_tag](#) [colhdr\\_struct](#)
- typedef struct [rowhdr\\_struct\\_tag](#) [rowhdr\\_struct](#)
- typedef struct [attvhdr\\_struct\\_tag](#) [attvhdr\\_struct](#)

## Enumerations

- enum `contyp_enum` {  
`contypINV` = 0, `contypNB`, `contypGE`, `contypEQ`,  
`contypLE`, `contypRNG` }
- enum `vartyp_enum` { `vartypINV` = 0, `vartypCON`, `vartypINT`, `vartypBIN` }

## Functions

- `consys_struct * consys_create` (const char \*nme, flags parts, flags opts, int concnt, int varcnt, double infinity)
- `bool consys_dupsys` (consys\_struct \*src, consys\_struct \*\*dst, flags dstvecs)
- `void consys_free` (consys\_struct \*consys)
- `bool consys_realloc` (consys\_struct \*consys, char rowcol, int incr)
- `bool consys_attach` (consys\_struct \*consys, flags what, int elsze, void \*\*pvec)
- `bool consys_update` (consys\_struct \*consys, void \*oldvec, void \*newvec)
- `bool consys_detach` (consys\_struct \*consys, void \*\*pvec, bool all)
- `bool consys_addcol_pk` (consys\_struct \*consys, vartyp\_enum vartyp, pkvec\_struct \*pkcol, double obj, double vlb, double vub)
- `bool consys_addcol_ex` (consys\_struct \*consys, vartyp\_enum vartyp, const char \*\*nme, double \*excol, double obj, double vlb, double vub)
- `bool consys_addrow_pk` (consys\_struct \*consys, char rowclass, contyp\_enum contyp, pkvec\_struct \*pkrow, double rhs, double rhslow, conbnd\_struct \*cub, conbnd\_struct \*clb)
- `bool consys_getcol_pk` (consys\_struct \*consys, int colndx, pkvec\_struct \*\*pkvec)
- `bool consys_getcol_ex` (consys\_struct \*consys, int colndx, double \*\*vec)
- `bool consys_getrow_pk` (consys\_struct \*consys, int rowndx, pkvec\_struct \*\*pkvec)
- `bool consys_getrow_ex` (consys\_struct \*consys, int rowndx, double \*\*vec)
- `bool consys_delcol` (consys\_struct \*consys, int colndx)
- `bool consys_delrow` (consys\_struct \*consys, int rowndx)
- `bool consys_delrow_stable` (consys\_struct \*consys, int rowndx)
- `bool consys_setcoeff` (consys\_struct \*consys, int rowndx, int colndx, double val)
- `double consys_getcoeff` (consys\_struct \*consys, int rowndx, int colndx)
- `bool consys_logicals` (consys\_struct \*consys)
- `int consys_gcdrow` (consys\_struct \*consys, int rowndx)
- `double consys_dotcol` (consys\_struct \*consys, int colndx, double \*vec)
- `double consys_dotrow` (consys\_struct \*consys, int rowndx, double \*vec)
- `double consys_1normrow` (consys\_struct \*consys, int rowndx)
- `double consys_ssqrow` (consys\_struct \*consys, int rowndx)
- `double consys_2normrow` (consys\_struct \*consys, int rowndx)
- `double consys_infnormrow` (consys\_struct \*consys, int rowndx)
- `double consys_1normcol` (consys\_struct \*consys, int rowndx)
- `double consys_ssqcol` (consys\_struct \*consys, int rowndx)
- `double consys_2normcol` (consys\_struct \*consys, int rowndx)
- `double consys_infnormcol` (consys\_struct \*consys, int rowndx)
- `bool consys_mulrow` (consys\_struct \*consys, int rowndx, double scalar)
- `bool consys_divrow` (consys\_struct \*consys, int rowndx, double scalar)
- `bool consys_accumcol` (consys\_struct \*consys, int colndx, double \*vec)
- `bool consys_mulaccumcol` (consys\_struct \*consys, int colndx, double scalar, double \*vec)
- `bool consys_evalsys` (consys\_struct \*consys, double \*scm, int \*gecnt)
- `bool consys_geomscale` (consys\_struct \*consys, double \*\*rowscale, double \*\*colscale)
- `bool consys_equiscale` (consys\_struct \*consys, double \*\*rowscale, double \*\*colscale)
- `bool consys_appliescale` (consys\_struct \*consys, bool convctyp, double \*rowscale, double \*colscale)

- const char \* [consys\\_prtvartyp](#) ([vartyp\\_enum](#) vartyp)
- const char \* [consys\\_prtcontyp](#) ([contyp\\_enum](#) contyp)
- char \* [consys\\_assocnme](#) ([consys\\_struct](#) \*consys, [flags](#) which)
- char \* [consys\\_conbndnme](#) (char bndlett, int cndx, [conbnd\\_struct](#) \*bnd)
- char \* [consys\\_conbndval](#) ([conbnd\\_struct](#) \*bnd)
- void [consys\\_prtcon](#) (ioid chn, bool echo, [consys\\_struct](#) \*consys, int i, const char \*pfx)
- void [consys\\_chgnme](#) ([consys\\_struct](#) \*consys, char cv, int ndx, const char \*newnme)
- const char \* [consys\\_nme](#) ([consys\\_struct](#) \*consys, char cv, int ndx, bool pfx, char \*clientbuf)

## 6.2.1 Macro Definition Documentation

### 6.2.1.1 #define CONSYS\_MTX ((flags) 1<<0)

Definition at line 208 of file dy\_consys.h.

### 6.2.1.2 #define CONSYS\_ROW ((flags) 1<<1)

Definition at line 209 of file dy\_consys.h.

### 6.2.1.3 #define CONSYS\_COL ((flags) 1<<2)

Definition at line 210 of file dy\_consys.h.

### 6.2.1.4 #define CONSYS\_OBJ ((flags) 1<<3)

Definition at line 211 of file dy\_consys.h.

### 6.2.1.5 #define CONSYS\_VUB ((flags) 1<<4)

Definition at line 212 of file dy\_consys.h.

### 6.2.1.6 #define CONSYS\_VLB ((flags) 1<<5)

Definition at line 213 of file dy\_consys.h.

### 6.2.1.7 #define CONSYS\_RHS ((flags) 1<<6)

Definition at line 214 of file dy\_consys.h.

### 6.2.1.8 #define CONSYS\_CUB ((flags) 1<<7)

Definition at line 215 of file dy\_consys.h.

### 6.2.1.9 #define CONSYS\_CLB ((flags) 1<<8)

Definition at line 216 of file dy\_consys.h.

### 6.2.1.10 #define CONSYS\_RHSLOW ((flags) 1<<9)

Definition at line 217 of file dy\_consys.h.

### 6.2.1.11 #define CONSYS\_VTYP ((flags) 1<<10)

Definition at line 218 of file dy\_consys.h.

**6.2.1.12 #define CONSYS\_CTYP ((flags) 1<<11)**

Definition at line 219 of file dy\_consys.h.

**6.2.1.13 #define CONSYS\_COLHDR ((flags) 1<<12)**

Definition at line 220 of file dy\_consys.h.

**6.2.1.14 #define CONSYS\_ROWHDR ((flags) 1<<13)**

Definition at line 221 of file dy\_consys.h.

**6.2.1.15 #define CONSYS\_RSCALE ((flags) 1<<14)**

Definition at line 222 of file dy\_consys.h.

**6.2.1.16 #define CONSYS\_CSCALE ((flags) 1<<15)**

Definition at line 223 of file dy\_consys.h.

**6.2.1.17 #define CONSYS\_ROWVEC**

**Value:**

```
(CONSYS_OBJ|CONSYS_VUB|CONSYS_VLB|CONSYS_VTYP|
 CONSYS_CSCALE| \
 CONSYS_COLHDR|CONSYS_ROW)
```

Definition at line 228 of file dy\_consys.h.

**6.2.1.18 #define CONSYS\_COLVEC**

**Value:**

```
(CONSYS_RHS|CONSYS_RHSLOW|CONSYS_CUB|CONSYS_CLB|
 CONSYS_CTYP|CONSYS_RSCALE| \
 CONSYS_ROWHDR|CONSYS_COL)
```

Definition at line 232 of file dy\_consys.h.

**6.2.1.19 #define VALID\_ATTVTTYPE( zz\_vectype\_zz )**

**Value:**

```
(zz_vectype_zz == CONSYS_OBJ || \
 zz_vectype_zz == CONSYS_VUB || zz_vectype_zz == CONSYS_VLB || \
 zz_vectype_zz == CONSYS_RHS || zz_vectype_zz == CONSYS_RHSLOW || \
 zz_vectype_zz == CONSYS_CUB || zz_vectype_zz == CONSYS_CUB || \
 zz_vectype_zz == CONSYS_VTYP || zz_vectype_zz == CONSYS_CTYP || \
 zz_vectype_zz == CONSYS_RSCALE || zz_vectype_zz == CONSYS_CSCALE || \
 zz_vectype_zz == CONSYS_ROW || zz_vectype_zz == CONSYS_COL)
```

Definition at line 240 of file dy\_consys.h.

**6.2.1.20 #define VALID\_CONTYPE( zz\_ctyp\_zz )**

**Value:**

```
(zz_ctyp_zz == contypGE || zz_ctyp_zz == contypEQ || \
 zz_ctyp_zz == contypLE || zz_ctyp_zz == contypRNG)
```

Definition at line 341 of file dy\_consys.h.

**6.2.1.21 #define VALID\_VARTYPE( zz\_vtyp\_zz )****Value:**

```
(zz_vtyp_zz == vartypCON || \
 zz_vtyp_zz == vartypINT || \
 zz_vtyp_zz == vartypBIN)
```

Definition at line 357 of file dy\_consys.h.

**6.2.1.22 #define INT\_VARTYPE( zz\_vtyp\_zz )****Value:**

```
(zz_vtyp_zz == vartypINT || \
 zz_vtyp_zz == vartypBIN)
```

Definition at line 362 of file dy\_consys.h.

**6.2.1.23 #define CONSYS\_LVARS ((flags) 1<<0)**

Definition at line 394 of file dy\_consys.h.

**6.2.1.24 #define CONSYS\_WRNZERO ((flags) 1<<1)**

Definition at line 395 of file dy\_consys.h.

**6.2.1.25 #define CONSYS\_WRNATT ((flags) 1<<2)**

Definition at line 396 of file dy\_consys.h.

**6.2.1.26 #define CONSYS\_FININF ((flags) 1<<3)**

Definition at line 397 of file dy\_consys.h.

**6.2.1.27 #define CONSYS\_CORRUPT ((flags) 1<<4)**

Definition at line 398 of file dy\_consys.h.

**6.2.1.28 #define CONSYS\_MAXBUFLN 32**

Definition at line 625 of file dy\_consys.h.

**6.2.2 Typedef Documentation****6.2.2.1 typedef struct coeff\_struct\_tag coeff\_struct****6.2.2.2 typedef struct colhdr\_struct\_tag colhdr\_struct****6.2.2.3 typedef struct rowhdr\_struct\_tag rowhdr\_struct****6.2.2.4 typedef struct attvhdr\_struct\_tag attvhdr\_struct****6.2.3 Enumeration Type Documentation**



## 6.2.3.1 enum contyp\_enum

Enumerator

***contypINV***  
***contypNB***  
***contypGE***  
***contypEQ***  
***contypLE***  
***contypRNG***

Definition at line 338 of file dy\_consys.h.

## 6.2.3.2 enum vartyp\_enum

Enumerator

***vartypINV***  
***vartypCON***  
***vartypINT***  
***vartypBIN***

Definition at line 354 of file dy\_consys.h.

## 6.2.4 Function Documentation

6.2.4.1 **consys\_struct\*** consys\_create ( const char \* *nme*, flags *parts*, flags *opts*, int *concnt*, int *varcnt*, double *infinity* )

6.2.4.2 **bool** consys\_dupsys ( consys\_struct \* *src*, consys\_struct \*\* *dst*, flags *dstvecs* )

6.2.4.3 **void** consys\_free ( consys\_struct \* *consys* )

6.2.4.4 **bool** consys\_realloc ( consys\_struct \* *consys*, char *rowcol*, int *incr* )

6.2.4.5 **bool** consys\_attach ( consys\_struct \* *consys*, flags *what*, int *elsze*, void \*\* *pvec* )

6.2.4.6 **bool** consys\_update ( consys\_struct \* *consys*, void \* *oldvec*, void \* *newvec* )

6.2.4.7 **bool** consys\_detach ( consys\_struct \* *consys*, void \*\* *pvec*, bool *all* )

6.2.4.8 **bool** consys\_addcol\_pk ( consys\_struct \* *consys*, vartyp\_enum *vartyp*, pkvec\_struct \* *pkcol*, double *obj*, double *vlb*, double *vub* )

6.2.4.9 **bool** consys\_addcol\_ex ( consys\_struct \* *consys*, vartyp\_enum *vartyp*, const char \*\* *nme*, double \* *excol*, double *obj*, double *vlb*, double *vub* )

6.2.4.10 **bool** consys\_addrow\_pk ( consys\_struct \* *consys*, char *rowclass*, contyp\_enum *contyp*, pkvec\_struct \* *pkrow*, double *rhs*, double *rhslow*, conbnd\_struct \* *cub*, conbnd\_struct \* *clb* )

6.2.4.11 **bool** consys\_getcol\_pk ( consys\_struct \* *consys*, int *colndx*, pkvec\_struct \*\* *pkvec* )

6.2.4.12 **bool** consys\_getcol\_ex ( consys\_struct \* *consys*, int *colndx*, double \*\* *vec* )

6.2.4.13 **bool** consys\_getrow\_pk ( consys\_struct \* *consys*, int *rowndx*, pkvec\_struct \*\* *pkvec* )

- 6.2.4.14 `bool consys_getrow_ex ( consys_struct * consys, int rowndx, double ** vec )`
- 6.2.4.15 `bool consys_delcol ( consys_struct * consys, int colndx )`
- 6.2.4.16 `bool consys_delrow ( consys_struct * consys, int rowndx )`
- 6.2.4.17 `bool consys_delrow_stable ( consys_struct * consys, int rowndx )`
- 6.2.4.18 `bool consys_setcoeff ( consys_struct * consys, int rowndx, int colndx, double val )`
- 6.2.4.19 `double consys_getcoeff ( consys_struct * consys, int rowndx, int colndx )`
- 6.2.4.20 `bool consys_logicals ( consys_struct * consys )`
- 6.2.4.21 `int consys_gcdrow ( consys_struct * consys, int rowndx )`
- 6.2.4.22 `double consys_dotcol ( consys_struct * consys, int colndx, double * vec )`
- 6.2.4.23 `double consys_dotrow ( consys_struct * consys, int rowndx, double * vec )`
- 6.2.4.24 `double consys_1normrow ( consys_struct * consys, int rowndx )`
- 6.2.4.25 `double consys_ssqrrow ( consys_struct * consys, int rowndx )`
- 6.2.4.26 `double consys_2normrow ( consys_struct * consys, int rowndx )`
- 6.2.4.27 `double consys_infnormrow ( consys_struct * consys, int rowndx )`
- 6.2.4.28 `double consys_1normcol ( consys_struct * consys, int rowndx )`
- 6.2.4.29 `double consys_ssqrcol ( consys_struct * consys, int rowndx )`
- 6.2.4.30 `double consys_2normcol ( consys_struct * consys, int rowndx )`
- 6.2.4.31 `double consys_infnormcol ( consys_struct * consys, int rowndx )`
- 6.2.4.32 `bool consys_mulrow ( consys_struct * consys, int rowndx, double scalar )`
- 6.2.4.33 `bool consys_divrow ( consys_struct * consys, int rowndx, double scalar )`
- 6.2.4.34 `bool consys_accumcol ( consys_struct * consys, int colndx, double * vec )`
- 6.2.4.35 `bool consys_mulaccumcol ( consys_struct * consys, int colndx, double scalar, double * vec )`
- 6.2.4.36 `bool consys_evalsys ( consys_struct * consys, double * scm, int * gecnt )`
- 6.2.4.37 `bool consys_geomscale ( consys_struct * consys, double ** rowscale, double ** colscale )`
- 6.2.4.38 `bool consys_equiscale ( consys_struct * consys, double ** rowscale, double ** colscale )`
- 6.2.4.39 `bool consys_applyscale ( consys_struct * consys, bool convctyp, double * rowscale, double * colscale )`
- 6.2.4.40 `const char* consys_prtvartyp ( vartyp_enum vartyp )`
- 6.2.4.41 `const char * consys_prtcontyp ( contyp_enum contyp )`
- 6.2.4.42 `char* consys_assocnme ( consys_struct * consys, flags which )`

6.2.4.43 `char * consys_conbndnme ( char bndlett, int cndx, conbnd_struct * bnd )`

6.2.4.44 `char * consys_conbndval ( conbnd_struct * bnd )`

6.2.4.45 `void consys_prtcon ( ioid chn, bool echo, consys_struct * consys, int i, const char * pfx )`

6.2.4.46 `void consys_chgnme ( consys_struct * consys, char cv, int ndx, const char * newnme )`

6.2.4.47 `const char* consys_nme ( consys_struct * consys, char cv, int ndx, bool pfx, char * clientbuf )`

## 6.3 /home/ted/COIN/trunk/DyLP/src/DyLP/dy\_vector.h File Reference

```
#include <DyLPConfig.h>
#include <ctype.h>
#include "dylib_std.h"
#include <math.h>
```

### Classes

- struct [pkcoeff\\_struct](#)
- struct [pkvec\\_struct](#)

### Functions

- [pkvec\\_struct \\* pkvec\\_new](#) (int *size*)
- [bool pkvec\\_resize](#) ([pkvec\\_struct](#) \**pkvec*, int *size*)
- [void pkvec\\_free](#) ([pkvec\\_struct](#) \**pkvec*)
- [bool pkvec\\_check](#) ([pkvec\\_struct](#) \**pkvec*, const char \**caller*)
- [double pkvec\\_2norm](#) ([pkvec\\_struct](#) \**vec*)
- [double exvec\\_1norm](#) (double \**vec*, int *len*)
- [double exvec\\_ssqr](#) (double \**vec*, int *len*)
- [double exvec\\_2norm](#) (double \**vec*, int *len*)
- [double exvec\\_infnorm](#) (double \**vec*, int *len*, int \**p\_jmax*)
- [double pkvec\\_dotexvec](#) ([pkvec\\_struct](#) \**pkvec*, double \**exvec*)

### 6.3.1 Function Documentation

6.3.1.1 `pkvec\_struct\* pkvec\_new ( int size )`

6.3.1.2 `bool pkvec\_resize ( pkvec\_struct * pkvec, int size )`

6.3.1.3 `void pkvec\_free ( pkvec\_struct * pkvec )`

6.3.1.4 `bool pkvec\_check ( pkvec\_struct * pkvec, const char * caller )`

6.3.1.5 `double pkvec\_2norm ( pkvec\_struct * vec )`

6.3.1.6 `double exvec\_1norm ( double * vec, int len )`

6.3.1.7 `double exvec\_ssqr ( double * vec, int len )`

6.3.1.8 double exvec\_2norm ( double \* *vec*, int *len* )

6.3.1.9 double exvec\_infnorm ( double \* *vec*, int *len*, int \* *p\_jmax* )

6.3.1.10 double pkvec\_dotexvec ( pkvec\_struct \* *pkvec*, double \* *exvec* )

## 6.4 /home/ted/COIN/trunk/DyLP/src/DyLP/dyLP.h File Reference

```
#include "dylib_errs.h"
#include "dylib_io.h"
#include "dy_consys.h"
```

### Classes

- struct [basisel\\_struct](#)
- struct [basis\\_struct](#)
- struct [lpprob\\_struct](#)
- struct [lptols\\_struct](#)
- struct [lpopts\\_struct](#)
- struct [lpstats\\_struct](#)

### Macros

- #define [ladPRIMFEAS](#) 1<<0
- #define [ladPRIMALCHK](#) 1<<1
- #define [ladPFQUIET](#) 1<<2
- #define [ladDUALFEAS](#) 1<<3
- #define [ladDUALCHK](#) 1<<4
- #define [ladDFQUIET](#) 1<<5
- #define [ladDUALS](#) 1<<6
- #define [ladPRIMALS](#) 1<<7
- #define [ladFACTOR](#) 1<<8
- #define [ladEXPAND](#) 1<<9
- #define [vstatINV](#) 0
- #define [vstatBFX](#) 1<<0
- #define [vstatBUB](#) 1<<1
- #define [vstatB](#) 1<<2
- #define [vstatBLB](#) 1<<3
- #define [vstatBFR](#) 1<<4
- #define [vstatNBFX](#) 1<<5
- #define [vstatNBUB](#) 1<<6
- #define [vstatNBLB](#) 1<<7
- #define [vstatNBFR](#) 1<<8
- #define [vstatSB](#) 1<<9
- #define [vstatBUUB](#) 1<<10
- #define [vstatBLLB](#) 1<<11
- #define [vstatNOPIVOT](#) (([flags](#)) 1<<(sizeof([flags](#))\*8-2))
- #define [vstatNOPER](#) (([flags](#)) 1<<(sizeof([flags](#))\*8-3))
- #define [vstatNOLOAD](#) (([flags](#)) 1<<(sizeof([flags](#))\*8-4))
- #define [vstatBASIC](#) ([vstatBFX](#)|[vstatBUUB](#)|[vstatBUB](#)|[vstatB](#)|[vstatBLB](#)|[vstatBLLB](#)|[vstatBFR](#))

- #define `vstatNONBASIC` (`vstatNBFX|vstatNBUB|vstatNBLB`)
- #define `vstatEXOTIC` (`vstatSB|vstatNBFR`)
- #define `vstatSTATUS` (`vstatBASIC|vstatNONBASIC|vstatEXOTIC`)
- #define `vstatQUALS` (`vstatNOPIVOT|vstatNOPER|vstatNOLOAD`)
- #define `VALID_STATUS`(`zz_status_zz`)
- #define `lpctlNOFREE` 1<<0
- #define `lpctlONLYFREE` 1<<1
- #define `lpctlUBNDCHG` 1<<2
- #define `lpctlLBNDCHG` 1<<3
- #define `lpctlRHSCHG` 1<<4
- #define `lpctlOBJCHG` 1<<5
- #define `lpctlACTVARSIN` 1<<6
- #define `lpctlINITACTVAR` 1<<7
- #define `lpctlINITACTCON` 1<<8
- #define `lpctlACTVARSOOT` 1<<10
- #define `lpctlDYVALID` 1<<11
- #define `DYSTATS_MAXDEGEN` 25
- #define `DYSTATS_HISTBINS` 37

#### Enumerations

- enum `lpnet_enum` {  
`lpFATAL` = -1, `lpINV` = 0, `lpOPTIMAL`, `lpUNBOUNDED`,  
`lpSWING`, `lpINFEAS`, `lpACCCHK`, `lpSTALLED`,  
`lpITERLIM`, `lpNOSPACE`, `lpLOSTFEAS`, `lpPUNT`,  
`lpFORCEDUAL`, `lpFORCEPRIMAL`, `lpFORCEFULL` }
- enum `dyphase_enum` {  
`dyINV` = 0, `dyINIT`, `dyPRIMAL1`, `dyPRIMAL2`,  
`dyDUAL`, `dyPURGEVAR`, `dyGENVAR`, `dyADDVAR`,  
`dyPURGECON`, `dyGENCON`, `dyADDCON`, `dyFORCEDUAL`,  
`dyFORCEPRIMAL`, `dyFORCEFULL`, `dyDONE` }
- enum `dyret_enum` {  
`dyrFATAL` = -10, `dyrITERLIM`, `dyrSTALLED`, `dyrBSPACE` = -7,  
`dyrSINGULAR` = -6, `dyrNUMERIC` = -5, `dyrLOSTPFEAS`, `dyrLOSTDFEAS`,  
`dyrDEGEN`, `dyrMADPIV`, `dyrINV` = 0, `dyrOK` = 1,  
`dyrPATCHED` = 2, `dyrRESELECT`, `dyrREQCHK`, `dyrACCCHK`,  
`dyrPUNT`, `dyrOPTIMAL`, `dyrUNBOUND`, `dyrSWING`,  
`dyrINFEAS` }
- enum `ibtype_enum` { `ibINV` = 0, `ibLOGICAL`, `ibSLACK`, `ibARCH` }
- enum `cxtype_enum` {  
`cxINV` = 0, `cxLOAD`, `cxUNLOAD`, `cxSINGLELP`,  
`cxINITIALLP`, `cxBANDC`, `cxUSERPIV` }

#### Functions

- void `dy_defaults` (`lpopts_struct **opts`, `lptols_struct **tols`)
- void `dy_checkdefaults` (`consys_struct *sys`, `lpopts_struct *opts`, `lptols_struct *tols`)
- void `dy_setprintopts` (int `lvl`, `lpopts_struct *opts`)
- `lpnet_enum` `dyLP` (`lpprob_struct *orig_lp`, `lpopts_struct *orig_opts`, `lptols_struct *orig_tols`, `lpstats_struct *orig_stats`)
- void \* `dy_getOwner` ()

- [bool dy\\_dupbasis](#) (int dst\_basissize, [basis\\_struct](#) \*\*p\_dst\_basis, [basis\\_struct](#) \*src\_basis, int dst\_statussize, [flags](#) \*\*p\_dst\_status, int src\_statuslen, [flags](#) \*src\_status)
- void [dy\\_freesoln](#) ([lpprob\\_struct](#) \*lpprob)
- [bool dy\\_pricenbvars](#) ([lpprob\\_struct](#) \*orig\_lp, [flags](#) priceme, double \*\*p\_ocbar, int \*p\_nbcnt, int \*\*p\_nbvars)
- [bool dy\\_pricedualpiv](#) ([lpprob\\_struct](#) \*orig\_lp, int oxindx, double nubi, double xi, double nlbi, int nbcnt, int \*nbvars, double \*cbar, double \*p\_upeni, double \*p\_dpeni)
- [bool dy\\_abarj](#) ([lpprob\\_struct](#) \*orig\_lp, int tgt\_j, double \*\*p\_abarj)
- [bool dy\\_betaj](#) ([lpprob\\_struct](#) \*orig\_lp, int tgt\_j, double \*\*p\_betaj)
- [bool dy\\_betak](#) ([lpprob\\_struct](#) \*orig\_lp, int col\_k, double \*\*p\_betaj)
- [bool dy\\_betai](#) ([lpprob\\_struct](#) \*orig\_lp, int tgt\_i, double \*\*p\_betai)
- [bool dy\\_abari](#) ([lpprob\\_struct](#) \*orig\_lp, int tgt\_i, double \*\*p\_abari, double \*\*p\_betai)
- [bool dy\\_primalRays](#) ([lpprob\\_struct](#) \*orig\_lp, int \*p\_numRays, double \*\*\*p\_rays)
- [bool dy\\_dualRays](#) ([lpprob\\_struct](#) \*orig\_lp, [bool](#) fullRay, int \*p\_numRays, double \*\*\*p\_rays, [bool](#) trueDuals)
- void [dy\\_colDuals](#) ([lpprob\\_struct](#) \*orig\_lp, double \*\*p\_cbar, [bool](#) trueDuals)
- void [dy\\_rowDuals](#) ([lpprob\\_struct](#) \*orig\_lp, double \*\*p\_y, [bool](#) trueDuals)
- void [dy\\_rowDualsGivenC](#) ([lpprob\\_struct](#) \*orig\_lp, double \*\*p\_y, const double \*c, [bool](#) trueDuals)
- void [dy\\_colPrimals](#) ([lpprob\\_struct](#) \*orig\_lp, double \*\*p\_x)
- void [dy\\_rowPrimals](#) ([lpprob\\_struct](#) \*orig\_lp, double \*\*p\_xB, int \*\*p\_indB)
- void [dy\\_logPrimals](#) ([lpprob\\_struct](#) \*orig\_lp, double \*\*p\_logx)
- void [dy\\_colStatus](#) ([lpprob\\_struct](#) \*orig\_lp, [flags](#) \*\*p\_colstat)
- void [dy\\_logStatus](#) ([lpprob\\_struct](#) \*orig\_lp, [flags](#) \*\*p\_logstat)
- [bool dy\\_expandxopt](#) ([lpprob\\_struct](#) \*lp, double \*\*p\_xopt)
- const char \* [dy\\_prtlpret](#) ([lpret\\_enum](#) lpret)
- const char \* [dy\\_prtlpphase](#) ([dyphase\\_enum](#) phase, [bool](#) abbrev)
- char \* [dy\\_prtvstat](#) ([flags](#) status)
- [bool dy\\_dumpcompact](#) ([ioid](#) chn, [bool](#) echo, [lpprob\\_struct](#) \*soln, [bool](#) nbzeros)
- void [dy\\_setlogchn](#) ([ioid](#) chn)
- void [dy\\_setgtxecho](#) ([bool](#) echo)
- void [dy\\_initstats](#) ([lpstats\\_struct](#) \*\*p\_lpstats, [consys\\_struct](#) \*orig\_sys)
- void [dy\\_dumpstats](#) ([ioid](#) chn, [bool](#) echo, [lpstats\\_struct](#) \*lpstats, [consys\\_struct](#) \*orig\_sys)
- void [dy\\_freestats](#) ([lpstats\\_struct](#) \*\*p\_lpstats)

#### 6.4.1 Macro Definition Documentation

##### 6.4.1.1 #define ladPRIMFEAS 1<<0

Definition at line 317 of file dylp.h.

##### 6.4.1.2 #define ladPRIMALCHK 1<<1

Definition at line 318 of file dylp.h.

##### 6.4.1.3 #define ladPFQUIET 1<<2

Definition at line 319 of file dylp.h.

##### 6.4.1.4 #define ladDUALFEAS 1<<3

Definition at line 320 of file dylp.h.

##### 6.4.1.5 #define ladDUALCHK 1<<4

Definition at line 321 of file dylp.h.

#### 6.4.1.6 `#define ladDFQUIET 1<<5`

Definition at line 322 of file dyLP.h.

#### 6.4.1.7 `#define ladDUALS 1<<6`

Definition at line 323 of file dyLP.h.

#### 6.4.1.8 `#define ladPRIMALS 1<<7`

Definition at line 324 of file dyLP.h.

#### 6.4.1.9 `#define ladFACTOR 1<<8`

Definition at line 325 of file dyLP.h.

#### 6.4.1.10 `#define ladEXPAND 1<<9`

Definition at line 326 of file dyLP.h.

#### 6.4.1.11 `#define vstatINV 0`

Definition at line 377 of file dyLP.h.

#### 6.4.1.12 `#define vstatBFX 1<<0`

Definition at line 378 of file dyLP.h.

#### 6.4.1.13 `#define vstatBUB 1<<1`

Definition at line 379 of file dyLP.h.

#### 6.4.1.14 `#define vstatB 1<<2`

Definition at line 380 of file dyLP.h.

#### 6.4.1.15 `#define vstatBLB 1<<3`

Definition at line 381 of file dyLP.h.

#### 6.4.1.16 `#define vstatBFR 1<<4`

Definition at line 382 of file dyLP.h.

#### 6.4.1.17 `#define vstatNBFX 1<<5`

Definition at line 383 of file dyLP.h.

#### 6.4.1.18 `#define vstatNBUB 1<<6`

Definition at line 384 of file dyLP.h.

#### 6.4.1.19 `#define vstatNBLB 1<<7`

Definition at line 385 of file dyLP.h.

6.4.1.20 **#define vstatNBFR 1<<8**

Definition at line 386 of file dylp.h.

6.4.1.21 **#define vstatSB 1<<9**

Definition at line 387 of file dylp.h.

6.4.1.22 **#define vstatBUUB 1<<10**

Definition at line 388 of file dylp.h.

6.4.1.23 **#define vstatBLLB 1<<11**

Definition at line 389 of file dylp.h.

6.4.1.24 **#define vstatNOPIVOT ((flags) 1<<(sizeof(flags)\*8-2))**

Definition at line 396 of file dylp.h.

6.4.1.25 **#define vstatNOPER ((flags) 1<<(sizeof(flags)\*8-3))**

Definition at line 397 of file dylp.h.

6.4.1.26 **#define vstatNOLOAD ((flags) 1<<(sizeof(flags)\*8-4))**

Definition at line 398 of file dylp.h.

6.4.1.27 **#define vstatBASIC (vstatBFX|vstatBUUB|vstatBUB|vstatB|vstatBLB|vstatBLLB|vstatBFR)**

Definition at line 400 of file dylp.h.

6.4.1.28 **#define vstatNONBASIC (vstatNBFX|vstatNBUB|vstatNBLB)**

Definition at line 402 of file dylp.h.

6.4.1.29 **#define vstatEXOTIC (vstatSB|vstatNBFR)**

Definition at line 403 of file dylp.h.

6.4.1.30 **#define vstatSTATUS (vstatBASIC|vstatNONBASIC|vstatEXOTIC)**

Definition at line 405 of file dylp.h.

6.4.1.31 **#define vstatQUALS (vstatNOPIVOT|vstatNOPER|vstatNOLOAD)**

Definition at line 406 of file dylp.h.

6.4.1.32 **#define VALID\_STATUS( zz\_status\_zz )**

**Value:**

```
(zz_status_zz == vstatBFX || zz_status_zz == vstatBUB || \
 zz_status_zz == vstatB || zz_status_zz == vstatBLB || \
 zz_status_zz == vstatBFR || \
 zz_status_zz == vstatNBFX || zz_status_zz == vstatNBUB || \
 zz_status_zz == vstatNBLB || zz_status_zz == vstatNBFR || \
 zz_status_zz == vstatSB)
```

Definition at line 414 of file dylp.h.



6.4.1.33 `#define lpctlNOFREE 1<<0`

Definition at line 495 of file dyLP.h.

6.4.1.34 `#define lpctlONLYFREE 1<<1`

Definition at line 496 of file dyLP.h.

6.4.1.35 `#define lpctlUBNDCHG 1<<2`

Definition at line 497 of file dyLP.h.

6.4.1.36 `#define lpctlLBNDCHG 1<<3`

Definition at line 498 of file dyLP.h.

6.4.1.37 `#define lpctlRHSCHG 1<<4`

Definition at line 499 of file dyLP.h.

6.4.1.38 `#define lpctlOBJCHG 1<<5`

Definition at line 500 of file dyLP.h.

6.4.1.39 `#define lpctlACTVARSIN 1<<6`

Definition at line 501 of file dyLP.h.

6.4.1.40 `#define lpctlINITACTVAR 1<<7`

Definition at line 502 of file dyLP.h.

6.4.1.41 `#define lpctlINITACTCON 1<<8`

Definition at line 503 of file dyLP.h.

6.4.1.42 `#define lpctlACTVARSOOUT 1<<10`

Definition at line 505 of file dyLP.h.

6.4.1.43 `#define lpctlDYVALID 1<<11`

Definition at line 507 of file dyLP.h.

6.4.1.44 `#define DYSTATS_MAXDEGEN 25`

Definition at line 1300 of file dyLP.h.

6.4.1.45 `#define DYSTATS_HISTBINS 37`

Definition at line 1301 of file dyLP.h.

## 6.4.2 Enumeration Type Documentation

#### 6.4.2.1 enum lpret\_enum

Enumerator

***lpFATAL***  
***lpINV***  
***lpOPTIMAL***  
***lpUNBOUNDED***  
***lpSWING***  
***lpINFEAS***  
***lpACCCHK***  
***lpSTALLED***  
***lpITERLIM***  
***lpNOSPACE***  
***lpLOSTFEAS***  
***lpPUNT***  
***lpFORCEDUAL***  
***lpFORCEPRIMAL***  
***lpFORCEFULL***

Definition at line 170 of file dylp.h.

#### 6.4.2.2 enum dyphase\_enum

Enumerator

***dyINV***  
***dyINIT***  
***dyPRIMAL1***  
***dyPRIMAL2***  
***dyDUAL***  
***dyPURGEVAR***  
***dyGENVAR***  
***dyADDVAR***  
***dyPURGECON***  
***dyGENCON***  
***dyADDCON***  
***dyFORCEDUAL***  
***dyFORCEPRIMAL***  
***dyFORCEFULL***  
***dyDONE***

Definition at line 212 of file dylp.h.

## 6.4.2.3 enum dyret\_enum

Enumerator

***dyrFATAL***  
***dyrITERLIM***  
***dyrSTALLED***  
***dyrBSPACE***  
***dyrSINGULAR***  
***dyrNUMERIC***  
***dyrLOSTPFEAS***  
***dyrLOSTDFEAS***  
***dyrDEGEN***  
***dyrMADPIV***  
***dyrINV***  
***dyrOK***  
***dyrPATCHED***  
***dyrRESELECT***  
***dyrREQCHK***  
***dyrACCCHK***  
***dyrPUNT***  
***dyrOPTIMAL***  
***dyrUNBOUND***  
***dyrSWING***  
***dyrINFEAS***

Definition at line 274 of file dyLP.h.

## 6.4.2.4 enum ibtype\_enum

Enumerator

***ibINV***  
***ibLOGICAL***  
***ibSLACK***  
***ibARCH***

Definition at line 766 of file dyLP.h.

## 6.4.2.5 enum cxtype\_enum

Enumerator

***cxINV***  
***cxLOAD***  
***cxUNLOAD***  
***cxSINGLELP***  
***cxINITIALLP***  
***cxBANDC***  
***cxUSERPIV***

Definition at line 788 of file dyLP.h.

### 6.4.3 Function Documentation

- 6.4.3.1 void dy\_defaults ( lpopts\_struct \*\* *opts*, lptols\_struct \*\* *tols* )
- 6.4.3.2 void dy\_checkdefaults ( consys\_struct \* *sys*, lpopts\_struct \* *opts*, lptols\_struct \* *tols* )
- 6.4.3.3 void dy\_setprintopts ( int *lvl*, lpopts\_struct \* *opts* )
- 6.4.3.4 lpret\_enum dylp ( lpprob\_struct \* *orig\_lp*, lpopts\_struct \* *orig\_opts*, lptols\_struct \* *orig\_tols*, lpstats\_struct \* *orig\_stats* )
- 6.4.3.5 void\* dy\_getOwner ( )
- 6.4.3.6 bool dy\_dupbasis ( int *dst\_basissize*, basis\_struct \*\* *p\_dst\_basis*, basis\_struct \* *src\_basis*, int *dst\_statussize*, flags \*\* *p\_dst\_status*, int *src\_statuslen*, flags \* *src\_status* )
- 6.4.3.7 void dy\_freesoln ( lpprob\_struct \* *lpprob* )
- 6.4.3.8 bool dy\_pricenbvars ( lpprob\_struct \* *orig\_lp*, flags *priceme*, double \*\* *p\_ocbar*, int \* *p\_nbcnt*, int \*\* *p\_nbvars* )
- 6.4.3.9 bool dy\_pricedualpiv ( lpprob\_struct \* *orig\_lp*, int *oxindx*, double *nubi*, double *xi*, double *nlbi*, int *nbcnt*, int \* *nbvars*, double \* *cbar*, double \* *p\_upeni*, double \* *p\_dpeni* )
- 6.4.3.10 bool dy\_abarj ( lpprob\_struct \* *orig\_lp*, int *tgt\_j*, double \*\* *p\_abarj* )
- 6.4.3.11 bool dy\_betaj ( lpprob\_struct \* *orig\_lp*, int *tgt\_j*, double \*\* *p\_betaj* )
- 6.4.3.12 bool dy\_betak ( lpprob\_struct \* *orig\_lp*, int *col\_k*, double \*\* *p\_betaj* )
- 6.4.3.13 bool dy\_betai ( lpprob\_struct \* *orig\_lp*, int *tgt\_i*, double \*\* *p\_betai* )
- 6.4.3.14 bool dy\_abari ( lpprob\_struct \* *orig\_lp*, int *tgt\_i*, double \*\* *p\_abari*, double \*\* *p\_betai* )
- 6.4.3.15 bool dy\_primalRays ( lpprob\_struct \* *orig\_lp*, int \* *p\_numRays*, double \*\*\* *p\_rays* )
- 6.4.3.16 bool dy\_dualRays ( lpprob\_struct \* *orig\_lp*, bool *fullRay*, int \* *p\_numRays*, double \*\*\* *p\_rays*, bool *trueDuals* )
- 6.4.3.17 void dy\_colDuals ( lpprob\_struct \* *orig\_lp*, double \*\* *p\_cbar*, bool *trueDuals* )
- 6.4.3.18 void dy\_rowDuals ( lpprob\_struct \* *orig\_lp*, double \*\* *p\_y*, bool *trueDuals* )
- 6.4.3.19 void dy\_rowDualsGivenC ( lpprob\_struct \* *orig\_lp*, double \*\* *p\_y*, const double \* *c*, bool *trueDuals* )
- 6.4.3.20 void dy\_colPrimals ( lpprob\_struct \* *orig\_lp*, double \*\* *p\_x* )
- 6.4.3.21 void dy\_rowPrimals ( lpprob\_struct \* *orig\_lp*, double \*\* *p\_xB*, int \*\* *p\_indB* )
- 6.4.3.22 void dy\_logPrimals ( lpprob\_struct \* *orig\_lp*, double \*\* *p\_logx* )
- 6.4.3.23 void dy\_colStatus ( lpprob\_struct \* *orig\_lp*, flags \*\* *p\_colstat* )
- 6.4.3.24 void dy\_logStatus ( lpprob\_struct \* *orig\_lp*, flags \*\* *p\_logstat* )
- 6.4.3.25 bool dy\_expandxopt ( lpprob\_struct \* *lp*, double \*\* *p\_xopt* )
- 6.4.3.26 const char\* dy\_prtlpref ( lpret\_enum *lpret* )

- 6.4.3.27 `const char * dy_prtlpphase ( dyphase_enum phase, bool abbrev )`
- 6.4.3.28 `char* dy_prtvstat ( flags status )`
- 6.4.3.29 `bool dy_dumpcompact ( ioid chn, bool echo, lpprob_struct * soln, bool nbzeros )`
- 6.4.3.30 `void dy_setlogchn ( ioid chn )`
- 6.4.3.31 `void dy_setgtxecho ( bool echo )`
- 6.4.3.32 `void dy_initstats ( lpstats_struct ** p_lpstats, consys_struct * orig_sys )`
- 6.4.3.33 `void dy_dumpstats ( ioid chn, bool echo, lpstats_struct * lpstats, consys_struct * orig_sys )`
- 6.4.3.34 `void dy_freestats ( lpstats_struct ** p_lpstats )`

## 6.5 /home/ted/COIN/trunk/DyLP/src/DyLP/inv.h File Reference

```
#include "glpluf.h"
```

### Classes

- struct [INV](#)

### Macros

- `#define inv\_create dy_glp_inv_create`
- `#define inv\_decomp dy_glp_inv_decomp`
- `#define inv\_h\_solve dy_glp_inv_h_solve`
- `#define inv\_ftran dy_glp_inv_ftran`
- `#define inv\_btran dy_glp_inv_btran`
- `#define inv\_update dy_glp_inv_update`
- `#define inv\_delete dy_glp_inv_delete`

### Typedefs

- `typedef struct INV INV`

### Functions

- `INV * inv\_create (int m, int max_upd)`
- `int inv\_decomp (INV *inv, void *info, int(*col)(void *info, int j, int rn[], double bj[]))`
- `void inv\_h\_solve (INV *inv, int tr, double x[])`
- `void inv\_ftran (INV *inv, double x[], int save)`
- `void inv\_btran (INV *inv, double x[])`
- `int inv\_update (INV *inv, int j)`
- `void inv\_delete (INV *inv)`

### 6.5.1 Macro Definition Documentation

#### 6.5.1.1 `#define inv_create dy_glp_inv_create`

Definition at line 25 of file `glpinv.h`.

#### 6.5.1.2 `#define inv_decomp dy_glp_inv_decomp`

Definition at line 26 of file `glpinv.h`.

#### 6.5.1.3 `#define inv_h_solve dy_glp_inv_h_solve`

Definition at line 27 of file `glpinv.h`.

#### 6.5.1.4 `#define inv_ftran dy_glp_inv_ftran`

Definition at line 28 of file `glpinv.h`.

#### 6.5.1.5 `#define inv_btran dy_glp_inv_btran`

Definition at line 29 of file `glpinv.h`.

#### 6.5.1.6 `#define inv_update dy_glp_inv_update`

Definition at line 30 of file `glpinv.h`.

#### 6.5.1.7 `#define inv_delete dy_glp_inv_delete`

Definition at line 31 of file `glpinv.h`.

### 6.5.2 Typedef Documentation

#### 6.5.2.1 `typedef struct INV INV`

Definition at line 76 of file `glpinv.h`.

### 6.5.3 Function Documentation

#### 6.5.3.1 `INV* inv_create ( int m, int max_upd )`

#### 6.5.3.2 `int inv_decomp ( INV * inv, void * info, int(*) (void *info, int j, int rn[], double bj[]) col )`

#### 6.5.3.3 `void inv_h_solve ( INV * inv, int tr, double x[] )`

#### 6.5.3.4 `void inv_ftran ( INV * inv, double x[], int save )`

#### 6.5.3.5 `void inv_btran ( INV * inv, double x[] )`

#### 6.5.3.6 `int inv_update ( INV * inv, int j )`

#### 6.5.3.7 `void inv_delete ( INV * inv )`

## 6.6 `/home/ted/COIN/trunk/DyLP/src/DyLP/glpplib.h` File Reference

## Classes

- struct [ENV](#)
- struct [MEM](#)
- struct [POOL](#)

## Macros

- #define [save\\_pointer](#) dy\_glp\_save\_pointer
- #define [read\\_pointer](#) dy\_glp\_read\_pointer
- #define [init\\_lib\\_env](#) dy\_glp\_init\_lib\_env
- #define [get\\_env\\_ptr](#) dy\_glp\_get\_env\_ptr
- #define [free\\_lib\\_env](#) dy\_glp\_free\_lib\_env
- #define [print](#) dy\_glp\_print
- #define [fault](#) dy\_glp\_fault
- #define [\\_insist](#) dy\_glp\_insist
- #define [watch](#) dy\_glp\_watch
- #define [umalloc](#) dy\_glp\_umalloc
- #define [ucalloc](#) dy\_glp\_ucalloc
- #define [ufree](#) dy\_glp\_ufree
- #define [create\\_pool](#) dy\_glp\_create\_pool
- #define [get\\_atom](#) dy\_glp\_get\_atom
- #define [free\\_atom](#) dy\_glp\_free\_atom
- #define [get\\_atomv](#) dy\_glp\_get\_atomv
- #define [clear\\_pool](#) dy\_glp\_clear\_pool
- #define [delete\\_pool](#) dy\_glp\_delete\_pool
- #define [insist](#)(expr) ((void)((expr) || (\_insist(#expr, \_\_FILE\_\_, \_\_LINE\_\_), 1)))
- #define [align\\_boundary](#) sizeof(double)
- #define [align\\_datasize](#)(size) (((size) + ([align\\_boundary](#) - 1)) / [align\\_boundary](#)) \* [align\\_boundary](#))

## Typedefs

- typedef struct [ENV](#) [ENV](#)
- typedef struct [MEM](#) [MEM](#)
- typedef struct [POOL](#) [POOL](#)

## Functions

- void [save\\_pointer](#) (void \*ptr)
- void \* [read\\_pointer](#) (void)
- int [init\\_lib\\_env](#) (void)
- [ENV](#) \* [get\\_env\\_ptr](#) (void)
- int [free\\_lib\\_env](#) (void)
- void [print](#) (const char \*fmt,...)
- void [fault](#) (const char \*fmt,...)
- void [\\_insist](#) (const char \*expr, const char \*file, int line)
- double [watch](#) (void)
- void \* [umalloc](#) (int size)
- void \* [ucalloc](#) (int nmemb, int size)
- void [ufree](#) (void \*ptr)

- `POOL * create_pool (int size)`
- `void * get_atom (POOL *pool)`
- `void free_atom (POOL *pool, void *ptr)`
- `void * get_atmv (POOL *pool, int size)`
- `void clear_pool (POOL *pool)`
- `void delete_pool (POOL *pool)`

## 6.6.1 Macro Definition Documentation

### 6.6.1.1 `#define save_pointer dy_glp_save_pointer`

Definition at line 21 of file `glplib.h`.

### 6.6.1.2 `#define read_pointer dy_glp_read_pointer`

Definition at line 22 of file `glplib.h`.

### 6.6.1.3 `#define init_lib_env dy_glp_init_lib_env`

Definition at line 24 of file `glplib.h`.

### 6.6.1.4 `#define get_env_ptr dy_glp_get_env_ptr`

Definition at line 25 of file `glplib.h`.

### 6.6.1.5 `#define free_lib_env dy_glp_free_lib_env`

Definition at line 26 of file `glplib.h`.

### 6.6.1.6 `#define print dy_glp_print`

Definition at line 28 of file `glplib.h`.

### 6.6.1.7 `#define fault dy_glp_fault`

Definition at line 29 of file `glplib.h`.

### 6.6.1.8 `#define _insist dy_glp_insist`

Definition at line 30 of file `glplib.h`.

### 6.6.1.9 `#define watch dy_glp_watch`

Definition at line 31 of file `glplib.h`.

### 6.6.1.10 `#define umalloc dy_glp_umalloc`

Definition at line 33 of file `glplib.h`.

### 6.6.1.11 `#define ucalloc dy_glp_ucalloc`

Definition at line 34 of file `glplib.h`.

### 6.6.1.12 `#define ufree dy_glp_ufree`

Definition at line 35 of file `glplib.h`.



#### 6.6.1.13 `#define create_pool dy_glp_create_pool`

Definition at line 37 of file `gplib.h`.

#### 6.6.1.14 `#define get_atom dy_glp_get_atom`

Definition at line 38 of file `gplib.h`.

#### 6.6.1.15 `#define free_atom dy_glp_free_atom`

Definition at line 39 of file `gplib.h`.

#### 6.6.1.16 `#define get_atomv dy_glp_get_atomv`

Definition at line 40 of file `gplib.h`.

#### 6.6.1.17 `#define clear_pool dy_glp_clear_pool`

Definition at line 41 of file `gplib.h`.

#### 6.6.1.18 `#define delete_pool dy_glp_delete_pool`

Definition at line 42 of file `gplib.h`.

#### 6.6.1.19 `#define insist( expr ) ((void)((expr) || (_insist(#expr, __FILE__, __LINE__), 1)))`

Definition at line 86 of file `gplib.h`.

#### 6.6.1.20 `#define align_boundary sizeof(double)`

Definition at line 100 of file `gplib.h`.

#### 6.6.1.21 `#define align_datasize( size ) (((size) + (align_boundary - 1)) / align_boundary) * align_boundary`

Definition at line 102 of file `gplib.h`.

### 6.6.2 Typedef Documentation

#### 6.6.2.1 `typedef struct ENV ENV`

Definition at line 50 of file `gplib.h`.

#### 6.6.2.2 `typedef struct MEM MEM`

Definition at line 51 of file `gplib.h`.

#### 6.6.2.3 `typedef struct POOL POOL`

Definition at line 128 of file `gplib.h`.

### 6.6.3 Function Documentation

#### 6.6.3.1 `void save_pointer ( void * ptr )`

#### 6.6.3.2 `void* read_pointer ( void )`

- 6.6.3.3 `int init_lib_env ( void )`
- 6.6.3.4 `ENV* get_env_ptr ( void )`
- 6.6.3.5 `int free_lib_env ( void )`
- 6.6.3.6 `void print ( const char * fmt, ... )`
- 6.6.3.7 `void fault ( const char * fmt, ... )`
- 6.6.3.8 `void _insist ( const char * expr, const char * file, int line )`
- 6.6.3.9 `double watch ( void )`
- 6.6.3.10 `void* umalloc ( int size )`
- 6.6.3.11 `void* ucalloc ( int nmem, int size )`
- 6.6.3.12 `void ufree ( void * ptr )`
- 6.6.3.13 `POOL* create_pool ( int size )`
- 6.6.3.14 `void* get_atom ( POOL * pool )`
- 6.6.3.15 `void free_atom ( POOL * pool, void * ptr )`
- 6.6.3.16 `void* get_atomv ( POOL * pool, int size )`
- 6.6.3.17 `void clear_pool ( POOL * pool )`
- 6.6.3.18 `void delete_pool ( POOL * pool )`

## 6.7 /home/ted/COIN/trunk/DyLP/src/DyLP/glpuf.h File Reference

### Classes

- struct [LUF](#)
- struct [LUF\\_WA](#)

### Macros

- `#define luf\_create dy_glp_luf_create`
- `#define luf\_defrag\_sva dy_glp_luf_defrag_sva`
- `#define luf\_enlarge\_row dy_glp_luf_enlarge_row`
- `#define luf\_enlarge\_col dy_glp_luf_enlarge_col`
- `#define luf\_alloc\_wa dy_glp_luf_alloc_wa`
- `#define luf\_free\_wa dy_glp_luf_free_wa`
- `#define luf\_decomp dy_glp_luf_decomp`
- `#define luf\_f\_solve dy_glp_luf_f_solve`
- `#define luf\_v\_solve dy_glp_luf_v_solve`
- `#define luf\_solve dy_glp_luf_solve`
- `#define luf\_delete dy_glp_luf_delete`

## Typedefs

- typedef struct [LUF](#) [LUF](#)
- typedef struct [LUF\\_WA](#) [LUF\\_WA](#)

## Functions

- [LUF](#) \* [luf\\_create](#) (int n, int sv\_size)
- void [luf\\_defrag\\_sva](#) ([LUF](#) \*luf)
- int [luf\\_enlarge\\_row](#) ([LUF](#) \*luf, int i, int cap)
- int [luf\\_enlarge\\_col](#) ([LUF](#) \*luf, int j, int cap)
- [LUF\\_WA](#) \* [luf\\_alloc\\_wa](#) ([LUF](#) \*luf)
- void [luf\\_free\\_wa](#) ([LUF\\_WA](#) \*wa)
- int [luf\\_decomp](#) ([LUF](#) \*luf, void \*info, int(\*col)(void \*info, int j, int rn[], double aj[]), [LUF\\_WA](#) \*wa)
- void [luf\\_f\\_solve](#) ([LUF](#) \*luf, int tr, double x[])
- void [luf\\_v\\_solve](#) ([LUF](#) \*luf, int tr, double x[])
- void [luf\\_solve](#) ([LUF](#) \*luf, int tr, double x[])
- void [luf\\_delete](#) ([LUF](#) \*luf)

### 6.7.1 Macro Definition Documentation

#### 6.7.1.1 #define [luf\\_create](#) [dy\\_glp\\_luf\\_create](#)

Definition at line 22 of file [gpluf.h](#).

#### 6.7.1.2 #define [luf\\_defrag\\_sva](#) [dy\\_glp\\_luf\\_defrag\\_sva](#)

Definition at line 23 of file [gpluf.h](#).

#### 6.7.1.3 #define [luf\\_enlarge\\_row](#) [dy\\_glp\\_luf\\_enlarge\\_row](#)

Definition at line 24 of file [gpluf.h](#).

#### 6.7.1.4 #define [luf\\_enlarge\\_col](#) [dy\\_glp\\_luf\\_enlarge\\_col](#)

Definition at line 25 of file [gpluf.h](#).

#### 6.7.1.5 #define [luf\\_alloc\\_wa](#) [dy\\_glp\\_luf\\_alloc\\_wa](#)

Definition at line 26 of file [gpluf.h](#).

#### 6.7.1.6 #define [luf\\_free\\_wa](#) [dy\\_glp\\_luf\\_free\\_wa](#)

Definition at line 27 of file [gpluf.h](#).

#### 6.7.1.7 #define [luf\\_decomp](#) [dy\\_glp\\_luf\\_decomp](#)

Definition at line 28 of file [gpluf.h](#).

#### 6.7.1.8 #define [luf\\_f\\_solve](#) [dy\\_glp\\_luf\\_f\\_solve](#)

Definition at line 29 of file [gpluf.h](#).

#### 6.7.1.9 `#define luf_v_solve dy_glp_luf_v_solve`

Definition at line 30 of file `glpluf.h`.

#### 6.7.1.10 `#define luf_solve dy_glp_luf_solve`

Definition at line 31 of file `glpluf.h`.

#### 6.7.1.11 `#define luf_delete dy_glp_luf_delete`

Definition at line 32 of file `glpluf.h`.

### 6.7.2 Typedef Documentation

#### 6.7.2.1 `typedef struct LUF LUF`

Definition at line 80 of file `glpluf.h`.

#### 6.7.2.2 `typedef struct LUF_WA LUF_WA`

Definition at line 81 of file `glpluf.h`.

### 6.7.3 Function Documentation

#### 6.7.3.1 `LUF* luf_create ( int n, int sv_size )`

#### 6.7.3.2 `void luf_defrag_sva ( LUF * luf )`

#### 6.7.3.3 `int luf_enlarge_row ( LUF * luf, int i, int cap )`

#### 6.7.3.4 `int luf_enlarge_col ( LUF * luf, int j, int cap )`

#### 6.7.3.5 `LUF_WA* luf_alloc_wa ( LUF * luf )`

#### 6.7.3.6 `void luf_free_wa ( LUF_WA * wa )`

#### 6.7.3.7 `int luf_decomp ( LUF * luf, void * info, int(*) (void *info, int j, int rn[], double aj[]) col, LUF_WA * wa )`

#### 6.7.3.8 `void luf_f_solve ( LUF * luf, int tr, double x[] )`

#### 6.7.3.9 `void luf_v_solve ( LUF * luf, int tr, double x[] )`

#### 6.7.3.10 `void luf_solve ( LUF * luf, int tr, double x[] )`

#### 6.7.3.11 `void luf_delete ( LUF * luf )`

### 6.8 `/home/ted/COIN/trunk/DyLP/src/DyLPStdLib/config_default.h` File Reference

```
#include "configall_system.h"
#include "config_dylp_default.h"
#include <float.h>
```

## Macros

- #define COIN\_DYLP\_CHECKLEVEL 0
- #define COIN\_DYLP\_VERBOSITY 0
- #define DYLP\_STATISTICS 1
- #define DYLP\_ERRMSGDIR "..\\src\\DyLP\\"
- #define DYLP\_ISFINITE \_finite
- #define DYLP\_ISNAN \_isnan

### 6.8.1 Macro Definition Documentation

#### 6.8.1.1 #define COIN\_DYLP\_CHECKLEVEL 0

Definition at line 17 of file config\_default.h.

#### 6.8.1.2 #define COIN\_DYLP\_VERBOSITY 0

Definition at line 20 of file config\_default.h.

#### 6.8.1.3 #define DYLP\_STATISTICS 1

Definition at line 38 of file config\_default.h.

#### 6.8.1.4 #define DYLP\_ERRMSGDIR "..\\src\\DyLP\\"

Definition at line 51 of file config\_default.h.

#### 6.8.1.5 #define DYLP\_ISFINITE \_finite

Definition at line 73 of file config\_default.h.

#### 6.8.1.6 #define DYLP\_ISNAN \_isnan

Definition at line 80 of file config\_default.h.

## 6.9 /home/ted/COIN/trunk/DyLP/src/DyLPStdLib/config\_dylp\_default.h File Reference

## Macros

- #define DYLP\_VERSION "trunk"
- #define DYLP\_VERSION\_MAJOR 9999
- #define DYLP\_VERSION\_MINOR 9999
- #define DYLP\_VERSION\_RELEASE 9999
- #define BOOL char

### 6.9.1 Macro Definition Documentation

#### 6.9.1.1 #define DYLP\_VERSION "trunk"

Definition at line 8 of file config\_dylp\_default.h.

#### 6.9.1.2 #define DYLP\_VERSION\_MAJOR 9999

Definition at line 11 of file config\_dylp\_default.h.

#### 6.9.1.3 `#define DYLP_VERSION_MINOR 9999`

Definition at line 14 of file `config_dylp_default.h`.

#### 6.9.1.4 `#define DYLP_VERSION_RELEASE 9999`

Definition at line 17 of file `config_dylp_default.h`.

#### 6.9.1.5 `#define BOOL char`

Definition at line 23 of file `config_dylp_default.h`.

### 6.10 `/home/ted/COIN/trunk/DyLP/src/DyLPStdLib/dylib_bnfrdr.h` File Reference

```
#include "dylib_io.h"
```

#### Classes

- struct [bnfdef\\_struct](#)
- struct [bnfGdef\\_struct](#)
- struct [bnfNPdef\\_struct](#)
- struct [bnfTdef\\_struct](#)
- struct [bnfIdef\\_struct](#)
- struct [bnfLdef\\_struct](#)
- struct [bnfLBdef\\_struct](#)
- union [bnfdef\\_any](#)
- struct [bnfref\\_struct\\_tag](#)
- struct [bnfref\\_type2](#)
- struct [bnfref\\_type3](#)
- union [bnfref\\_any](#)
- union [parse\\_any](#)

#### Macros

- `#define bnfadv 1<<0`
- `#define bnfsvnd 1<<1`
- `#define bnfsvnm 1<<2`
- `#define bnflst 1<<0`
- `#define bnfstore 1<<1`
- `#define bnfatsgn 1<<2`
- `#define bnfstbg 1<<3`
- `#define bnfflt 1<<4`
- `#define bnfcs 1<<5`
- `#define bnfmin 1<<6`
- `#define bnfsv 1<<7`
- `#define bnfexact 1<<8`
- `#define bnfdebug 1<<9`
- `#define bnfdbl 1<<10`
- `#define addrToInt(zz_addr_zz) ((int) (((char *) (zz_addr_zz)) - ((char *) (0))))`
- `#define bnfdef\_common`

- #define `bnfref_common`
- #define `NULLP` \*((char \*) 0))
- #define `mksav`(qqoff) \*((char \*) qqoff))
- #define `mkoff`(qqtype, qqfield) ((size\_t) (&((qqtype \*) 0)->qqfield))
- #define `althd`(qqnme) `bnfref_struct` \*\*qqnme[]
- #define `altcnt`(qqcnt) (`bnfref_struct` \*\*) (qqcnt)
- #define `mkaref`(qqref) (`bnfref_struct` \*\*) (qqref)
- #define `comphd`(qqnme) `bnfref_struct` \*qqnme[]
- #define `compcnt`(qqcnt) (`bnfref_struct` \*) (qqcnt)
- #define `mkcref`(qqref) (`bnfref_struct` \*) (&qqref)
- #define `gdef`(qqnme, qqsze, qqlnk, qqcomps)
- #define `npdef`(qqnme, qqalts) `bnfNPdef_struct` qqnme = { `bnfNP`, #qqnme, (`bnfref_struct` \*\*\*) qqalts }
- #define `pdef`(qqnme, qqalts) `bnfPdef_struct` qqnme = { `bnfP`, #qqnme, (`bnfref_struct` \*\*\*) qqalts }
- #define `tdef`(qqnme, qqtype, qqparm, qqval)
- #define `tqdef`(qqnme, qqschr, qqechr, qqval)
- #define `dfdef`(qqnme, qqdflds, qqnmcd, qqnm, qqsavnm, qqndcd, qqnd, qqsavnd, qqoff)
- #define `dbdef`(qqnme, qqdflds, qqnmcd, qqnm, qqsavnm, qqndcd, qqnd, qqsavnd, qqoff)
- #define `rfdef`(qqnme, qqdflds, qqnmcd, qqnm, qqsavnm, qqoff, qqndcd, qqnd, qqsavnd, qqoff2)
- #define `rbdef`(qqnme, qqdflds, qqnmcd, qqnm, qqsavnm, qqoff, qqndcd, qqnd, qqsavnd, qqoff2)
- #define `idef`(qqnme, qqval) `bnfLdef_struct` qqnme = { `bnfL`, #qqnme, (int) (qqval) }
- #define `ldef`(qqnme, qqdflds, qtxt) `bnfLdef_struct` qqnme = { `bnfL`, #qqnme, (`flags`) (qqdflds), (char \*) (qtxt) }
- #define `gref`(qqnme, qqref, qquflds, qqoff, qqsep)
- #define `npref`(qqnme, qqref, qquflds, qqsep)
- #define `pref`(qqnme, qqref, qquflds, qqoff, qqsep)
- #define `tref`(qqnme, qqref, qquflds, qqoff)
- #define `dfref`(qqnme, qqref) `bnfLBref_struct` qqnme = { `bnfDS`, #qqnme, (`bnfdef_struct` \*) &qqref, (`flags`) 0 }
- #define `dbref`(qqnme, qqref) `bnfLBref_struct` qqnme = { `bnfDL`, #qqnme, (`bnfdef_struct` \*) &qqref, (`flags`) 0 }
- #define `rfref`(qqnme, qqref) `bnfLBref_struct` qqnme = { `bnfRS`, #qqnme, (`bnfdef_struct` \*) &qqref, (`flags`) 0 }
- #define `rbref`(qqnme, qqref) `bnfLBref_struct` qqnme = { `bnfRL`, #qqnme, (`bnfdef_struct` \*) &qqref, (`flags`) 0 }
- #define `iref`(qqnme, qqref, qqoff)
- #define `lref`(qqnme, qqref) `bnfLref_struct` qqnme = { `bnfL`, #qqnme, (`bnfdef_struct` \*) &qqref, (`flags`) 0 }
- #define `dfrefdbg`(qqnme, qqref)
- #define `dbrefdbg`(qqnme, qqref)
- #define `rfrefdbg`(qqnme, qqref)
- #define `rbrefdbg`(qqnme, qqref)
- #define `lrefdbg`(qqnme, qqref)

#### Typedefs

- typedef `bnfNPdef_struct` `bnfPdef_struct`
- typedef struct `bnfref_struct_tag` `bnfref_struct`
- typedef `bnfref_struct` `bnfLBref_struct`
- typedef `bnfref_struct` `bnfLref_struct`
- typedef struct `bnfref_type2` `bnfTref_struct`
- typedef struct `bnfref_type2` `bnfLref_struct`
- typedef struct `bnfref_type3` `bnfGref_struct`
- typedef struct `bnfref_type3` `bnfNPref_struct`
- typedef struct `bnfref_type3` `bnfPref_struct`

## Enumerations

- enum `bnftype_enum` {  
    `bnfG`, `bnfNP`, `bnfP`, `bnfT`,  
    `bnfDS`, `bnfDL`, `bnfRS`, `bnfRL`,  
    `bnfI`, `bnfL` }
- enum `bnftype_enum` {  
    `bnfttNIL`, `bnfttN`, `bnfttID`, `bnfttD`,  
    `bnfttF`, `bnfttQ` }
- enum `bnfblsrc_enum` { `bnfncBNF`, `bnfncS`, `bnfncC`, `bnfncN` }

## Functions

- void `rdrinit` (void)
- void `rdrclear` (void)
- bool `parse` (ioid chn, struct `bnfref_type3` \*bnfid, `parse_any` \*result)
- void `bnfdbgctl` (ioid dbgchn, bool dbgecho, bool warnzlbl, bool numlvl, bool tablvl)
- void `prtbnfref` (ioid chn, bool echo, `bnfref_struct` \*ref)
- void `prtbnfdef` (ioid chn, bool echo, `bnfdef_struct` \*def)

### 6.10.1 Macro Definition Documentation

#### 6.10.1.1 `#define bnfadv 1<<0`

Definition at line 167 of file `dylib_bnfrdr.h`.

#### 6.10.1.2 `#define bnfsvnd 1<<1`

Definition at line 168 of file `dylib_bnfrdr.h`.

#### 6.10.1.3 `#define bnfsvnm 1<<2`

Definition at line 169 of file `dylib_bnfrdr.h`.

#### 6.10.1.4 `#define bnflst 1<<0`

Definition at line 205 of file `dylib_bnfrdr.h`.

#### 6.10.1.5 `#define bnfstore 1<<1`

Definition at line 206 of file `dylib_bnfrdr.h`.

#### 6.10.1.6 `#define bnfatsgn 1<<2`

Definition at line 207 of file `dylib_bnfrdr.h`.

#### 6.10.1.7 `#define bnfstbg 1<<3`

Definition at line 208 of file `dylib_bnfrdr.h`.

#### 6.10.1.8 `#define bnfflt 1<<4`

Definition at line 209 of file `dylib_bnfrdr.h`.



**6.10.1.9 #define bnfc5 1<<5**

Definition at line 210 of file dylib\_bnfrdr.h.

**6.10.1.10 #define bnfm6 1<<6**

Definition at line 211 of file dylib\_bnfrdr.h.

**6.10.1.11 #define bnfs7 1<<7**

Definition at line 212 of file dylib\_bnfrdr.h.

**6.10.1.12 #define bnfx8 1<<8**

Definition at line 213 of file dylib\_bnfrdr.h.

**6.10.1.13 #define bnfd9 1<<9**

Definition at line 214 of file dylib\_bnfrdr.h.

**6.10.1.14 #define bnfd10 1<<10**

Definition at line 215 of file dylib\_bnfrdr.h.

**6.10.1.15 #define addrToInt( zz\_addr\_zz ) ((int) (((char \*) (zz\_addr\_zz)) - ((char \*) (0))))**

Definition at line 225 of file dylib\_bnfrdr.h.

**6.10.1.16 #define bnfdef\_common**

**Value:**

```
bnftype_enum type ; \
 const char *name ;
```

Definition at line 263 of file dylib\_bnfrdr.h.

**6.10.1.17 #define bnfref\_common**

**Value:**

```
bnftype_enum type ; \
 const char *name ; \
 bnfdef_struct *defn ; \
 flags uflags ;
```

Definition at line 459 of file dylib\_bnfrdr.h.

**6.10.1.18 #define NULLP (\*(char \*) 0)**

Definition at line 557 of file dylib\_bnfrdr.h.

**6.10.1.19 #define mksav( qqoff ) (\*(char \*) qqoff)**

Definition at line 558 of file dylib\_bnfrdr.h.

**6.10.1.20 #define mkoff( qqtype, qqfield ) ((size\_t) (&((qqtype \*) 0)->qqfield))**

Definition at line 559 of file dylib\_bnfrdr.h.

6.10.1.21 `#define althd( qqnme ) bnfref_struct **qqnme[]`

Definition at line 576 of file `dylib_bnfdr.h`.

6.10.1.22 `#define altcnt( qqcnt ) (bnfref_struct **) (qqcnt)`

Definition at line 577 of file `dylib_bnfdr.h`.

6.10.1.23 `#define mkaref( qqref ) (bnfref_struct **) (qqref)`

Definition at line 578 of file `dylib_bnfdr.h`.

6.10.1.24 `#define comphd( qqnme ) bnfref_struct *qqnme[]`

Definition at line 580 of file `dylib_bnfdr.h`.

6.10.1.25 `#define compcnt( qqcnt ) (bnfref_struct *) (qqcnt)`

Definition at line 581 of file `dylib_bnfdr.h`.

6.10.1.26 `#define mkcref( qqref ) (bnfref_struct *) (&qqref)`

Definition at line 582 of file `dylib_bnfdr.h`.

6.10.1.27 `#define gdef( qqnme, qqsze, qqlnk, qqcomps )`

**Value:**

```
bnfGdef_struct qqnme = { bnfG, #qqnme, (int) (qqsze), (int) (qqlnk), \
 (bnfref_struct **) qqcomps }
```

Definition at line 592 of file `dylib_bnfdr.h`.

6.10.1.28 `#define npdef( qqnme, qqalts ) bnfNPdef_struct qqnme = { bnfNP, #qqnme, (bnfref_struct ***) qqalts }`

Definition at line 596 of file `dylib_bnfdr.h`.

6.10.1.29 `#define pdef( qqnme, qqalts ) bnfPdef_struct qqnme = { bnfP, #qqnme, (bnfref_struct ***) qqalts }`

Definition at line 599 of file `dylib_bnfdr.h`.

6.10.1.30 `#define tdef( qqnme, qqtype, qqparm, qqval )`

**Value:**

```
bnfTdef_struct qqnme = { bnfT, #qqnme, qqtype, '\0', '\0', \
 (int) (qqparm), (const char *) (qqval) }
```

Definition at line 602 of file `dylib_bnfdr.h`.

6.10.1.31 `#define tqdef( qqnme, qqschr, qqechr, qqval )`

**Value:**

```
bnfTdef_struct qqnme = { bnfT, #qqnme, bnfttQ, (char) qqschr, (char) qqechr, \
 0, (char *) (qqval) }
```

Definition at line 606 of file `dylib_bnfdr.h`.

6.10.1.32 **#define ddef( *qqnme, qqdflds, qqnmcd, qqnm, qqsavnm, qqndcd, qqnd, qqsavnd, qqoff* )**

**Value:**

```
bnfLBdef_struct qqnme = { bnfDS, #qqnme, (flags) (qqdflds), qqnmcd, qqndcd, \
 (int) (qqsavnm), (bnfref_struct *) &qqnm, \
 (int) (qqsavnd), (bnfref_struct *) &qqnd, \
 (int) (qqoff), 0 }
```

Definition at line 610 of file dylib\_bnfrdr.h.

6.10.1.33 **#define dbdef( *qqnme, qqdflds, qqnmcd, qqnm, qqsavnm, qqndcd, qqnd, qqsavnd, qqoff* )**

**Value:**

```
bnfLBdef_struct qqnme = { bnfDL, #qqnme, (flags) (qqdflds), qqnmcd, qqndcd, \
 (int) (qqsavnm), (bnfref_struct *) &qqnm, \
 (int) (qqsavnd), (bnfref_struct *) &qqnd, \
 (int) (qqoff), 0 }
```

Definition at line 616 of file dylib\_bnfrdr.h.

6.10.1.34 **#define rdef( *qqnme, qqdflds, qqnmcd, qqnm, qqsavnm, qqoff, qqndcd, qqnd, qqsavnd, qqoff2* )**

**Value:**

```
bnfLBdef_struct qqnme = { bnfRS, #qqnme, (flags) (qqdflds), qqnmcd, qqndcd, \
 (int) (qqsavnm), (bnfref_struct *) &qqnm, \
 (int) (qqsavnd), (bnfref_struct *) &qqnd, \
 (int) (qqoff), (int) (qqoff2) }
```

Definition at line 622 of file dylib\_bnfrdr.h.

6.10.1.35 **#define rbdef( *qqnme, qqdflds, qqnmcd, qqnm, qqsavnm, qqoff, qqndcd, qqnd, qqsavnd, qqoff2* )**

**Value:**

```
bnfLBdef_struct qqnme = { bnfRL, #qqnme, (flags) (qqdflds), qqnmcd, qqndcd, \
 (int) (qqsavnm), (bnfref_struct *) &qqnm, \
 (int) (qqsavnd), (bnfref_struct *) &qqnd, \
 (int) (qqoff), (int) (qqoff2) }
```

Definition at line 628 of file dylib\_bnfrdr.h.

6.10.1.36 **#define ideo( *qqnme, qqval* ) bnfldf\_struct qqnme = { bnfl, #qqnme, (int) (qqval) }**

Definition at line 634 of file dylib\_bnfrdr.h.

6.10.1.37 **#define ldef( *qqnme, qqdflds, qtxt* ) bnfLdef\_struct qqnme = { bnfL, #qqnme, (flags) (qqdflds), (char \*) (qtxt) }**

Definition at line 637 of file dylib\_bnfrdr.h.

6.10.1.38 **#define gref( *qqnme, qqref, qquflds, qqoff, qqsep* )**

**Value:**

```
bnfGref_struct qqnme = { bnfG, #qqnme, (bnfdef_struct *) &qqref, \
 (flags) (qquflds), (int) (qqoff), \
 (bnfref_struct *) &qqsep }
```

Definition at line 642 of file dylib\_bnfrdr.h.

6.10.1.39 **#define npref( qqnme, qqref, qqflgs, qqsep )****Value:**

```
bnfNPref_struct qqnme = { bnfNP, #qqnme, (bnfdef_struct *) &qqref, \
 (flags) (qqflgs), (int) 0, (bnfref_struct *) &qqsep }
```

Definition at line 647 of file dylib\_bnfdr.h.

6.10.1.40 **#define pref( qqnme, qqref, qqflgs, qqoff, qqsep )****Value:**

```
bnfPref_struct qqnme = { bnfP, #qqnme, (bnfdef_struct *) &qqref, \
 (flags) (qqflgs), (int) (qqoff), \
 (bnfref_struct *) &qqsep }
```

Definition at line 651 of file dylib\_bnfdr.h.

6.10.1.41 **#define tref( qqnme, qqref, qqflgs, qqoff )****Value:**

```
bnfTref_struct qqnme = { bnfT, #qqnme, (bnfdef_struct *) &qqref, \
 (flags) qqflgs, (int) qqoff }
```

Definition at line 656 of file dylib\_bnfdr.h.

6.10.1.42 **#define dtref( qqnme, qqref ) bnfLBref\_struct qqnme = { bnfDS, #qqnme, (bnfdef\_struct \*) &qqref, (flags) 0 }**

Definition at line 660 of file dylib\_bnfdr.h.

6.10.1.43 **#define dbref( qqnme, qqref ) bnfLBref\_struct qqnme = { bnfDL, #qqnme, (bnfdef\_struct \*) &qqref, (flags) 0 }**

Definition at line 663 of file dylib\_bnfdr.h.

6.10.1.44 **#define rref( qqnme, qqref ) bnfLBref\_struct qqnme = { bnfRS, #qqnme, (bnfdef\_struct \*) &qqref, (flags) 0 }**

Definition at line 666 of file dylib\_bnfdr.h.

6.10.1.45 **#define rbref( qqnme, qqref ) bnfLBref\_struct qqnme = { bnfRL, #qqnme, (bnfdef\_struct \*) &qqref, (flags) 0 }**

Definition at line 669 of file dylib\_bnfdr.h.

6.10.1.46 **#define iref( qqnme, qqref, qqoff )****Value:**

```
bnfIref_struct qqnme = { bnfI, #qqnme, (bnfdef_struct *) &qqref, \
 (flags) 0, (int) qqoff }
```

Definition at line 672 of file dylib\_bnfdr.h.

6.10.1.47 **#define lref( qqnme, qqref ) bnfLref\_struct qqnme = { bnfL, #qqnme, (bnfdef\_struct \*) &qqref, (flags) 0 }**

Definition at line 676 of file dylib\_bnfdr.h.

#### 6.10.1.48 `#define dfrefdbg( qqnme, qqref )`

##### Value:

```
bnfLBref_struct qqnme = { bnfDS, #qqnme, (bnfdef_struct *) &qqref, \
 (flags) bnfdebug }
```

Definition at line 687 of file dylib\_bnfdr.h.

#### 6.10.1.49 `#define dbrefdbg( qqnme, qqref )`

##### Value:

```
bnfLBref_struct qqnme = { bnfDL, #qqnme, (bnfdef_struct *) &qqref, \
 (flags) bnfdebug }
```

Definition at line 691 of file dylib\_bnfdr.h.

#### 6.10.1.50 `#define rfrefdbg( qqnme, qqref )`

##### Value:

```
bnfLBref_struct qqnme = { bnfRS, #qqnme, (bnfdef_struct *) &qqref, \
 (flags) bnfdebug }
```

Definition at line 695 of file dylib\_bnfdr.h.

#### 6.10.1.51 `#define rbrefdbg( qqnme, qqref )`

##### Value:

```
bnfLBref_struct qqnme = { bnfRL, #qqnme, (bnfdef_struct *) &qqref, \
 (flags) bnfdebug }
```

Definition at line 699 of file dylib\_bnfdr.h.

#### 6.10.1.52 `#define lrefdbg( qqnme, qqref )`

##### Value:

```
bnfLBref_struct qqnme = { bnfL, #qqnme, (bnfdef_struct *) &qqref, \
 (flags) bnfdebug }
```

Definition at line 703 of file dylib\_bnfdr.h.

### 6.10.2 Typedef Documentation

#### 6.10.2.1 `typedef bnfNPdef_struct bnfPdef_struct`

Definition at line 313 of file dylib\_bnfdr.h.

#### 6.10.2.2 `typedef struct bnfref_struct_tag bnfref_struct`

#### 6.10.2.3 `typedef bnfref_struct bnfLBref_struct`

Definition at line 473 of file dylib\_bnfdr.h.

#### 6.10.2.4 `typedef bnfref_struct bnfLref_struct`

Definition at line 474 of file `dylib_bnfdr.h`.

#### 6.10.2.5 `typedef struct bnfref_type2 bnfTref_struct`

Definition at line 490 of file `dylib_bnfdr.h`.

#### 6.10.2.6 `typedef struct bnfref_type2 bnflref_struct`

Definition at line 491 of file `dylib_bnfdr.h`.

#### 6.10.2.7 `typedef struct bnfref_type3 bnfGref_struct`

Definition at line 512 of file `dylib_bnfdr.h`.

#### 6.10.2.8 `typedef struct bnfref_type3 bnfNPref_struct`

Definition at line 513 of file `dylib_bnfdr.h`.

#### 6.10.2.9 `typedef struct bnfref_type3 bnfPref_struct`

Definition at line 514 of file `dylib_bnfdr.h`.

### 6.10.3 Enumeration Type Documentation

#### 6.10.3.1 `enum bnftype_enum`

Enumerator

***bnfG***  
***bnfNP***  
***bnfP***  
***bnfT***  
***bnfDS***  
***bnfDL***  
***bnfRS***  
***bnfRL***  
***bnfl***  
***bnfL***

Definition at line 107 of file `dylib_bnfdr.h`.

#### 6.10.3.2 `enum bnftt_enum`

Enumerator

***bnfttNIL***  
***bnfttN***  
***bnfttID***  
***bnfttD***  
***bnfttF***  
***bnfttQ***

Definition at line 124 of file `dylib_bnfdr.h`.

## 6.10.3.3 enum bnflblsrc\_enum

Enumerator

***bnfncBNF***

***bnfncS***

***bnfncC***

***bnfncN***

Definition at line 147 of file dylib\_bnfdr.h.

## 6.10.4 Function Documentation

6.10.4.1 void rdrinit ( void )

6.10.4.2 void rdrclear ( void )

6.10.4.3 bool parse ( ioid *chn*, struct bnfref\_type3 \* *bnfid*, parse\_any \* *result* )

6.10.4.4 void bnfdbgctl ( ioid *dbgchn*, bool *dbgecho*, bool *warnzlbl*, bool *numlvl*, bool *tablvl* )

6.10.4.5 void prtbnfref ( ioid *chn*, bool *echo*, bnfref\_struct \* *ref* )

6.10.4.6 void prtbnfdef ( ioid *chn*, bool *echo*, bnfdef\_struct \* *def* )

## 6.11 /home/ted/COIN/trunk/DyLP/src/DyLPStdLib/dylib\_errs.h File Reference

```
#include "dylib_std.h"
```

## Functions

- void [errinit](#) (const char \**emsgpath*, const char \**elogpath*, bool *errecho*)
- void [errterm](#) (void)
- void [errmsg](#) (int *errid*,...)
- void [warn](#) (int *errid*,...)

## 6.11.1 Function Documentation

6.11.1.1 void [errinit](#) ( const char \* *emsgpath*, const char \* *elogpath*, bool *errecho* )

6.11.1.2 void [errterm](#) ( void )

6.11.1.3 void [errmsg](#) ( int *errid*, ... )

6.11.1.4 void [warn](#) ( int *errid*, ... )

## 6.12 /home/ted/COIN/trunk/DyLP/src/DyLPStdLib/dylib\_fortran.h File Reference

## Macros

- #define [TRUEL](#) 1L

- `#define FALSEL 0L`
- `#define f_chr(zz_ptr, zz_ndx, zz_strsze) (*(zz_ptr+((zz_ndx)-1)*(zz_strsze)))`
- `#define f_arr1(zz_ptr, zz_ndx) (*(zz_ptr+(zz_ndx)-1))`
- `#define f_arr2(zz_ptr, zz_row, zz_col, zz_collen) (*(zz_ptr+((zz_col)-1)*(zz_collen)+((zz_row)-1)))`
- `#define ftnargINTEGER ((integer) 1)`
- `#define ftnargDOUBLE_PRECISION ((integer) 2)`
- `#define ftnargCHARACTER ((integer) 3)`
- `#define ftnargVARNAME ((integer) 4)`
- `#define ftnargCONNAME ((integer) 5)`
- `#define ftnargEND ((integer) 6)`

### Typedefs

- `typedef short int integer_2`
- `typedef long int integer`
- `typedef long int logical`
- `typedef float real`
- `typedef double double_precision`

### Variables

- `struct {`  
`integer integer_code`  
`integer double_precision_code`  
`integer character_code`  
`integer varname_code`  
`integer conname_code`  
`integer end_code`  
`} argcod_`

## 6.12.1 Macro Definition Documentation

### 6.12.1.1 `#define TRUEL 1L`

Definition at line 36 of file `dylib_fortran.h`.

### 6.12.1.2 `#define FALSEL 0L`

Definition at line 37 of file `dylib_fortran.h`.

### 6.12.1.3 `#define f_chr( zz_ptr, zz_ndx, zz_strsze ) (*(zz_ptr+((zz_ndx)-1)*(zz_strsze)))`

Definition at line 59 of file `dylib_fortran.h`.

### 6.12.1.4 `#define f_arr1( zz_ptr, zz_ndx ) (*(zz_ptr+(zz_ndx)-1))`

Definition at line 60 of file `dylib_fortran.h`.

### 6.12.1.5 `#define f_arr2( zz_ptr, zz_row, zz_col, zz_collen ) (*(zz_ptr+((zz_col)-1)*(zz_collen)+((zz_row)-1)))`

Definition at line 61 of file `dylib_fortran.h`.



#### 6.12.1.6 `#define ftnargINTEGER ((integer) 1)`

Definition at line 79 of file dylib\_fortran.h.

#### 6.12.1.7 `#define ftnargDOUBLE_PRECISION ((integer) 2)`

Definition at line 80 of file dylib\_fortran.h.

#### 6.12.1.8 `#define ftnargCHARACTER ((integer) 3)`

Definition at line 81 of file dylib\_fortran.h.

#### 6.12.1.9 `#define ftnargVARNAME ((integer) 4)`

Definition at line 82 of file dylib\_fortran.h.

#### 6.12.1.10 `#define ftnargCONNAME ((integer) 5)`

Definition at line 83 of file dylib\_fortran.h.

#### 6.12.1.11 `#define ftnargEND ((integer) 6)`

Definition at line 84 of file dylib\_fortran.h.

### 6.12.2 Typedef Documentation

#### 6.12.2.1 `typedef short int integer_2`

Definition at line 30 of file dylib\_fortran.h.

#### 6.12.2.2 `typedef long int integer`

Definition at line 31 of file dylib\_fortran.h.

#### 6.12.2.3 `typedef long int logical`

Definition at line 32 of file dylib\_fortran.h.

#### 6.12.2.4 `typedef float real`

Definition at line 33 of file dylib\_fortran.h.

#### 6.12.2.5 `typedef double double_precision`

Definition at line 34 of file dylib\_fortran.h.

### 6.12.3 Variable Documentation

#### 6.12.3.1 `integer integer_code`

Definition at line 86 of file dylib\_fortran.h.

#### 6.12.3.2 `integer double_precision_code`

Definition at line 87 of file dylib\_fortran.h.

### 6.12.3.3 integer character\_code

Definition at line 88 of file dylib\_fortran.h.

### 6.12.3.4 integer varname\_code

Definition at line 89 of file dylib\_fortran.h.

### 6.12.3.5 integer conname\_code

Definition at line 90 of file dylib\_fortran.h.

### 6.12.3.6 integer end\_code

Definition at line 91 of file dylib\_fortran.h.

### 6.12.3.7 struct { ... } argcod\_

## 6.13 /home/ted/COIN/trunk/DyLP/src/DyLPStdLib/dylib\_hash.h File Reference

### Classes

- struct [hel\\_tag](#)

### Typedefs

- typedef struct [hel\\_tag](#) [hel](#)

### Functions

- void \* [dyhash\\_lookup](#) (const char \*key, [hel](#) \*hashtab[], int size)
- void \* [dyhash\\_search](#) (const char \*key, [hel](#) \*hashtab[], int size, bool init)
- void \* [dyhash\\_enter](#) (const char \*key, [hel](#) \*hashtab[], int size, void \*entry)
- void \* [dyhash\\_erase](#) (const char \*key, [hel](#) \*hashtab[], int size)

### 6.13.1 Typedef Documentation

#### 6.13.1.1 typedef struct [hel\\_tag](#) [hel](#)

### 6.13.2 Function Documentation

#### 6.13.2.1 void\* [dyhash\\_lookup](#) ( const char \* *key*, [hel](#) \* *hashtab*[], int *size* )

#### 6.13.2.2 void \* [dyhash\\_search](#) ( const char \* *key*, [hel](#) \* *hashtab*[], int *size*, bool *init* )

#### 6.13.2.3 void \* [dyhash\\_enter](#) ( const char \* *key*, [hel](#) \* *hashtab*[], int *size*, void \* *entry* )

#### 6.13.2.4 void \* [dyhash\\_erase](#) ( const char \* *key*, [hel](#) \* *hashtab*[], int *size* )

## 6.14 /home/ted/COIN/trunk/DyLP/src/DyLPStdLib/dylib\_io.h File Reference

```
#include "dylib_std.h"
```

## Classes

- struct [lex\\_struct](#)

## Macros

- `#define IOID\_NOSTRM ((ioid) 0)`
- `#define IOID\_INV ((ioid) -1)`

## Typedefs

- typedef int [ioid](#)

## Enumerations

- enum [lexclass](#) {  
    [DY\\_LCNIL](#), [DY\\_LCNUM](#), [DY\\_LCID](#), [DY\\_LCDEL](#),  
    [DY\\_LCFS](#), [DY\\_LCQS](#), [DY\\_LCEOF](#), [DY\\_LCERR](#) }

## Functions

- [bool](#) [dyio\\_ioint](#) (void)
- void [dyio\\_ioterm](#) (void)
- [ioid](#) [dyio\\_openfile](#) (const char \*path, const char \*mode)
- [bool](#) [dyio\\_isactive](#) ([ioid](#) id)
- [bool](#) [dyio\\_closefile](#) ([ioid](#) id)
- [bool](#) [dyio\\_setmode](#) ([ioid](#) id, char mode)
- [bool](#) [dyio\\_ttyq](#) ([ioid](#) id)
- [bool](#) [dyio\\_chgerrlog](#) (const char \*path, [bool](#) echo)
- const char \* [dyio\\_idtopath](#) ([ioid](#) id)
- [ioid](#) [dyio\\_paththeid](#) (const char \*path, const char \*mode)
- long [dyio\\_mark](#) ([ioid](#) id)
- [bool](#) [dyio\\_backup](#) ([ioid](#) id, long there)
- [bool](#) [dyio\\_scan](#) ([ioid](#) id, const char pattern[], [bool](#) rwnd, [bool](#) wrap)
- [lex\\_struct](#) \* [dyio\\_scanlex](#) ([ioid](#) id)
- [lex\\_struct](#) \* [dyio\\_scanstr](#) ([ioid](#) id, [lexclass](#) stype, int fslen, char qschr, char qechr)
- void [dyio\\_flushio](#) ([ioid](#) id, [bool](#) echo)
- void [dyio\\_outfmt](#) ([ioid](#) id, [bool](#) echo, const char \*pattern,...)
- void [dyio\\_outchr](#) ([ioid](#) id, [bool](#) echo, char chr)
- int [dyio\\_outfxd](#) (char \*buffer, int fldsize, char lcr, const char \*pattern,...)

### 6.14.1 Macro Definition Documentation

#### 6.14.1.1 `#define IOID\_NOSTRM ((ioid) 0)`

Definition at line 41 of file [dylib\\_io.h](#).

#### 6.14.1.2 `#define IOID\_INV ((ioid) -1)`

Definition at line 42 of file [dylib\\_io.h](#).

### 6.14.2 Typedef Documentation

#### 6.14.2.1 typedef int ioid

Definition at line 39 of file dylib\_io.h.

### 6.14.3 Enumeration Type Documentation

#### 6.14.3.1 enum lexclass

Enumerator

***DY\_LCNIL***  
***DY\_LCNUM***  
***DY\_LCID***  
***DY\_LCDEL***  
***DY\_LCFS***  
***DY\_LCQS***  
***DY\_LCEOF***  
***DY\_LCERR***

Definition at line 67 of file dylib\_io.h.

### 6.14.4 Function Documentation

#### 6.14.4.1 bool dyio\_ioint ( void )

#### 6.14.4.2 void dyio\_ioterm ( void )

#### 6.14.4.3 ioid dyio\_openfile ( const char \* *path*, const char \* *mode* )

#### 6.14.4.4 bool dyio\_isactive ( ioid *id* )

#### 6.14.4.5 bool dyio\_closefile ( ioid *id* )

#### 6.14.4.6 bool dyio\_setmode ( ioid *id*, char *mode* )

#### 6.14.4.7 bool dyio\_ttyq ( ioid *id* )

#### 6.14.4.8 bool dyio\_chgerrlog ( const char \* *path*, bool *echo* )

#### 6.14.4.9 const char\* dyio\_idtopath ( ioid *id* )

#### 6.14.4.10 ioid dyio\_pathmoid ( const char \* *path*, const char \* *mode* )

#### 6.14.4.11 long dyio\_mark ( ioid *id* )

#### 6.14.4.12 bool dyio\_backup ( ioid *id*, long *there* )

#### 6.14.4.13 bool dyio\_scan ( ioid *id*, const char *pattern*[], bool *rwnd*, bool *wrap* )

#### 6.14.4.14 lex\_struct\* dyio\_scanlex ( ioid *id* )

6.14.4.15 `lex_struct * dyio_scanstr ( ioid id, lexclass stype, int fslen, char qschr, char qechr )`

6.14.4.16 `void dyio_flushio ( ioid id, bool echo )`

6.14.4.17 `void dyio_outfmt ( ioid id, bool echo, const char * pattern, ... )`

6.14.4.18 `void dyio_outchr ( ioid id, bool echo, char chr )`

6.14.4.19 `int dyio_outfxd ( char * buffer, int fldsize, char lcr, const char * pattern, ... )`

## 6.15 /home/ted/COIN/trunk/DyLP/src/DyLPStdLib/dylib\_keytab.h File Reference

### Classes

- struct [keytab\\_entry\\_internal](#)

### Typedefs

- typedef struct  
[keytab\\_entry\\_internal](#) [keytab\\_entry](#)

### Functions

- int [find](#) (char \**word*, [keytab\\_entry](#) *keytab*[], int *numkeys*)
- int [ambig](#) (char \**word*, [keytab\\_entry](#) *keytab*[], int *numkeys*)

#### 6.15.1 Typedef Documentation

6.15.1.1 typedef struct [keytab\\_entry\\_internal](#) [keytab\\_entry](#)

#### 6.15.2 Function Documentation

6.15.2.1 int [find](#) ( char \* *word*, [keytab\\_entry](#) *keytab*[], int *numkeys* )

6.15.2.2 int [ambig](#) ( char \* *word*, [keytab\\_entry](#) *keytab*[], int *numkeys* )

## 6.16 /home/ted/COIN/trunk/DyLP/src/DyLPStdLib/dylib\_std.h File Reference

```
#include <stddef.h>
#include <stdlib.h>
#include "DyLPConfig.h"
#include <string.h>
```

### Classes

- struct [lnk\\_struct\\_tag](#)

### Macros

- #define [UNUSED](#)

- `#define FALSE 0`
- `#define TRUE 1`
- `#define setflg(zz_flg, zz_flg) ((zz_flg) |= (zz_flg))`
- `#define clrflg(zz_flg, zz_flg) ((zz_flg) &= ~(zz_flg))`
- `#define comflg(zz_flg, zz_flg) ((zz_flg) ^= (zz_flg))`
- `#define getflg(zz_flg, zz_flg) ((zz_flg) & (zz_flg))`
- `#define flgon(zz_flg, zz_flg) ((zz_flg) & (zz_flg)?TRUE:FALSE)`
- `#define flgoff(zz_flg, zz_flg) ((zz_flg) & (zz_flg)?FALSE:TRUE)`
- `#define flgall(zz_flg, zz_flg) (((zz_flg) & (zz_flg)) == (zz_flg)?TRUE:FALSE)`
- `#define lnk_in(qqlnk, qqval) ((qqlnk)->llval = (void *) (qqval))`
- `#define lnk_out(qqlnk, qqtype) ((qqtype) (qqlnk)->llval)`
- `#define minn(qa, qb) (((qa) > (qb))?(qb):(qa))`
- `#define maxx(qa, qb) (((qa) > (qb))?(qa):(qb))`
- `#define MALLOC_DBG_INIT(chn)`
- `#define MALLOC(zz_sze_zz) malloc(zz_sze_zz)`
- `#define CALLOC(zz_cnt_zz, zz_sze_zz) calloc(zz_cnt_zz,zz_sze_zz)`
- `#define REALLOC(zz_rptr_zz, zz_sze_zz) realloc(zz_rptr_zz,zz_sze_zz)`
- `#define FREE(zz_fptr_zz) free((void *) zz_fptr_zz)`

#### Typedefs

- `typedef BOOL bool`
- `typedef unsigned int flags`
- `typedef struct lnk_struct_tag lnk_struct`

#### 6.16.1 Macro Definition Documentation

##### 6.16.1.1 `#define UNUSED`

Definition at line 44 of file dylib\_std.h.

##### 6.16.1.2 `#define FALSE 0`

Definition at line 64 of file dylib\_std.h.

##### 6.16.1.3 `#define TRUE 1`

Definition at line 65 of file dylib\_std.h.

##### 6.16.1.4 `#define setflg( zz_flg, zz_flg ) ((zz_flg) |= (zz_flg))`

Definition at line 97 of file dylib\_std.h.

##### 6.16.1.5 `#define clrflg( zz_flg, zz_flg ) ((zz_flg) &= ~(zz_flg))`

Definition at line 98 of file dylib\_std.h.

##### 6.16.1.6 `#define comflg( zz_flg, zz_flg ) ((zz_flg) ^= (zz_flg))`

Definition at line 99 of file dylib\_std.h.

##### 6.16.1.7 `#define getflg( zz_flg, zz_flg ) ((zz_flg) & (zz_flg))`

Definition at line 100 of file dylib\_std.h.

6.16.1.8 `#define flgon( zz_flg, zz_flg ) ((zz_flg)&(zz_flg)?TRUE:FALSE)`

Definition at line 101 of file dylib\_std.h.

6.16.1.9 `#define flgoff( zz_flg, zz_flg ) ((zz_flg)&(zz_flg)?FALSE:TRUE)`

Definition at line 102 of file dylib\_std.h.

6.16.1.10 `#define flgall( zz_flg, zz_flg ) (((zz_flg)&(zz_flg)) == (zz_flg))?TRUE:FALSE)`

Definition at line 103 of file dylib\_std.h.

6.16.1.11 `#define lnk_in( qqlnk, qqval ) ((qqlnk)->llval = (void *) (qqval))`

Definition at line 119 of file dylib\_std.h.

6.16.1.12 `#define lnk_out( qqlnk, qqtype ) ((qqtype) (qqlnk)->llval)`

Definition at line 120 of file dylib\_std.h.

6.16.1.13 `#define minn( qa, qb ) (((qa) > (qb))?(qb):(qa))`

Definition at line 125 of file dylib\_std.h.

6.16.1.14 `#define maxx( qa, qb ) (((qa) > (qb))?(qa):(qb))`

Definition at line 126 of file dylib\_std.h.

6.16.1.15 `#define MALLOC_DBG_INIT( chn )`

Definition at line 207 of file dylib\_std.h.

6.16.1.16 `#define MALLOC( zz_sze_zz ) malloc(zz_sze_zz)`

Definition at line 209 of file dylib\_std.h.

6.16.1.17 `#define CALLOC( zz_cnt_zz, zz_sze_zz ) calloc(zz_cnt_zz,zz_sze_zz)`

Definition at line 211 of file dylib\_std.h.

6.16.1.18 `#define REALLOC( zz_rptr_zz, zz_sze_zz ) realloc(zz_rptr_zz,zz_sze_zz)`

Definition at line 213 of file dylib\_std.h.

6.16.1.19 `#define FREE( zz_fptr_zz ) free((void *) zz_fptr_zz)`

Definition at line 215 of file dylib\_std.h.

## 6.16.2 Typedef Documentation

### 6.16.2.1 typedef **BOOL** **bool**

Definition at line 67 of file dylib\_std.h.

### 6.16.2.2 typedef **unsigned int** **flags**

Definition at line 95 of file dylib\_std.h.

### 6.16.2.3 typedef struct Ink\_struct\_tag Ink\_struct

## 6.17 /home/ted/COIN/trunk/DyLP/src/DyLPStdLib/dylib\_strtns.h File Reference

```
#include "dylib_std.h"
```

### Macros

- #define [STRALLOC](#)(zz\_sptr\_zz) [stralloc](#)(zz\_sptr\_zz)
- #define [STRFREE](#)(zz\_fptr\_zz) [strfree](#)(zz\_fptr\_zz)

### Functions

- int [cistrcmp](#) (const char \*str1, const char \*str2)
- int [cimstrcmp](#) (const char \*str1, const char \*str2)
- int [mstrcmp](#) (const char \*str1, const char \*str2)
- char \* [strsave](#) (char \*original)
- const char \* [stralloc](#) (const char \*string)
- bool [strfree](#) (const char \*string)

### 6.17.1 Macro Definition Documentation

#### 6.17.1.1 #define STRALLOC( zz\_sptr\_zz ) stralloc(zz\_sptr\_zz)

Definition at line 63 of file dylib\_strtns.h.

#### 6.17.1.2 #define STRFREE( zz\_fptr\_zz ) strfree(zz\_fptr\_zz)

Definition at line 65 of file dylib\_strtns.h.

### 6.17.2 Function Documentation

#### 6.17.2.1 int cistrcmp ( const char \* str1, const char \* str2 )

#### 6.17.2.2 int cimstrcmp ( const char \* str1, const char \* str2 )

#### 6.17.2.3 int mstrcmp ( const char \* str1, const char \* str2 )

#### 6.17.2.4 char\* strsave ( char \* original )

#### 6.17.2.5 const char\* stralloc ( const char \* string )

#### 6.17.2.6 bool strfree ( const char \* string )

## 6.18 /home/ted/COIN/trunk/DyLP/src/DyLPStdLib/DyLPConfig.h File Reference

```
#include "config_dylp_default.h"
```



## 6.19 /home/ted/COIN/trunk/DyLP/src/OsiDyLP/OsiDyLPMessages.hpp File Reference

```
#include "CoinMessageHandler.hpp"
```

### Enumerations

- enum [OsiDyLPMessageID\\_enum](#) {  
    [ODSI\\_TEST\\_MSG](#), [ODSI\\_MPSFILEIO](#), [ODSI\\_UNSUPFORCEDO](#), [ODSI\\_IGNOREDHINT](#),  
    [ODSI\\_EMPTYODWSB](#), [ODSI\\_NOTODWSB](#), [ODSI\\_ODWSBBADSIZE](#), [ODSI\\_ODWSBBADSTATUS](#),  
    [ODSI\\_ODWSBSHORTBASIS](#), [ODSI\\_CWSBREJECT](#), [ODSI\\_PRE SOL\\_STATS](#), [ODSI\\_PRE SOL\\_PASS](#),  
    [ODSI\\_POSTSOL](#), [ODSI\\_POSTSOL\\_ACT](#), [ODSI\\_COLD](#), [ODSI\\_WARM](#),  
    [ODSI\\_HOT](#), [ODSI\\_ALLDYLP](#), [ODSI\\_ATTACH](#), [ODSI\\_DETACH](#),  
    [ODSI\\_NOSOLVE](#), [ODSI\\_FAILEDCALL](#), [ODSI\\_ACCESS\\_STALE](#), [ODSI\\_SHORTSTATS](#),  
    [ODSI\\_CONFUSION](#), [ODSI\\_TABLEAU\\_INIT\\_FAIL](#), [ODSI\\_NOTOWNER](#), [ODSI\\_NOTOPTIMAL](#),  
    [ODSI\\_NOTVALID](#), [ODSI\\_NOTFULLSYS](#), [ODSI\\_NOTSIMPLEX](#), [ODSI\\_BADSTATE](#),  
    [ODSI\\_BADACTIVEBASIS](#), [ODSI\\_DUMMY\\_END](#) }

*Copyright (C) 2004 – 2007 Lou Hafer, International Business Machines Corporation and others.*

### 6.19.1 Enumeration Type Documentation

#### 6.19.1.1 enum [OsiDyLPMessageID\\_enum](#)

Copyright (C) 2004 – 2007 Lou Hafer, International Business Machines Corporation and others.

All Rights Reserved.

This file is a portion of the COIN/OSI interface for dyLP and is licensed under the terms of the Eclipse Public License (EPL)

### Enumerator

**[ODSI\\_TEST\\_MSG](#)**  
**[ODSI\\_MPSFILEIO](#)**  
**[ODSI\\_UNSUPFORCEDO](#)**  
**[ODSI\\_IGNOREDHINT](#)**  
**[ODSI\\_EMPTYODWSB](#)**  
**[ODSI\\_NOTODWSB](#)**  
**[ODSI\\_ODWSBBADSIZE](#)**  
**[ODSI\\_ODWSBBADSTATUS](#)**  
**[ODSI\\_ODWSBSHORTBASIS](#)**  
**[ODSI\\_CWSBREJECT](#)**  
**[ODSI\\_PRE SOL\\_STATS](#)**  
**[ODSI\\_PRE SOL\\_PASS](#)**  
**[ODSI\\_POSTSOL](#)**  
**[ODSI\\_POSTSOL\\_ACT](#)**  
**[ODSI\\_COLD](#)**  
**[ODSI\\_WARM](#)**  
**[ODSI\\_HOT](#)**

**ODSI\_ALLDYLP**  
**ODSI\_ATTACH**  
**ODSI\_DETACH**  
**ODSI\_NOSOLVE**  
**ODSI\_FAILEDCALL**  
**ODSI\_ACCESS\_STALE**  
**ODSI\_SHORTSTATS**  
**ODSI\_CONFUSION**  
**ODSI\_TABLEAU\_INIT\_FAIL**  
**ODSI\_NOTOWNER**  
**ODSI\_NOTOPTIMAL**  
**ODSI\_NOTVALID**  
**ODSI\_NOTFULLSYS**  
**ODSI\_NOTSIMPLEX**  
**ODSI\_BADSTATE**  
**ODSI\_BDACTIVEBASIS**  
**ODSI\_DUMMY\_END**

Definition at line 28 of file OsiDyIpMessages.hpp.

## 6.20 /home/ted/COIN/trunk/DyLP/src/OsiDyIp/OsiDyIpSolverInterface.hpp File Reference

Declarations of the COIN OSI API for the dylp solver.

```

#include "OsiConfig.h"
#include <CoinPackedMatrix.hpp>
#include <OsiSolverInterface.hpp>
#include <CoinWarmStart.hpp>
#include <CoinMessageHandler.hpp>
#include <CoinMpsIO.hpp>
#include <CoinPresolveMatrix.hpp>
#include "dylp.h"

```

### Classes

- class [OsiDyIpSolverInterface](#)  
COIN OSI API for dylp.

### Macros

- #define [DYLP\\_INTERNAL](#)

### Enumerations

- enum [ODSI\\_start\\_enum](#) { [startInvalid](#) = 0, [startCold](#) = 1, [startWarm](#), [startHot](#) }  
Enum to specify cold/warm/hot start.

### 6.20.1 Detailed Description

Declarations of the COIN OSI API for the dylp solver. This file contains the declaration of the class [OsiDyIpSolverInterface](#) (ODSI), an implementation of the COIN OSI API for the dylp LP solver. The documentation here most often provides only brief descriptions of methods. See the OsiSolverInterface documentation for additional details.

Definition in file [OsiDyIpSolverInterface.hpp](#).

### 6.20.2 Macro Definition Documentation

#### 6.20.2.1 #define DYLP\_INTERNAL

Definition at line 36 of file OsiDyIpSolverInterface.hpp.

### 6.20.3 Enumeration Type Documentation

#### 6.20.3.1 enum ODSI\_start\_enum

Enum to specify cold/warm/hot start.

#### Enumerator

***startInvalid***  
***startCold***  
***startWarm***  
***startHot***

Definition at line 45 of file OsiDyIpSolverInterface.hpp.

## 6.21 /home/ted/COIN/trunk/DyLP/src/OsiDyIp/OsiDyIpWarmStartBasis.hpp File Reference

Copyright (C) 2003 – 2007 Lou Hafer, International Business Machines Corporation and others.

```
#include "CoinWarmStartBasis.hpp"
#include "dylp.h"
```

#### Classes

- class [OsiDyIpWarmStartBasis](#)  
*The dylp warm start class.*
- class [OsiDyIpWarmStartBasisDiff](#)  
*A 'diff' between two [OsiDyIpWarmStartBasis](#) objects.*

#### Macros

- #define [DYLP\\_INTERNAL](#)

### 6.21.1 Detailed Description

Copyright (C) 2003 – 2007 Lou Hafer, International Business Machines Corporation and others. All Rights Reserved.

This file is a portion of the COIN/OSI interface for dylp and is licensed under the terms of the Eclipse Public License (EPL)

Declaration of the warm start class for dylp.

Definition in file [OsiDylpWarmStartBasis.hpp](#).

### 6.21.2 Macro Definition Documentation

#### 6.21.2.1 #define DYLP\_INTERNAL

Definition at line 26 of file OsiDylpWarmStartBasis.hpp.

## Index

### Symbols

~OsiDyIpSolverInterface

OsiDyIpSolverInterface, [57](#)

~OsiDyIpWarmStartBasis

OsiDyIpWarmStartBasis, [69](#)

~OsiDyIpWarmStartBasisDiff

OsiDyIpWarmStartBasisDiff, [72](#)

/home/ted/COIN/trunk/DyLP/src/DyIp/dy\_cmdint.h, [76](#)

/home/ted/COIN/trunk/DyLP/src/DyIp/dy\_consyst.h, [77](#)

/home/ted/COIN/trunk/DyLP/src/DyIp/dy\_vector.h, [85](#)

/home/ted/COIN/trunk/DyLP/src/DyIp/dyIp.h, [86](#)

/home/ted/COIN/trunk/DyLP/src/DyIp/glpinv.h, [95](#)

/home/ted/COIN/trunk/DyLP/src/DyIp/glpIib.h, [96](#)

/home/ted/COIN/trunk/DyLP/src/DyIp/glpIuf.h, [100](#)

/home/ted/COIN/trunk/DyLP/src/DyIpStdLib/DyIpConfig.h,  
[122](#)

/home/ted/COIN/trunk/DyLP/src/DyIpStdLib/config\_-  
default.h, [102](#)

/home/ted/COIN/trunk/DyLP/src/DyIpStdLib/config\_dyIp\_-  
default.h, [103](#)

/home/ted/COIN/trunk/DyLP/src/DyIpStdLib/dyIib\_bnfdr.-  
h, [104](#)

/home/ted/COIN/trunk/DyLP/src/DyIpStdLib/dyIib\_errs.h,  
[113](#)

/home/ted/COIN/trunk/DyLP/src/DyIpStdLib/dyIib\_fortran.-  
h, [113](#)

/home/ted/COIN/trunk/DyLP/src/DyIpStdLib/dyIib\_hash.h,  
[116](#)

/home/ted/COIN/trunk/DyLP/src/DyIpStdLib/dyIib\_io.h,  
[116](#)

/home/ted/COIN/trunk/DyLP/src/DyIpStdLib/dyIib\_keytab.-  
h, [119](#)

/home/ted/COIN/trunk/DyLP/src/DyIpStdLib/dyIib\_std.h,  
[119](#)

/home/ted/COIN/trunk/DyLP/src/DyIpStdLib/dyIib\_strtns.-  
h, [122](#)

/home/ted/COIN/trunk/DyLP/src/OsiDyIp/OsiDyIpMessages.-  
hpp, [123](#)

/home/ted/COIN/trunk/DyLP/src/OsiDyIp/OsiDyIpSolver-  
Interface.hpp, [124](#)

/home/ted/COIN/trunk/DyLP/src/OsiDyIp/OsiDyIpWarm-  
StartBasis.hpp, [125](#)

\_insist

glpIib.h, [98](#), [100](#)

### A

actcnt

lpstats\_struct, [40](#)

activateRowCutDebugger

OsiDyIpSolverInterface, [65](#)

active

lpopts\_struct, [34](#)

actlim

lpopts\_struct, [32](#)

actIvI

lpopts\_struct, [32](#)

actvars

lpprob\_struct, [38](#)

addCol

OsiDyIpSolverInterface, [61](#)

addRow

OsiDyIpSolverInterface, [61](#)

addrToInt

dyIib\_bnfdr.h, [107](#)

addvar

lpopts\_struct, [33](#)

align\_boundary

glpIib.h, [99](#)

align\_datasize

glpIib.h, [99](#)

allownopiv

lpopts\_struct, [32](#)

altcnt

dyIib\_bnfdr.h, [108](#)

althd

dyIib\_bnfdr.h, [107](#)

alts

bnfNPdef\_struct, [15](#)

ambig

dyIib\_keytab.h, [119](#)

angle

lpstats\_struct, [40](#), [41](#)

applyColCut

OsiDyIpSolverInterface, [62](#)

applyDiff

OsiDyIpWarmStartBasis, [70](#)

applyRowCut

OsiDyIpSolverInterface, [61](#)

archcnt

consys\_struct, [23](#)

archvcnt

consys\_struct, [23](#)

argcod\_

dyIib\_fortran.h, [116](#)

assignBasisStatus

OsiDyIpWarmStartBasis, [71](#)

assignProblem

OsiDyIpSolverInterface, [58](#)

attvecs

consys\_struct, [25](#)

attvhdr\_struct

dy\_consyst.h, [82](#)

- attvhdr\_struct\_tag, 9
  - elsze, 9
  - nxt, 9
  - pveclst, 9
  - vec, 9
  - what, 9
- avail
  - POOL, 75
- avgpivs
  - lpstats\_struct, 41
- avgsiz
  - lpstats\_struct, 43
- B
- BOOL
  - config\_dylp\_default.h, 104
- balance
  - OsiDyIpSolverInterface, 67
- basis
  - lpopts\_struct, 36
  - lpprob\_struct, 37
  - OsiDyIpSolverInterface, 66
- basis\_struct, 9
  - el, 10
  - len, 10
- basisIsAvailable
  - OsiDyIpSolverInterface, 65
- basisel\_struct, 10
  - cndx, 10
  - vndx, 10
- big\_v
  - LUF, 49
- binvcnt
  - consys\_struct, 23
- bnd
  - conbnd\_struct, 20
- bnfDL
  - dylib\_bnfrdr.h, 112
- bnfDS
  - dylib\_bnfrdr.h, 112
- bnfG
  - dylib\_bnfrdr.h, 112
- bnfI
  - dylib\_bnfrdr.h, 112
- bnfL
  - dylib\_bnfrdr.h, 112
- bnfNP
  - dylib\_bnfrdr.h, 112
- bnfP
  - dylib\_bnfrdr.h, 112
- bnfRL
  - dylib\_bnfrdr.h, 112
- bnfRS
  - dylib\_bnfrdr.h, 112
- bnfT
  - dylib\_bnfrdr.h, 112
- bnfGdef\_struct, 12
  - comps, 12
  - link, 12
  - size, 12
- bnfGref\_struct
  - dylib\_bnfrdr.h, 112
- bnfIdef\_struct, 12
  - ival, 13
- bnfIref\_struct
  - dylib\_bnfrdr.h, 112
- bnfLBdef\_struct, 13
  - dflgs, 13
  - ndcd, 13
  - ndsrc, 14
  - nmcd, 13
  - nmsrc, 14
  - offset, 14
  - offset2, 14
  - savnd, 14
  - savnm, 13
- bnfLBref\_struct
  - dylib\_bnfrdr.h, 111
- bnfLdef\_struct, 14
  - dflgs, 14
  - txt, 14
- bnfLref\_struct
  - dylib\_bnfrdr.h, 111
- bnfNPdef\_struct, 15
  - alts, 15
- bnfNPref\_struct
  - dylib\_bnfrdr.h, 112
- bnfPdef\_struct
  - dylib\_bnfrdr.h, 111
- bnfPref\_struct
  - dylib\_bnfrdr.h, 112
- bnfTdef\_struct, 18
  - parm1, 18
  - qechr, 18
  - qschr, 18
  - ttype, 18
  - val, 18
- bnfTref\_struct
  - dylib\_bnfrdr.h, 112
- bnfadv
  - dylib\_bnfrdr.h, 106
- bnfatsgn
  - dylib\_bnfrdr.h, 106
- bnfcs
  - dylib\_bnfrdr.h, 106
- bnfdbgctl
  - dylib\_bnfrdr.h, 113
- bnfdbI

- [dylib\\_bnfdr.h](#), [107](#)
- [bnfdebug](#)
  - [dylib\\_bnfdr.h](#), [107](#)
- [bnfdef\\_any](#), [11](#)
  - [com](#), [11](#)
  - [G](#), [11](#)
  - [I](#), [11](#)
  - [L](#), [11](#)
  - [LB](#), [11](#)
  - [NP](#), [11](#)
  - [P](#), [11](#)
  - [T](#), [11](#)
- [bnfdef\\_common](#)
  - [dylib\\_bnfdr.h](#), [107](#)
- [bnfdef\\_struct](#), [12](#)
- [bnfexact](#)
  - [dylib\\_bnfdr.h](#), [107](#)
- [bnfflt](#)
  - [dylib\\_bnfdr.h](#), [106](#)
- [bnfblsrc\\_enum](#)
  - [dylib\\_bnfdr.h](#), [112](#)
- [bnflst](#)
  - [dylib\\_bnfdr.h](#), [106](#)
- [bnfmin](#)
  - [dylib\\_bnfdr.h](#), [107](#)
- [bnfncBNF](#)
  - [dylib\\_bnfdr.h](#), [113](#)
- [bnfncC](#)
  - [dylib\\_bnfdr.h](#), [113](#)
- [bnfncN](#)
  - [dylib\\_bnfdr.h](#), [113](#)
- [bnfncS](#)
  - [dylib\\_bnfdr.h](#), [113](#)
- [bnfref\\_any](#), [15](#)
  - [com](#), [15](#)
  - [G](#), [16](#)
  - [I](#), [16](#)
  - [L](#), [16](#)
  - [LB](#), [16](#)
  - [NP](#), [16](#)
  - [P](#), [16](#)
  - [T](#), [16](#)
  - [t1](#), [15](#)
  - [t2](#), [16](#)
  - [t3](#), [16](#)
- [bnfref\\_common](#)
  - [dylib\\_bnfdr.h](#), [107](#)
- [bnfref\\_struct](#)
  - [dylib\\_bnfdr.h](#), [111](#)
- [bnfref\\_struct\\_tag](#), [16](#)
- [bnfref\\_type2](#), [17](#)
  - [offset](#), [17](#)
- [bnfref\\_type3](#), [17](#)
  - [offset](#), [17](#)
- [sep](#), [17](#)
- [bnfstbg](#)
  - [dylib\\_bnfdr.h](#), [106](#)
- [bnfststore](#)
  - [dylib\\_bnfdr.h](#), [106](#)
- [bnfstv](#)
  - [dylib\\_bnfdr.h](#), [107](#)
- [bnfstvnd](#)
  - [dylib\\_bnfdr.h](#), [106](#)
- [bnfstvnm](#)
  - [dylib\\_bnfdr.h](#), [106](#)
- [bnfttD](#)
  - [dylib\\_bnfdr.h](#), [112](#)
- [bnfttF](#)
  - [dylib\\_bnfdr.h](#), [112](#)
- [bnfttID](#)
  - [dylib\\_bnfdr.h](#), [112](#)
- [bnfttN](#)
  - [dylib\\_bnfdr.h](#), [112](#)
- [bnfttNIL](#)
  - [dylib\\_bnfdr.h](#), [112](#)
- [bnfttQ](#)
  - [dylib\\_bnfdr.h](#), [112](#)
- [bnfttype\\_enum](#)
  - [dylib\\_bnfdr.h](#), [112](#)
- [bnftype\\_enum](#)
  - [dylib\\_bnfdr.h](#), [112](#)
- [bogus](#)
  - [lptols\\_struct](#), [45](#)
- [bool](#)
  - [dylib\\_std.h](#), [121](#)
- [branchAndBound](#)
  - [OsiDyIpsolverInterface](#), [66](#)
- [C](#)
- [c](#)
  - [parse\\_any](#), [73](#)
- [CALLOC](#)
  - [dylib\\_std.h](#), [121](#)
- [CONSYS\\_CLB](#)
  - [dy\\_consys.h](#), [80](#)
- [CONSYS\\_COL](#)
  - [dy\\_consys.h](#), [80](#)
- [CONSYS\\_COLHDR](#)
  - [dy\\_consys.h](#), [81](#)
- [CONSYS\\_COLVEC](#)
  - [dy\\_consys.h](#), [81](#)
- [CONSYS\\_CORRUPT](#)
  - [dy\\_consys.h](#), [82](#)
- [CONSYS\\_CSCALE](#)
  - [dy\\_consys.h](#), [81](#)
- [CONSYS\\_CTYP](#)
  - [dy\\_consys.h](#), [80](#)
- [CONSYS\\_CUB](#)

- dy\_consys.h, [80](#)
- CONSYS\_FININF
  - dy\_consys.h, [82](#)
- CONSYS\_LVARS
  - dy\_consys.h, [82](#)
- CONSYS\_MAXBUFLEN
  - dy\_consys.h, [82](#)
- CONSYS\_MTX
  - dy\_consys.h, [80](#)
- CONSYS\_OBJ
  - dy\_consys.h, [80](#)
- CONSYS\_RHS
  - dy\_consys.h, [80](#)
- CONSYS\_RHSLOW
  - dy\_consys.h, [80](#)
- CONSYS\_ROW
  - dy\_consys.h, [80](#)
- CONSYS\_ROWHDR
  - dy\_consys.h, [81](#)
- CONSYS\_ROWVEC
  - dy\_consys.h, [81](#)
- CONSYS\_RSCALE
  - dy\_consys.h, [81](#)
- CONSYS\_VLB
  - dy\_consys.h, [80](#)
- CONSYS\_VTYP
  - dy\_consys.h, [80](#)
- CONSYS\_VUB
  - dy\_consys.h, [80](#)
- CONSYS\_WRNATT
  - dy\_consys.h, [82](#)
- CONSYS\_WRNZERO
  - dy\_consys.h, [82](#)
- canDoSimplexInterface
  - OsiDyIpsolverInterface, [64](#)
- cands
  - lpstats\_struct, [42](#)
- cc\_len
  - INV, [28](#)
- cc\_ndx
  - INV, [28](#)
- cc\_val
  - INV, [28](#)
- character\_code
  - dylib\_fortran.h, [115](#)
- check
  - lpopts\_struct, [32](#)
- checkBasis
  - OsiDyIpsolverWarmStartBasis, [71](#)
- chgcnt1
  - lpstats\_struct, [43](#)
- chgcnt2
  - lpstats\_struct, [43](#)
- cimstrcmp
  - dylib\_strerrorns.h, [122](#)
- cistrncmp
  - dylib\_strerrorns.h, [122](#)
- clb
  - consys\_struct, [25](#)
- clear\_pool
  - gplib.h, [99](#), [100](#)
- clone
  - OsiDyIpsolverInterface, [57](#)
  - OsiDyIpsolverWarmStartBasis, [71](#)
  - OsiDyIpsolverWarmStartBasisDiff, [72](#)
- clrflg
  - dylib\_std.h, [120](#)
- cmdHALTERROR
  - dy\_cmdint.h, [77](#)
- cmdHALTNOERROR
  - dy\_cmdint.h, [77](#)
- cmdOK
  - dy\_cmdint.h, [77](#)
- cmd\_retval
  - dy\_cmdint.h, [77](#)
- cnidx
  - basisel\_struct, [10](#)
- cnt
  - lpstats\_struct, [41](#)
  - pkvec\_struct, [74](#)
- coeff\_struct
  - dy\_consys.h, [82](#)
- coeff\_struct\_tag, [18](#)
  - colhdr, [19](#)
  - colnxt, [19](#)
  - rowhdr, [19](#)
  - rownxt, [19](#)
  - val, [19](#)
- coeffcnt
  - conmtx\_struct, [21](#)
- coeffs
  - colhdr\_struct\_tag, [20](#)
  - pkvec\_struct, [75](#)
  - rowhdr\_struct\_tag, [76](#)
- coldbasis
  - lpopts\_struct, [34](#)
- coldvars
  - lpopts\_struct, [33](#)
- colhdr
  - coeff\_struct\_tag, [19](#)
- colhdr\_struct
  - dy\_consys.h, [82](#)
- colhdr\_struct\_tag, [19](#)
  - coeffs, [20](#)
  - len, [20](#)
  - ndx, [20](#)
  - nme, [20](#)
- colnxt



- coeff\_struct\_tag, [19](#)
- cols
  - conmtx\_struct, [21](#)
- colscale
  - consys\_struct, [24](#)
- colsize
  - consys\_struct, [23](#)
  - lpprob\_struct, [38](#)
- com
  - bnfdef\_any, [11](#)
  - bnfref\_any, [15](#)
- comflg
  - dylib\_std.h, [120](#)
- compcnt
  - dylib\_bnfdr.h, [108](#)
- comphd
  - dylib\_bnfdr.h, [108](#)
- compressRows
  - OsiDyIpWarmStartBasis, [70](#)
- comps
  - bnfGdef\_struct, [12](#)
- con
  - lpopts\_struct, [33](#)
- conbnd\_struct, [20](#)
  - bnd, [20](#)
  - inf, [20](#)
  - revs, [20](#)
- concnt
  - consys\_struct, [23](#)
- condition
  - OsiDyIpSolverInterface, [67](#)
- config\_default.h
  - DYLP\_ERRMSGDIR, [103](#)
  - DYLP\_ISFINITE, [103](#)
  - DYLP\_ISNAN, [103](#)
  - DYLP\_STATISTICS, [103](#)
- config\_dylp\_default.h
  - BOOL, [104](#)
  - DYLP\_VERSION, [103](#)
- conmgmt
  - lpopts\_struct, [36](#)
- conmtx\_struct, [21](#)
  - coeffcnt, [21](#)
  - cols, [21](#)
  - rows, [21](#)
- conname\_code
  - dylib\_fortran.h, [116](#)
- cons
  - lpopts\_struct, [34](#), [35](#)
  - lpstats\_struct, [41](#)
- consys
  - lpprob\_struct, [37](#)
- consys\_1normcol
  - dy\_consys.h, [84](#)
- consys\_1normrow
  - dy\_consys.h, [84](#)
- consys\_2normcol
  - dy\_consys.h, [84](#)
- consys\_2normrow
  - dy\_consys.h, [84](#)
- consys\_accumcol
  - dy\_consys.h, [84](#)
- consys\_addcol\_ex
  - dy\_consys.h, [83](#)
- consys\_addcol\_pk
  - dy\_consys.h, [83](#)
- consys\_addrow\_pk
  - dy\_consys.h, [83](#)
- consys\_applyscale
  - dy\_consys.h, [84](#)
- consys\_assocnme
  - dy\_consys.h, [84](#)
- consys\_attach
  - dy\_consys.h, [83](#)
- consys\_chgnme
  - dy\_consys.h, [85](#)
- consys\_conbndnme
  - dy\_consys.h, [84](#)
- consys\_conbndval
  - dy\_consys.h, [85](#)
- consys\_create
  - dy\_consys.h, [83](#)
- consys\_delcol
  - dy\_consys.h, [84](#)
- consys\_delrow
  - dy\_consys.h, [84](#)
- consys\_delrow\_stable
  - dy\_consys.h, [84](#)
- consys\_detach
  - dy\_consys.h, [83](#)
- consys\_divrow
  - dy\_consys.h, [84](#)
- consys\_dotcol
  - dy\_consys.h, [84](#)
- consys\_dotrow
  - dy\_consys.h, [84](#)
- consys\_dupsys
  - dy\_consys.h, [83](#)
- consys\_equiscale
  - dy\_consys.h, [84](#)
- consys\_evalsys
  - dy\_consys.h, [84](#)
- consys\_free
  - dy\_consys.h, [83](#)
- consys\_gcdrow
  - dy\_consys.h, [84](#)
- consys\_geomscale
  - dy\_consys.h, [84](#)

- consys\_getcoeff
  - dy\_consys.h, [84](#)
- consys\_getcol\_ex
  - dy\_consys.h, [83](#)
- consys\_getcol\_pk
  - dy\_consys.h, [83](#)
- consys\_getrow\_ex
  - dy\_consys.h, [83](#)
- consys\_getrow\_pk
  - dy\_consys.h, [83](#)
- consys\_infnormcol
  - dy\_consys.h, [84](#)
- consys\_infnormrow
  - dy\_consys.h, [84](#)
- consys\_logicals
  - dy\_consys.h, [84](#)
- consys\_mulaccumcol
  - dy\_consys.h, [84](#)
- consys\_mulrow
  - dy\_consys.h, [84](#)
- consys\_nme
  - dy\_consys.h, [85](#)
- consys\_prtcon
  - dy\_consys.h, [85](#)
- consys\_prtcontyp
  - dy\_consys.h, [84](#)
- consys\_prtvartyp
  - dy\_consys.h, [84](#)
- consys\_realloc
  - dy\_consys.h, [83](#)
- consys\_setcoeff
  - dy\_consys.h, [84](#)
- consys\_ssqcol
  - dy\_consys.h, [84](#)
- consys\_ssqrow
  - dy\_consys.h, [84](#)
- consys\_struct, [21](#)
  - archcnt, [23](#)
  - archvnt, [23](#)
  - attvecs, [25](#)
  - binvnt, [23](#)
  - clb, [25](#)
  - colscale, [24](#)
  - colsize, [23](#)
  - concnt, [23](#)
  - ctyp, [24](#)
  - cub, [25](#)
  - cutccnt, [23](#)
  - inf, [22](#)
  - intvnt, [23](#)
  - logvnt, [23](#)
  - maxaij, [24](#)
  - maxcollen, [23](#)
  - maxcolndx, [23](#)
  - maxrowlen, [23](#)
  - maxrowndx, [23](#)
  - minaij, [24](#)
  - mtx, [23](#)
  - nme, [22](#)
  - obj, [24](#)
  - objndx, [24](#)
  - objnme, [24](#)
  - opts, [22](#)
  - parts, [22](#)
  - rhs, [24](#)
  - rhslow, [24](#)
  - rowscale, [24](#)
  - rowse, [23](#)
  - tiny, [22](#)
  - varcnt, [22](#)
  - vlb, [24](#)
  - vtyp, [24](#)
  - vub, [24](#)
  - xzndx, [24](#)
- consys\_update
  - dy\_consys.h, [83](#)
- context
  - lpopts\_struct, [32](#)
- contypEQ
  - dy\_consys.h, [83](#)
- contypGE
  - dy\_consys.h, [83](#)
- contypINV
  - dy\_consys.h, [83](#)
- contypLE
  - dy\_consys.h, [83](#)
- contypNB
  - dy\_consys.h, [83](#)
- contypRNG
  - dy\_consys.h, [83](#)
- contyp\_enum
  - dy\_consys.h, [82](#)
- copyorigsys
  - lpopts\_struct, [33](#)
- cost
  - lptols\_struct, [44](#)
- count
  - POOL, [75](#)
- crash
  - lpopts\_struct, [35](#)
- create\_pool
  - glib.h, [98](#), [100](#)
- cs\_head
  - LUF\_WA, [50](#)
- cs\_next
  - LUF\_WA, [50](#)
- cs\_prev
  - LUF\_WA, [50](#)

- ctlopts
  - lpprob\_struct, [37](#)
- ctyp
  - consys\_struct, [24](#)
- cub
  - consys\_struct, [25](#)
- cutcnt
  - consys\_struct, [23](#)
- cxBANDC
  - dylp.h, [93](#)
- cxINITIALLP
  - dylp.h, [93](#)
- cxINV
  - dylp.h, [93](#)
- cxLOAD
  - dylp.h, [93](#)
- cxSINGLELP
  - dylp.h, [93](#)
- cxUNLOAD
  - dylp.h, [93](#)
- cxUSERPIV
  - dylp.h, [93](#)
- cxtypes\_enum
  - dylp.h, [93](#)
- D
- d2
  - lpstats\_struct, [43](#)
- d2p
  - lpopts\_struct, [35](#)
- DY\_LCDEL
  - dylib\_io.h, [118](#)
- DY\_LCEOF
  - dylib\_io.h, [118](#)
- DY\_LCERR
  - dylib\_io.h, [118](#)
- DY\_LCFS
  - dylib\_io.h, [118](#)
- DY\_LCID
  - dylib\_io.h, [118](#)
- DY\_LCNIL
  - dylib\_io.h, [118](#)
- DY\_LCNUM
  - dylib\_io.h, [118](#)
- DY\_LCQS
  - dylib\_io.h, [118](#)
- DYLP\_ERRMSGDIR
  - config\_default.h, [103](#)
- DYLP\_INTERNAL
  - dy\_cmdint.h, [77](#)
  - OsiDyLpSolverInterface.hpp, [125](#)
  - OsiDyLpWarmStartBasis.hpp, [126](#)
- DYLP\_ISFINITE
  - config\_default.h, [103](#)
- DYLP\_ISNAN
  - config\_default.h, [103](#)
- DYLP\_STATISTICS
  - config\_default.h, [103](#)
- DYLP\_VERSION
  - config\_dylp\_default.h, [103](#)
- DYSTATS\_HISTBINS
  - dylp.h, [91](#)
- DYSTATS\_MAXDEGEN
  - dylp.h, [91](#)
- dbdef
  - dylib\_bnfrdr.h, [109](#)
- dbref
  - dylib\_bnfrdr.h, [110](#)
- dbrefdbg
  - dylib\_bnfrdr.h, [111](#)
- dchk
  - lptols\_struct, [44](#)
- ddegen
  - lpstats\_struct, [43](#)
- deactcnt
  - lpstats\_struct, [41](#)
- deactivl
  - lpopts\_struct, [33](#)
- degen
  - lpopts\_struct, [33](#), [35](#)
- degenlite
  - lpopts\_struct, [33](#)
- degenpivlim
  - lpopts\_struct, [33](#)
- delete\_pool
  - glplib.h, [99](#), [100](#)
- deleteCols
  - OsiDyLpSolverInterface, [61](#)
- deleteRows
  - OsiDyLpSolverInterface, [61](#)
  - OsiDyLpWarmStartBasis, [70](#)
- dfdef
  - dylib\_bnfrdr.h, [108](#)
- dfeas
  - lptols\_struct, [44](#)
- dfeas\_scale
  - lptols\_struct, [45](#)
- dfgs
  - bnfLBdef\_struct, [13](#)
  - bnfLdef\_struct, [14](#)
- dflt
  - pkvec\_struct, [74](#)
- dfref
  - dylib\_bnfrdr.h, [110](#)
- dfrefdbg
  - dylib\_bnfrdr.h, [110](#)
- dim
  - pkvec\_struct, [74](#)

- disableFactorization
  - OsiDyIpsolverInterface, 64
- dmulti
  - lpstats\_struct, 42
- double\_precision
  - dylb\_fortran.h, 115
- double\_precision\_code
  - dylb\_fortran.h, 115
- dpisel
  - lpopts\_struct, 32
- dual
  - lpopts\_struct, 36
- dualadd
  - lpopts\_struct, 33
- dy\_cmdint.h
  - cmdHALTERROR, 77
  - cmdHALTNOERROR, 77
  - cmdOK, 77
- dy\_consys.h
  - contypEQ, 83
  - contypGE, 83
  - contypINV, 83
  - contypLE, 83
  - contypNB, 83
  - contypRNG, 83
  - vartypBIN, 83
  - vartypCON, 83
  - vartypINT, 83
  - vartypINV, 83
- dyADDCON
  - dylp.h, 92
- dyADDVAR
  - dylp.h, 92
- dyDONE
  - dylp.h, 92
- dyDUAL
  - dylp.h, 92
- dyFORCEDUAL
  - dylp.h, 92
- dyFORCEFULL
  - dylp.h, 92
- dyFORCEPRIMAL
  - dylp.h, 92
- dyGENCON
  - dylp.h, 92
- dyGENVAR
  - dylp.h, 92
- dyINIT
  - dylp.h, 92
- dyINV
  - dylp.h, 92
- dyPRIMAL1
  - dylp.h, 92
- dyPRIMAL2
  - dylp.h, 92
- dyPURGECON
  - dylp.h, 92
- dyPURGEVAR
  - dylp.h, 92
- dy\_abari
  - dylp.h, 94
- dy\_abarj
  - dylp.h, 94
- dy\_betai
  - dylp.h, 94
- dy\_betaj
  - dylp.h, 94
- dy\_betak
  - dylp.h, 94
- dy\_checkdefaults
  - dylp.h, 94
- dy\_cmdint.h
  - cmd\_retval, 77
  - DYLP\_INTERNAL, 77
  - dy\_processcmds, 77
- dy\_colDuals
  - dylp.h, 94
- dy\_colPrimals
  - dylp.h, 94
- dy\_colStatus
  - dylp.h, 94
- dy\_consys.h
  - attvhdr\_struct, 82
  - CONSYS\_CLB, 80
  - CONSYS\_COL, 80
  - CONSYS\_COLHDR, 81
  - CONSYS\_COLVEC, 81
  - CONSYS\_CORRUPT, 82
  - CONSYS\_CSCALE, 81
  - CONSYS\_CTYP, 80
  - CONSYS\_CUB, 80
  - CONSYS\_FININF, 82
  - CONSYS\_LVARS, 82
  - CONSYS\_MAXBUFLEN, 82
  - CONSYS\_MTX, 80
  - CONSYS\_OBJ, 80
  - CONSYS\_RHS, 80
  - CONSYS\_RHSLOW, 80
  - CONSYS\_ROW, 80
  - CONSYS\_ROWHDR, 81
  - CONSYS\_ROWVEC, 81
  - CONSYS\_RSCALE, 81
  - CONSYS\_VLB, 80
  - CONSYS\_VTYP, 80
  - CONSYS\_VUB, 80
  - CONSYS\_WRNATT, 82
  - CONSYS\_WRNZERO, 82
  - coeff\_struct, 82

- colhdr\_struct, [82](#)
- consys\_1normcol, [84](#)
- consys\_1normrow, [84](#)
- consys\_2normcol, [84](#)
- consys\_2normrow, [84](#)
- consys\_accumcol, [84](#)
- consys\_addcol\_ex, [83](#)
- consys\_addcol\_pk, [83](#)
- consys\_addrow\_pk, [83](#)
- consys\_applyscale, [84](#)
- consys\_assocnme, [84](#)
- consys\_attach, [83](#)
- consys\_chgnme, [85](#)
- consys\_conbndnme, [84](#)
- consys\_conbndval, [85](#)
- consys\_create, [83](#)
- consys\_delcol, [84](#)
- consys\_delrow, [84](#)
- consys\_delrow\_stable, [84](#)
- consys\_detach, [83](#)
- consys\_divrow, [84](#)
- consys\_dotcol, [84](#)
- consys\_dotrow, [84](#)
- consys\_dupsys, [83](#)
- consys\_equiscale, [84](#)
- consys\_evalsys, [84](#)
- consys\_free, [83](#)
- consys\_gcdrow, [84](#)
- consys\_geomscale, [84](#)
- consys\_getcoeff, [84](#)
- consys\_getcol\_ex, [83](#)
- consys\_getcol\_pk, [83](#)
- consys\_getrow\_ex, [83](#)
- consys\_getrow\_pk, [83](#)
- consys\_infnormcol, [84](#)
- consys\_infnormrow, [84](#)
- consys\_logicals, [84](#)
- consys\_mulaccumcol, [84](#)
- consys\_mulrow, [84](#)
- consys\_nme, [85](#)
- consys\_prtcon, [85](#)
- consys\_prtcontyp, [84](#)
- consys\_prtvartyp, [84](#)
- consys\_realloc, [83](#)
- consys\_setcoeff, [84](#)
- consys\_ssqcol, [84](#)
- consys\_ssqrow, [84](#)
- consys\_update, [83](#)
- contyp\_enum, [82](#)
- INT\_VARTYPE, [82](#)
- rowhdr\_struct, [82](#)
- VALID\_ATTVMYPE, [81](#)
- VALID\_CONTYPE, [81](#)
- VALID\_VARTYPE, [81](#)
- vartyp\_enum, [83](#)
- dy\_defaults
  - dylp.h, [94](#)
- dy\_dualRays
  - dylp.h, [94](#)
- dy\_dumpcompact
  - dylp.h, [95](#)
- dy\_dumpstats
  - dylp.h, [95](#)
- dy\_dupbasis
  - dylp.h, [94](#)
- dy\_expandxopt
  - dylp.h, [94](#)
- dy\_freesoln
  - dylp.h, [94](#)
- dy\_freestats
  - dylp.h, [95](#)
- dy\_getOwner
  - dylp.h, [94](#)
- dy\_initstats
  - dylp.h, [95](#)
- dy\_logPrimals
  - dylp.h, [94](#)
- dy\_logStatus
  - dylp.h, [94](#)
- dy\_pricedualpiv
  - dylp.h, [94](#)
- dy\_pricenbvars
  - dylp.h, [94](#)
- dy\_primalRays
  - dylp.h, [94](#)
- dy\_processcmds
  - dy\_cmdint.h, [77](#)
- dy\_prtlpphase
  - dylp.h, [94](#)
- dy\_prtlpret
  - dylp.h, [94](#)
- dy\_prtvstat
  - dylp.h, [95](#)
- dy\_rowDuals
  - dylp.h, [94](#)
- dy\_rowDualsGivenC
  - dylp.h, [94](#)
- dy\_rowPrimals
  - dylp.h, [94](#)
- dy\_setgtxecho
  - dylp.h, [95](#)
- dy\_setlogchn
  - dylp.h, [95](#)
- dy\_setprintopts
  - dylp.h, [94](#)
- dy\_vector.h
  - exvec\_1norm, [85](#)
  - exvec\_2norm, [85](#)

- exvec\_infnorm, [86](#)
- exvec\_ssq, [85](#)
- pkvec\_2norm, [85](#)
- pkvec\_check, [85](#)
- pkvec\_dotexvec, [86](#)
- pkvec\_free, [85](#)
- pkvec\_new, [85](#)
- pkvec\_resize, [85](#)
- dyhash\_enter
  - dylib\_hash.h, [116](#)
- dyhash\_erase
  - dylib\_hash.h, [116](#)
- dyhash\_lookup
  - dylib\_hash.h, [116](#)
- dyhash\_search
  - dylib\_hash.h, [116](#)
- dyio\_backup
  - dylib\_io.h, [118](#)
- dyio\_chgerrlog
  - dylib\_io.h, [118](#)
- dyio\_closefile
  - dylib\_io.h, [118](#)
- dyio\_flushio
  - dylib\_io.h, [119](#)
- dyio\_idtopath
  - dylib\_io.h, [118](#)
- dyio\_oinit
  - dylib\_io.h, [118](#)
- dyio\_ioterm
  - dylib\_io.h, [118](#)
- dyio\_isactive
  - dylib\_io.h, [118](#)
- dyio\_mark
  - dylib\_io.h, [118](#)
- dyio\_openfile
  - dylib\_io.h, [118](#)
- dyio\_outchr
  - dylib\_io.h, [119](#)
- dyio\_outfmt
  - dylib\_io.h, [119](#)
- dyio\_outfxd
  - dylib\_io.h, [119](#)
- dyio\_pathoid
  - dylib\_io.h, [118](#)
- dyio\_scan
  - dylib\_io.h, [118](#)
- dyio\_scanlex
  - dylib\_io.h, [118](#)
- dyio\_scanstr
  - dylib\_io.h, [118](#)
- dyio\_setmode
  - dylib\_io.h, [118](#)
- dyio\_ttyq
  - dylib\_io.h, [118](#)

- dylib\_bnfdr.h
  - bnfDL, [112](#)
  - bnfDS, [112](#)
  - bnfG, [112](#)
  - bnfI, [112](#)
  - bnfL, [112](#)
  - bnfNP, [112](#)
  - bnfP, [112](#)
  - bnfRL, [112](#)
  - bnfRS, [112](#)
  - bnfT, [112](#)
  - bnfncBNF, [113](#)
  - bnfncC, [113](#)
  - bnfncN, [113](#)
  - bnfncS, [113](#)
  - bnfttD, [112](#)
  - bnfttF, [112](#)
  - bnfttID, [112](#)
  - bnfttN, [112](#)
  - bnfttNIL, [112](#)
  - bnfttQ, [112](#)
- dylib\_io.h
  - DY\_LCDEL, [118](#)
  - DY\_LCEOF, [118](#)
  - DY\_LCERR, [118](#)
  - DY\_LCFS, [118](#)
  - DY\_LCID, [118](#)
  - DY\_LCNIL, [118](#)
  - DY\_LCNUM, [118](#)
  - DY\_LCQS, [118](#)
- dylib\_bnfdr.h
  - addrToInt, [107](#)
  - altcnt, [108](#)
  - althd, [107](#)
  - bnfGref\_struct, [112](#)
  - bnfIref\_struct, [112](#)
  - bnfLBref\_struct, [111](#)
  - bnfLref\_struct, [111](#)
  - bnfNPref\_struct, [112](#)
  - bnfPdef\_struct, [111](#)
  - bnfPref\_struct, [112](#)
  - bnfTref\_struct, [112](#)
  - bnfadv, [106](#)
  - bnfatsgn, [106](#)
  - bnfcs, [106](#)
  - bnfdbgctl, [113](#)
  - bnfdbl, [107](#)
  - bnfdebug, [107](#)
  - bnfdef\_common, [107](#)
  - bnfexact, [107](#)
  - bnfflt, [106](#)
  - bnfblsrc\_enum, [112](#)
  - bnflst, [106](#)
  - bnfmin, [107](#)

- bnfref\_common, 107
- bnfref\_struct, 111
- bnfstbg, 106
- bnfstore, 106
- bnfsv, 107
- bnfsvnd, 106
- bnfsvnm, 106
- bnftype\_enum, 112
- bnftype\_enum, 112
- compcnt, 108
- comphd, 108
- dbdef, 109
- dbref, 110
- dbrefdbg, 111
- dfdef, 108
- dfref, 110
- dfrefdbg, 110
- gdef, 108
- gref, 109
- idef, 109
- iref, 110
- ldef, 109
- lref, 110
- lrefdbg, 111
- mkaref, 108
- mkcref, 108
- mkoff, 107
- mksav, 107
- NULLP, 107
- npdef, 108
- npref, 109
- parse, 113
- pdef, 108
- pref, 110
- prtbnfdef, 113
- prtbnfref, 113
- rbdef, 109
- rbref, 110
- rbrefdbg, 111
- rdrclear, 113
- rdrinit, 113
- rfdef, 109
- rfref, 110
- rfrefdbg, 111
- tdef, 108
- tqdef, 108
- tref, 110
- dylib\_errs.h
  - errinit, 113
  - errmsg, 113
  - errterm, 113
  - warn, 113
- dylib\_fortran.h
  - argcod\_, 116
- character\_code, 115
- conname\_code, 116
- double\_precision, 115
- double\_precision\_code, 115
- end\_code, 116
- f\_arr1, 114
- f\_arr2, 114
- f\_chr, 114
- FALSEL, 114
- ftnargCHARACTER, 115
- ftnargCONNAME, 115
- ftnargDOUBLE\_PRECISION, 115
- ftnargEND, 115
- ftnargINTEGER, 114
- ftnargVARNAME, 115
- integer, 115
- integer\_2, 115
- integer\_code, 115
- logical, 115
- real, 115
- TRUEL, 114
- varname\_code, 116
- dylib\_hash.h
  - dyhash\_enter, 116
  - dyhash\_erase, 116
  - dyhash\_lookup, 116
  - dyhash\_search, 116
  - hel, 116
- dylib\_io.h
  - dyio\_backup, 118
  - dyio\_chgerrlog, 118
  - dyio\_closefile, 118
  - dyio\_flushio, 119
  - dyio\_idtopath, 118
  - dyio\_ioint, 118
  - dyio\_ioterm, 118
  - dyio\_isactive, 118
  - dyio\_mark, 118
  - dyio\_openfile, 118
  - dyio\_outchr, 119
  - dyio\_outfmt, 119
  - dyio\_outfxd, 119
  - dyio\_pathtoid, 118
  - dyio\_scan, 118
  - dyio\_scanlex, 118
  - dyio\_scanstr, 118
  - dyio\_setmode, 118
  - dyio\_ttyq, 118
  - IOID\_INV, 117
  - IOID\_NOSTRM, 117
  - ioid, 118
  - lexclass, 118
- dylib\_keytab.h
  - ambig, 119

- find, [119](#)
- keytab\_entry, [119](#)
- dylib\_std.h
  - bool, [121](#)
  - CALLOC, [121](#)
  - clrflg, [120](#)
  - comflg, [120](#)
  - FALSE, [120](#)
  - FREE, [121](#)
  - flags, [121](#)
  - flgall, [121](#)
  - flgoff, [121](#)
  - flgon, [120](#)
  - getflg, [120](#)
  - Ink\_in, [121](#)
  - Ink\_out, [121](#)
  - Ink\_struct, [121](#)
  - MALLOC, [121](#)
  - MALLOC\_DBG\_INIT, [121](#)
  - maxx, [121](#)
  - minn, [121](#)
  - REALLOC, [121](#)
  - setflg, [120](#)
  - TRUE, [120](#)
  - UNUSED, [120](#)
- dylib\_strtrns.h
  - cimstrcmp, [122](#)
  - cistrncmp, [122](#)
  - mstrncmp, [122](#)
  - STRALLOC, [122](#)
  - STRFREE, [122](#)
  - stralloc, [122](#)
  - strfree, [122](#)
  - strsave, [122](#)
- dylp
  - dylp.h, [94](#)
- dylp.h
  - cxBANDC, [93](#)
  - cxINITIALLP, [93](#)
  - cxINV, [93](#)
  - cxLOAD, [93](#)
  - cxSINGLELP, [93](#)
  - cxUNLOAD, [93](#)
  - cxUSERPIV, [93](#)
  - dyADDCON, [92](#)
  - dyADDVAR, [92](#)
  - dyDONE, [92](#)
  - dyDUAL, [92](#)
  - dyFORCEDUAL, [92](#)
  - dyFORCEFULL, [92](#)
  - dyFORCEPRIMAL, [92](#)
  - dyGENCON, [92](#)
  - dyGENVAR, [92](#)
  - dyINIT, [92](#)
  - dyINV, [92](#)
  - dyPRIMAL1, [92](#)
  - dyPRIMAL2, [92](#)
  - dyPURGECON, [92](#)
  - dyPURGEVAR, [92](#)
  - dyrACCCHK, [93](#)
  - dyrBSPACE, [93](#)
  - dyrDEGEN, [93](#)
  - dyrFATAL, [93](#)
  - dyrINFEAS, [93](#)
  - dyrINV, [93](#)
  - dyrITERLIM, [93](#)
  - dyrLOSTDFEAS, [93](#)
  - dyrLOSTPFEAS, [93](#)
  - dyrMADPIV, [93](#)
  - dyrNUMERIC, [93](#)
  - dyrOK, [93](#)
  - dyrOPTIMAL, [93](#)
  - dyrPATCHED, [93](#)
  - dyrPUNT, [93](#)
  - dyrREQCHK, [93](#)
  - dyrRESELECT, [93](#)
  - dyrSINGULAR, [93](#)
  - dyrSTALLED, [93](#)
  - dyrSWING, [93](#)
  - dyrUNBOUND, [93](#)
  - ibARCH, [93](#)
  - ibINV, [93](#)
  - ibLOGICAL, [93](#)
  - ibSLACK, [93](#)
  - lpACCCHK, [92](#)
  - lpFATAL, [92](#)
  - lpFORCEDUAL, [92](#)
  - lpFORCEFULL, [92](#)
  - lpFORCEPRIMAL, [92](#)
  - lpINFEAS, [92](#)
  - lpINV, [92](#)
  - lpITERLIM, [92](#)
  - lpLOSTFEAS, [92](#)
  - lpNOSPACE, [92](#)
  - lpOPTIMAL, [92](#)
  - lpPUNT, [92](#)
  - lpSTALLED, [92](#)
  - lpSWING, [92](#)
  - lpUNBOUNDED, [92](#)
- dylp.h
  - cxtype\_enum, [93](#)
  - DYSTATS\_HISTBINS, [91](#)
  - DYSTATS\_MAXDEGEN, [91](#)
  - dy\_abari, [94](#)
  - dy\_abarj, [94](#)
  - dy\_betai, [94](#)
  - dy\_betaj, [94](#)
  - dy\_betak, [94](#)



dy\_checkdefaults, 94  
 dy\_colDuals, 94  
 dy\_colPrimals, 94  
 dy\_colStatus, 94  
 dy\_defaults, 94  
 dy\_dualRays, 94  
 dy\_dumpcompact, 95  
 dy\_dumpstats, 95  
 dy\_dupbasis, 94  
 dy\_expandxopt, 94  
 dy\_freesoln, 94  
 dy\_freestats, 95  
 dy\_getOwner, 94  
 dy\_initstats, 95  
 dy\_logPrimals, 94  
 dy\_logStatus, 94  
 dy\_pricedualpiv, 94  
 dy\_pricenbvars, 94  
 dy\_primalRays, 94  
 dy\_prtlpphase, 94  
 dy\_prtlpret, 94  
 dy\_prtvstat, 95  
 dy\_rowDuals, 94  
 dy\_rowDualsGivenC, 94  
 dy\_rowPrimals, 94  
 dy\_setgtxecho, 95  
 dy\_setlogchn, 95  
 dy\_setprintopts, 94  
 dylp, 94  
 dyphase\_enum, 92  
 dyret\_enum, 92  
 ibtype\_enum, 93  
 ladDFQUIET, 88  
 ladDUALCHK, 88  
 ladDUALFEAS, 88  
 ladDUALS, 89  
 ladEXPAND, 89  
 ladFACTOR, 89  
 ladPFQUIET, 88  
 ladPRIMALCHK, 88  
 ladPRIMALS, 89  
 ladPRIMFEAS, 88  
 lpctlACTVARSIN, 91  
 lpctlACTVARSOUT, 91  
 lpctlIDYVALID, 91  
 lpctlINITACTCON, 91  
 lpctlINITACTVAR, 91  
 lpctlLBNDCHG, 91  
 lpctlNOFREE, 90  
 lpctlOBJCHG, 91  
 lpctlONLYFREE, 91  
 lpctlRHSCHG, 91  
 lpctlUBNDCHG, 91  
 lpret\_enum, 91  
 VALID\_STATUS, 90  
 vstatB, 89  
 vstatBASIC, 90  
 vstatBFR, 89  
 vstatBFX, 89  
 vstatBLB, 89  
 vstatBLLB, 90  
 vstatBUB, 89  
 vstatBUUB, 90  
 vstatEXOTIC, 90  
 vstatINV, 89  
 vstatNBFR, 89  
 vstatNBFX, 89  
 vstatNBLB, 89  
 vstatNBUB, 89  
 vstatNOLOAD, 90  
 vstatNONBASIC, 90  
 vstatNOPER, 90  
 vstatNOPIVOT, 90  
 vstatQUALS, 90  
 vstatSB, 90  
 vstatSTATUS, 90  
 dylp\_controlfile  
     OsiDylpSolverInterface, 66  
 dylp\_logfile  
     OsiDylpSolverInterface, 66  
 dylp\_outfile  
     OsiDylpSolverInterface, 66  
 dylp\_printsoln  
     OsiDylpSolverInterface, 66  
 dyphase\_enum  
     dylp.h, 92  
 dyrACCHK  
     dylp.h, 93  
 dyrBSPACE  
     dylp.h, 93  
 dyrDEGEN  
     dylp.h, 93  
 dyrFATAL  
     dylp.h, 93  
 dyrINFEAS  
     dylp.h, 93  
 dyrINV  
     dylp.h, 93  
 dyrITERLIM  
     dylp.h, 93  
 dyrLOSTDFEAS  
     dylp.h, 93  
 dyrLOSTPFEAS  
     dylp.h, 93  
 dyrMADPIV  
     dylp.h, 93  
 dyrNUMERIC  
     dylp.h, 93

- dyrOK
  - dylp.h, [93](#)
- dyrOPTIMAL
  - dylp.h, [93](#)
- dyrPATCHED
  - dylp.h, [93](#)
- dyrPUNT
  - dylp.h, [93](#)
- dyrREQCHK
  - dylp.h, [93](#)
- dyrRESELECT
  - dylp.h, [93](#)
- dyrSINGULAR
  - dylp.h, [93](#)
- dyrSTALLED
  - dylp.h, [93](#)
- dyrSWING
  - dylp.h, [93](#)
- dyrUNBOUND
  - dylp.h, [93](#)
- dyret\_enum
  - dylp.h, [92](#)
- E
- ENV, [25](#)
  - glplib.h, [99](#)
  - mem\_count, [25](#)
  - mem\_cpeak, [26](#)
  - mem\_limit, [25](#)
  - mem\_ptr, [25](#)
  - mem\_total, [25](#)
  - mem\_tpeak, [25](#)
- el
  - basis\_struct, [10](#)
- else
  - attvhdr\_struct\_tag, [9](#)
- enableFactorization
  - OsiDyIpSolverInterface, [64](#)
- end\_code
  - dylib\_fortran.h, [116](#)
- ent
  - hel\_tag, [26](#)
- eps\_tol
  - LUF, [49](#)
- errinit
  - dylib\_errs.h, [113](#)
- errmsg
  - dylib\_errs.h, [113](#)
- errterm
  - dylib\_errs.h, [113](#)
- evals
  - lpstats\_struct, [42](#)
- exvec\_1norm
  - dy\_vector.h, [85](#)
- exvec\_2norm
  - dy\_vector.h, [85](#)
- exvec\_infnorm
  - dy\_vector.h, [86](#)
- exvec\_ssqr
  - dy\_vector.h, [85](#)
- F
- f\_arr1
  - dylib\_fortran.h, [114](#)
- f\_arr2
  - dylib\_fortran.h, [114](#)
- f\_chr
  - dylib\_fortran.h, [114](#)
- FALSE
  - dylib\_std.h, [120](#)
- FALSEL
  - dylib\_fortran.h, [114](#)
- FREE
  - dylib\_std.h, [121](#)
- factor
  - lpopts\_struct, [32](#)
  - lpstats\_struct, [41](#)
- fault
  - glplib.h, [98](#), [100](#)
- fc\_len
  - LUF, [47](#)
- fc\_ptr
  - LUF, [47](#)
- fin
  - lpstats\_struct, [41](#)
- find
  - dylib\_keytab.h, [119](#)
- finpurge
  - lpopts\_struct, [35](#)
- flag
  - LUF, [48](#)
  - MEM, [51](#)
- flags
  - dylib\_std.h, [121](#)
- flex
  - lpopts\_struct, [32](#)
- flgall
  - dylib\_std.h, [121](#)
- flgoff
  - dylib\_std.h, [121](#)
- flgon
  - dylib\_std.h, [120](#)
- flippable
  - lpstats\_struct, [42](#)
- flips
  - lpopts\_struct, [35](#)
  - lpstats\_struct, [42](#)
- force

- lpopts\_struct, 36
- forcecold
  - lpopts\_struct, 33
- forcewarm
  - lpopts\_struct, 33
- fr\_len
  - LUF, 46
- fr\_ptr
  - LUF, 46
- frac
  - lpopts\_struct, 34
- free\_atom
  - glplib.h, 99, 100
- free\_lib\_env
  - glplib.h, 98, 100
- ftnargCHARACTER
  - dylib\_fortran.h, 115
- ftnargCONNAME
  - dylib\_fortran.h, 115
- ftnargDOUBLE\_PRECISION
  - dylib\_fortran.h, 115
- ftnargEND
  - dylib\_fortran.h, 115
- ftnargINTEGER
  - dylib\_fortran.h, 114
- ftnargVARNAME
  - dylib\_fortran.h, 115
- fullsys
  - lpopts\_struct, 33
  - lpprob\_struct, 37
- G
- G
  - bnfdef\_any, 11
  - bnfref\_any, 16
- g
  - parse\_any, 73
- gdef
  - dylib\_bnfdr.h, 108
- generateDiff
  - OsiDyIpWarmStartBasis, 70
- get\_atom
  - glplib.h, 99, 100
- get\_atomv
  - glplib.h, 99, 100
- get\_env\_ptr
  - glplib.h, 98, 100
- getBInvACol
  - OsiDyIpSolverInterface, 65
- getBInvARow
  - OsiDyIpSolverInterface, 65
- getBInvCol
  - OsiDyIpSolverInterface, 65
- getBInvRow
  - OsiDyIpSolverInterface, 65
- getBasics
  - OsiDyIpSolverInterface, 65
- getBasisStatus
  - OsiDyIpSolverInterface, 65
- getColLower
  - OsiDyIpSolverInterface, 59
- getColSolution
  - OsiDyIpSolverInterface, 64
- getColUpper
  - OsiDyIpSolverInterface, 59
- getConStatus
  - OsiDyIpWarmStartBasis, 69
- getConstraintStatus
  - OsiDyIpWarmStartBasis, 69
- getDbParam
  - OsiDyIpSolverInterface, 63
- getDualRays
  - OsiDyIpSolverInterface, 64
- getEmptyWarmStart
  - OsiDyIpSolverInterface, 62
- getHintParam
  - OsiDyIpSolverInterface, 63
- getInfinity
  - OsiDyIpSolverInterface, 63
- getIntParam
  - OsiDyIpSolverInterface, 63
- getIterationCount
  - OsiDyIpSolverInterface, 63
- getMatrixByCol
  - OsiDyIpSolverInterface, 60
- getMatrixByRow
  - OsiDyIpSolverInterface, 60
- getNumCols
  - OsiDyIpSolverInterface, 59
- getNumElements
  - OsiDyIpSolverInterface, 59
- getNumIntegers
  - OsiDyIpSolverInterface, 59
- getNumRows
  - OsiDyIpSolverInterface, 59
- getObjCoefficients
  - OsiDyIpSolverInterface, 60
- getObjSense
  - OsiDyIpSolverInterface, 60
- getObjValue
  - OsiDyIpSolverInterface, 64
- getPhase
  - OsiDyIpWarmStartBasis, 70
- getPrimalRays
  - OsiDyIpSolverInterface, 64
- getReducedCost
  - OsiDyIpSolverInterface, 64
- getReducedGradient
  - OsiDyIpSolverInterface, 65

- OsiDyLpSolverInterface, 65
- getRightHandSide
  - OsiDyLpSolverInterface, 59
- getRowActivity
  - OsiDyLpSolverInterface, 64
- getRowLower
  - OsiDyLpSolverInterface, 59
- getRowPrice
  - OsiDyLpSolverInterface, 64
- getRowRange
  - OsiDyLpSolverInterface, 59
- getRowSense
  - OsiDyLpSolverInterface, 59
- getRowUpper
  - OsiDyLpSolverInterface, 60
- getStrParam
  - OsiDyLpSolverInterface, 63
- getWarmStart
  - OsiDyLpSolverInterface, 62
- getflg
  - dylib\_std.h, 120
- glpinv.h
  - INV, 96
  - inv\_btran, 96
  - inv\_create, 96
  - inv\_decomp, 96
  - inv\_delete, 96
  - inv\_ftran, 96
  - inv\_h\_solve, 96
  - inv\_update, 96
- glplib.h
  - \_insist, 98, 100
  - align\_boundary, 99
  - align\_datasize, 99
  - clear\_pool, 99, 100
  - create\_pool, 98, 100
  - delete\_pool, 99, 100
  - ENV, 99
  - fault, 98, 100
  - free\_atom, 99, 100
  - free\_lib\_env, 98, 100
  - get\_atom, 99, 100
  - get\_atomv, 99, 100
  - get\_env\_ptr, 98, 100
  - init\_lib\_env, 98, 99
  - insist, 99
  - MEM, 99
  - POOL, 99
  - print, 98, 100
  - read\_pointer, 98, 99
  - save\_pointer, 98, 99
  - ucalloc, 98, 100
  - ufree, 98, 100
  - umalloc, 98, 100
  - watch, 98, 100
- glpluf.h
  - LUF, 102
  - LUF\_WA, 102
  - luf\_alloc\_wa, 101, 102
  - luf\_create, 101, 102
  - luf\_decomp, 101, 102
  - luf\_defrag\_sva, 101, 102
  - luf\_delete, 102
  - luf\_enlarge\_col, 101, 102
  - luf\_enlarge\_row, 101, 102
  - luf\_f\_solve, 101, 102
  - luf\_free\_wa, 101, 102
  - luf\_solve, 102
  - luf\_v\_solve, 101, 102
- gref
  - dylib\_bnfrdr.h, 109
- groom
  - lpopts\_struct, 32
- H
- hel
  - dylib\_hash.h, 116
- hel\_tag, 26
  - ent, 26
  - key, 26
  - next, 26
- heroics
  - lpopts\_struct, 35
- hh\_len
  - INV, 27
- hh\_max
  - INV, 27
- hh\_ndx
  - INV, 27
- hh\_nfs
  - INV, 27
- hh\_ptr
  - INV, 27
- hist
  - lpstats\_struct, 41
- I
- I
  - bnfdef\_any, 11
  - bnfref\_any, 16
- i1I
  - lpopts\_struct, 34
- i1Iopen
  - lpopts\_struct, 34
- i1u
  - lpopts\_struct, 34
- i1uopen
  - lpopts\_struct, 34
- i2I

- lpopts\_struct, 34
- i2lopen
  - lpopts\_struct, 34
- i2u
  - lpopts\_struct, 34
- i2uopen
  - lpopts\_struct, 34
- i2valid
  - lpopts\_struct, 34
- INT\_VARTYPE
  - dy\_consys.h, 82
- INV, 26
  - cc\_len, 28
  - cc\_ndx, 28
  - cc\_val, 28
  - glpinv.h, 96
  - hh\_len, 27
  - hh\_max, 27
  - hh\_ndx, 27
  - hh\_nfs, 27
  - hh\_ptr, 27
  - luf, 27
  - m, 27
  - min\_vrratio, 28
  - nnz\_h, 28
  - p0\_col, 28
  - p0\_row, 27
  - upd\_tol, 28
  - valid, 27
- IOID\_INV
  - dylib\_io.h, 117
- IOID\_NOSTRM
  - dylib\_io.h, 117
- ibARCH
  - dylp.h, 93
- ibINV
  - dylp.h, 93
- ibLOGICAL
  - dylp.h, 93
- ibSLACK
  - dylp.h, 93
- ibtype\_enum
  - dylp.h, 93
- idef
  - dylib\_bnrdr.h, 109
- idlelim
  - lpopts\_struct, 32
- inf
  - conbnd\_struct, 20
  - consys\_struct, 22
  - lptols\_struct, 44
- infeas
  - lpstats\_struct, 43
- ini\_simplex
  - lpstats\_struct, 40
- init
  - lpstats\_struct, 41
- init\_lib\_env
  - glplib.h, 98, 99
- initcons
  - lpopts\_struct, 34
- initialSolve
  - OsiDylpSolverInterface, 62
- initialSolveOptions
  - OsiDylpSolverInterface, 66
- insist
  - glplib.h, 99
- integer
  - dylib\_fortran.h, 115
- integer\_2
  - dylib\_fortran.h, 115
- integer\_code
  - dylib\_fortran.h, 115
- intvcnt
  - consys\_struct, 23
- inv\_btran
  - glpinv.h, 96
- inv\_create
  - glpinv.h, 96
- inv\_decomp
  - glpinv.h, 96
- inv\_delete
  - glpinv.h, 96
- inv\_ftran
  - glpinv.h, 96
- inv\_h\_solve
  - glpinv.h, 96
- inv\_update
  - glpinv.h, 96
- ioid
  - dylib\_io.h, 118
- iref
  - dylib\_bnrdr.h, 110
- isAbandoned
  - OsiDylpSolverInterface, 62
- isBinary
  - OsiDylpSolverInterface, 59
- isContinuous
  - OsiDylpSolverInterface, 59
- isDualObjectiveLimitReached
  - OsiDylpSolverInterface, 63
- isInteger
  - OsiDylpSolverInterface, 59
- isIntegerNonBinary
  - OsiDylpSolverInterface, 59
- isIterationLimitReached
  - OsiDylpSolverInterface, 62
- isPrimalObjectiveLimitReached

- OsiDyIpSolverInterface, 63
- isProvenDualInfeasible
  - OsiDyIpSolverInterface, 62
- isProvenOptimal
  - OsiDyIpSolverInterface, 62
- isProvenPrimalInfeasible
  - OsiDyIpSolverInterface, 62
- iterlim
  - lpopts\_struct, 32
- iters
  - lpprob\_struct, 37
  - lpstats\_struct, 43
- ival
  - bnfdef\_struct, 13
- K
- key
  - hel\_tag, 26
- keytab\_entry
  - dylib\_keytab.h, 119
- keytab\_entry\_internal, 28
  - keyword, 28
  - min, 29
  - token, 29
- keyword
  - keytab\_entry\_internal, 28
- L
- L
  - bnfdef\_any, 11
  - bnfref\_any, 16
- LB
  - bnfdef\_any, 11
  - bnfref\_any, 16
- LUF, 45
  - big\_v, 49
  - eps\_tol, 49
  - fc\_len, 47
  - fc\_ptr, 47
  - flag, 48
  - fr\_len, 46
  - fr\_ptr, 46
  - glpluf.h, 102
  - max\_a, 49
  - max\_gro, 49
  - n, 46
  - new\_sva, 48
  - nnz\_a, 49
  - nnz\_f, 49
  - nnz\_v, 49
  - piv\_lim, 48
  - piv\_tol, 48
  - pp\_col, 47
  - pp\_row, 47
  - qq\_col, 47
  - qq\_row, 47
  - rank, 49
  - suhl, 48
  - sv\_beg, 48
  - sv\_end, 48
  - sv\_head, 48
  - sv\_ndx, 48
  - sv\_next, 48
  - sv\_prev, 48
  - sv\_size, 47
  - sv\_tail, 48
  - sv\_val, 48
  - valid, 46
  - vc\_cap, 47
  - vc\_len, 47
  - vc\_ptr, 47
  - vr\_cap, 47
  - vr\_len, 47
  - vr\_piv, 47
  - vr\_ptr, 47
  - work, 48
- LUF\_WA, 49
  - cs\_head, 50
  - cs\_next, 50
  - cs\_prev, 50
  - glpluf.h, 102
  - rs\_head, 50
  - rs\_max, 50
  - rs\_next, 50
  - rs\_prev, 50
- ladDFQUIET
  - dylp.h, 88
- ladDUALCHK
  - dylp.h, 88
- ladDUALFEAS
  - dylp.h, 88
- ladDUALS
  - dylp.h, 89
- ladEXPAND
  - dylp.h, 89
- ladFACTOR
  - dylp.h, 89
- ladPFQUIET
  - dylp.h, 88
- ladPRIMALCHK
  - dylp.h, 88
- ladPRIMALS
  - dylp.h, 89
- ladPRIMFEAS
  - dylp.h, 88
- ldef
  - dylib\_bnfdr.h, 109
- len
  - basis\_struct, 10

- colhdr\_struct\_tag, [20](#)
- rowhdr\_struct\_tag, [76](#)
- lex\_struct, [29](#)
  - string, [29](#)
- lexclass
  - dylib\_io.h, [118](#)
- link
  - bnfGdef\_struct, [12](#)
  - POOL, [75](#)
- llnxt
  - lnk\_struct\_tag, [30](#)
- llval
  - lnk\_struct\_tag, [30](#)
- lnk\_in
  - dylib\_std.h, [121](#)
- lnk\_out
  - dylib\_std.h, [121](#)
- lnk\_struct
  - dylib\_std.h, [121](#)
- lnk\_struct\_tag, [29](#)
  - llnxt, [30](#)
  - llval, [30](#)
- loadProblem
  - OsiDyLpSolverInterface, [58](#)
- logical
  - dylib\_fortran.h, [115](#)
- logvcnt
  - consys\_struct, [23](#)
- lpACCCHK
  - dylp.h, [92](#)
- lpFATAL
  - dylp.h, [92](#)
- lpFORCEDUAL
  - dylp.h, [92](#)
- lpFORCEFULL
  - dylp.h, [92](#)
- lpFORCEPRIMAL
  - dylp.h, [92](#)
- lpINFEAS
  - dylp.h, [92](#)
- lpINV
  - dylp.h, [92](#)
- lpITERLIM
  - dylp.h, [92](#)
- lpLOSTFEAS
  - dylp.h, [92](#)
- lpNOSPACE
  - dylp.h, [92](#)
- lpOPTIMAL
  - dylp.h, [92](#)
- lpPUNT
  - dylp.h, [92](#)
- lpSTALLED
  - dylp.h, [92](#)
- lpSWING
  - dylp.h, [92](#)
- lpUNBOUNDED
  - dylp.h, [92](#)
- lpctlACTVARSIN
  - dylp.h, [91](#)
- lpctlACTVARSOOT
  - dylp.h, [91](#)
- lpctlDYVALID
  - dylp.h, [91](#)
- lpctlINITACTCON
  - dylp.h, [91](#)
- lpctlINITACTVAR
  - dylp.h, [91](#)
- lpctlLBNDCHG
  - dylp.h, [91](#)
- lpctlNOFREE
  - dylp.h, [90](#)
- lpctlOBJCHG
  - dylp.h, [91](#)
- lpctlONLYFREE
  - dylp.h, [91](#)
- lpctlRHSCHG
  - dylp.h, [91](#)
- lpctlUBNDCHG
  - dylp.h, [91](#)
- lpopts\_struct, [30](#)
  - active, [34](#)
  - actlim, [32](#)
  - actlvl, [32](#)
  - addvar, [33](#)
  - allownopiv, [32](#)
  - basis, [36](#)
  - check, [32](#)
  - coldbasis, [34](#)
  - coldvars, [33](#)
  - con, [33](#)
  - conmgmt, [36](#)
  - cons, [34](#), [35](#)
  - context, [32](#)
  - copyorigsys, [33](#)
  - crash, [35](#)
  - d2p, [35](#)
  - deactlvl, [33](#)
  - degen, [33](#), [35](#)
  - degenlite, [33](#)
  - degenpivlim, [33](#)
  - dpsel, [32](#)
  - dual, [36](#)
  - dualadd, [33](#)
  - factor, [32](#)
  - finpurge, [35](#)
  - flex, [32](#)
  - flips, [35](#)

- force, [36](#)
- forcecold, [33](#)
- forcewarm, [33](#)
- frac, [34](#)
- fullsys, [33](#)
- groom, [32](#)
- heroics, [35](#)
- i1l, [34](#)
- i1lopen, [34](#)
- i1u, [34](#)
- i1uopen, [34](#)
- i2l, [34](#)
- i2lopen, [34](#)
- i2u, [34](#)
- i2uopen, [34](#)
- i2valid, [34](#)
- idlelim, [32](#)
- initcons, [34](#)
- iterlim, [32](#)
- major, [35](#)
- p2d, [35](#)
- patch, [33](#)
- phase1, [35](#)
- phase2, [36](#)
- pivoting, [35](#)
- pivreject, [35](#)
- ppsel, [32](#)
- pricing, [35](#)
- print, [36](#)
- rays, [36](#)
- scaling, [33](#)
- scan, [32](#)
- setup, [35](#)
- soln, [36](#)
- strat, [32](#)
- tableau, [36](#)
- usedual, [33](#)
- varmgmt, [36](#)
- vars, [34](#), [35](#)
- lpprob\_struct, [36](#)
  - actvars, [38](#)
  - basis, [37](#)
  - colsize, [38](#)
  - consys, [37](#)
  - ctlopts, [37](#)
  - fullsys, [37](#)
  - iters, [37](#)
  - lpret, [37](#)
  - obj, [37](#)
  - owner, [37](#)
  - phase, [37](#)
  - rowsize, [38](#)
  - status, [37](#)
  - x, [38](#)
  - y, [38](#)
- lpret
  - lpprob\_struct, [37](#)
- lpret\_enum
  - dylp.h, [91](#)
- lpstats\_struct, [38](#)
  - actcnt, [40](#)
  - angle, [40](#), [41](#)
  - avgpivs, [41](#)
  - avgsiz, [43](#)
  - cands, [42](#)
  - chgcnt1, [43](#)
  - chgcnt2, [43](#)
  - cnt, [41](#)
  - cons, [41](#)
  - d2, [43](#)
  - ddegen, [43](#)
  - deactcnt, [41](#)
  - dmulti, [42](#)
  - evals, [42](#)
  - factor, [41](#)
  - fin, [41](#)
  - flippable, [42](#)
  - flips, [42](#)
  - hist, [41](#)
  - infeas, [43](#)
  - ini\_simplex, [40](#)
  - init, [41](#)
  - iters, [43](#)
  - mad, [41](#)
  - max, [41](#)
  - maxcnt, [43](#)
  - maxpivs, [41](#)
  - maxrnk, [42](#)
  - maxsiz, [43](#)
  - min, [41](#)
  - min\_pivtol, [42](#)
  - nontrivial, [42](#)
  - p1, [43](#)
  - p2, [43](#)
  - pdegen, [43](#)
  - phasecnts, [40](#)
  - pivrej, [42](#)
  - pivrnks, [42](#)
  - pivs, [43](#)
  - pivtol\_red, [42](#)
  - pmulti, [43](#)
  - prevpiv, [41](#)
  - promote, [42](#)
  - puncall, [42](#)
  - puntret, [42](#)
  - sing, [42](#)
  - sz, [40](#)
  - tot, [43](#)



- totpivs, [43](#)
- vars, [41](#)
- lptols\_struct, [43](#)
  - bogus, [45](#)
  - cost, [44](#)
  - dchk, [44](#)
  - dfeas, [44](#)
  - dfeas\_scale, [45](#)
  - inf, [44](#)
  - pchk, [44](#)
  - pfeas, [44](#)
  - pfeas\_scale, [44](#)
  - pivot, [45](#)
  - purge, [45](#)
  - purgevar, [45](#)
  - reframe, [45](#)
  - swing, [45](#)
  - toobig, [45](#)
  - zero, [44](#)
- lref
  - dylib\_bnrdr.h, [110](#)
- lrefdbg
  - dylib\_bnrdr.h, [111](#)
- luf
  - INV, [27](#)
- luf\_alloc\_wa
  - glpluf.h, [101](#), [102](#)
- luf\_create
  - glpluf.h, [101](#), [102](#)
- luf\_decomp
  - glpluf.h, [101](#), [102](#)
- luf\_defrag\_sva
  - glpluf.h, [101](#), [102](#)
- luf\_delete
  - glpluf.h, [102](#)
- luf\_enlarge\_col
  - glpluf.h, [101](#), [102](#)
- luf\_enlarge\_row
  - glpluf.h, [101](#), [102](#)
- luf\_f\_solve
  - glpluf.h, [101](#), [102](#)
- luf\_free\_wa
  - glpluf.h, [101](#), [102](#)
- luf\_solve
  - glpluf.h, [102](#)
- luf\_v\_solve
  - glpluf.h, [101](#), [102](#)
- M
- m
  - INV, [27](#)
- MALLOC
  - dylib\_std.h, [121](#)
- MALLOC\_DBG\_INIT
  - dylib\_std.h, [121](#)
- MEM, [50](#)
  - flag, [51](#)
  - glplib.h, [99](#)
  - next, [51](#)
  - prev, [51](#)
  - size, [51](#)
- mad
  - lpstats\_struct, [41](#)
- major
  - lpopts\_struct, [35](#)
- markHotStart
  - OsiDyIpSolverInterface, [62](#)
- max
  - lpstats\_struct, [41](#)
- max\_a
  - LUF, [49](#)
- max\_gro
  - LUF, [49](#)
- maxaij
  - consys\_struct, [24](#)
- maxcnt
  - lpstats\_struct, [43](#)
- maxcollen
  - consys\_struct, [23](#)
- maxcolndx
  - consys\_struct, [23](#)
- maxpivs
  - lpstats\_struct, [41](#)
- maxrnk
  - lpstats\_struct, [42](#)
- maxrowlen
  - consys\_struct, [23](#)
- maxrowndx
  - consys\_struct, [23](#)
- maxsiz
  - lpstats\_struct, [43](#)
- maxx
  - dylib\_std.h, [121](#)
- mem\_count
  - ENV, [25](#)
- mem\_cpeak
  - ENV, [26](#)
- mem\_limit
  - ENV, [25](#)
- mem\_ptr
  - ENV, [25](#)
- mem\_total
  - ENV, [25](#)
- mem\_tpeak
  - ENV, [25](#)
- mergeBasis
  - OsiDyIpWarmStartBasis, [70](#)
- min

- keytab\_entry\_internal, [29](#)
- lpstats\_struct, [41](#)
- min\_pivtol
  - lpstats\_struct, [42](#)
- min\_vrratio
  - INV, [28](#)
- minaij
  - consys\_struct, [24](#)
- minn
  - dylib\_std.h, [121](#)
- mkaref
  - dylib\_bnfdr.h, [108](#)
- mkcref
  - dylib\_bnfdr.h, [108](#)
- mkoff
  - dylib\_bnfdr.h, [107](#)
- mksav
  - dylib\_bnfdr.h, [107](#)
- mstrcmp
  - dylib\_strrtns.h, [122](#)
- mtx
  - consys\_struct, [23](#)
- N
- n
  - LUF, [46](#)
- NP
  - bnfdef\_any, [11](#)
  - bnfref\_any, [16](#)
- NULLP
  - dylib\_bnfdr.h, [107](#)
- ndcd
  - bnfLBdef\_struct, [13](#)
- ndsrc
  - bnfLBdef\_struct, [14](#)
- ndx
  - colhdr\_struct\_tag, [20](#)
  - pkcoeff\_struct, [74](#)
  - pkvec\_struct, [74](#)
  - rowhdr\_struct\_tag, [76](#)
- new\_sva
  - LUF, [48](#)
- newLanguage
  - OsiDyLpSolverInterface, [63](#)
- next
  - hel\_tag, [26](#)
  - MEM, [51](#)
- nmcd
  - bnfLBdef\_struct, [13](#)
- nme
  - colhdr\_struct\_tag, [20](#)
  - consys\_struct, [22](#)
  - pkvec\_struct, [74](#)
  - rowhdr\_struct\_tag, [76](#)
- nmsrc
  - bnfLBdef\_struct, [14](#)
- nnz\_a
  - LUF, [49](#)
- nnz\_f
  - LUF, [49](#)
- nnz\_h
  - INV, [28](#)
- nnz\_v
  - LUF, [49](#)
- nontrivial
  - lpstats\_struct, [42](#)
- npdef
  - dylib\_bnfdr.h, [108](#)
- npref
  - dylib\_bnfdr.h, [109](#)
- numberActiveConstraints
  - OsiDyLpWarmStartBasis, [69](#)
- nxt
  - attvhdr\_struct\_tag, [9](#)
- O
- ODSI\_ACCESS\_STALE
  - OsiDyLpMessages.hpp, [124](#)
- ODSI\_ALLDYLP
  - OsiDyLpMessages.hpp, [123](#)
- ODSI\_ATTACH
  - OsiDyLpMessages.hpp, [124](#)
- ODSI\_BADACTIVEBASIS
  - OsiDyLpMessages.hpp, [124](#)
- ODSI\_BADSTATE
  - OsiDyLpMessages.hpp, [124](#)
- ODSI\_COLD
  - OsiDyLpMessages.hpp, [123](#)
- ODSI\_CONFUSION
  - OsiDyLpMessages.hpp, [124](#)
- ODSI\_CWSBREJECT
  - OsiDyLpMessages.hpp, [123](#)
- ODSI\_DETACH
  - OsiDyLpMessages.hpp, [124](#)
- ODSI\_DUMMY\_END
  - OsiDyLpMessages.hpp, [124](#)
- ODSI\_EMPTYODWSB
  - OsiDyLpMessages.hpp, [123](#)
- ODSI\_FAILEDCALL
  - OsiDyLpMessages.hpp, [124](#)
- ODSI\_HOT
  - OsiDyLpMessages.hpp, [123](#)
- ODSI\_IGNOREDHINT
  - OsiDyLpMessages.hpp, [123](#)
- ODSI\_MPSFILEIO
  - OsiDyLpMessages.hpp, [123](#)
- ODSI\_NOSOLVE
  - OsiDyLpMessages.hpp, [124](#)

- ODSI\_NOTFULLSYS
  - OsiDyIplMessages.hpp, [124](#)
- ODSI\_NOTODWSB
  - OsiDyIplMessages.hpp, [123](#)
- ODSI\_NOTOPTIMAL
  - OsiDyIplMessages.hpp, [124](#)
- ODSI\_NOTOWNER
  - OsiDyIplMessages.hpp, [124](#)
- ODSI\_NOTSIMPLEX
  - OsiDyIplMessages.hpp, [124](#)
- ODSI\_NOTVALID
  - OsiDyIplMessages.hpp, [124](#)
- ODSI\_ODWSBBADSIZE
  - OsiDyIplMessages.hpp, [123](#)
- ODSI\_ODWSBBADSTATUS
  - OsiDyIplMessages.hpp, [123](#)
- ODSI\_ODWSBSHORTBASIS
  - OsiDyIplMessages.hpp, [123](#)
- ODSI\_POSTSOL
  - OsiDyIplMessages.hpp, [123](#)
- ODSI\_POSTSOL\_ACT
  - OsiDyIplMessages.hpp, [123](#)
- ODSI\_PRE SOL\_PASS
  - OsiDyIplMessages.hpp, [123](#)
- ODSI\_PRE SOL\_STATS
  - OsiDyIplMessages.hpp, [123](#)
- ODSI\_SHORTSTATS
  - OsiDyIplMessages.hpp, [124](#)
- ODSI\_TABLEAU\_INIT\_FAIL
  - OsiDyIplMessages.hpp, [124](#)
- ODSI\_TEST\_MSG
  - OsiDyIplMessages.hpp, [123](#)
- ODSI\_UNSUPFORCEDO
  - OsiDyIplMessages.hpp, [123](#)
- ODSI\_WARM
  - OsiDyIplMessages.hpp, [123](#)
- ODSI\_start\_enum
  - OsiDyIplSolverInterface.hpp, [125](#)
- obj
  - consys\_struct, [24](#)
  - lpprob\_struct, [37](#)
- objndx
  - consys\_struct, [24](#)
- objnme
  - consys\_struct, [24](#)
- offset
  - bnfLBdef\_struct, [14](#)
  - bnfref\_type2, [17](#)
  - bnfref\_type3, [17](#)
- offset2
  - bnfLBdef\_struct, [14](#)
- operator=
  - OsiDyIplSolverInterface, [58](#)
  - OsiDyIplWarmStartBasisDiff, [72](#)
- opts
  - consys\_struct, [22](#)
- OsiDyIplMessages.hpp
  - ODSI\_ACCESS\_STALE, [124](#)
  - ODSI\_ALLDYLP, [123](#)
  - ODSI\_ATTACH, [124](#)
  - ODSI\_BADACTIVEBASIS, [124](#)
  - ODSI\_BADSTATE, [124](#)
  - ODSI\_COLD, [123](#)
  - ODSI\_CONFUSION, [124](#)
  - ODSI\_CWSBREJECT, [123](#)
  - ODSI\_DETACH, [124](#)
  - ODSI\_DUMMY\_END, [124](#)
  - ODSI\_EMPTYODWSB, [123](#)
  - ODSI\_FAILEDCALL, [124](#)
  - ODSI\_HOT, [123](#)
  - ODSI\_IGNOREDHINT, [123](#)
  - ODSI\_MPSFILEIO, [123](#)
  - ODSI\_NOSOLVE, [124](#)
  - ODSI\_NOTFULLSYS, [124](#)
  - ODSI\_NOTODWSB, [123](#)
  - ODSI\_NOTOPTIMAL, [124](#)
  - ODSI\_NOTOWNER, [124](#)
  - ODSI\_NOTSIMPLEX, [124](#)
  - ODSI\_NOTVALID, [124](#)
  - ODSI\_ODWSBBADSIZE, [123](#)
  - ODSI\_ODWSBBADSTATUS, [123](#)
  - ODSI\_ODWSBSHORTBASIS, [123](#)
  - ODSI\_POSTSOL, [123](#)
  - ODSI\_POSTSOL\_ACT, [123](#)
  - ODSI\_PRE SOL\_PASS, [123](#)
  - ODSI\_PRE SOL\_STATS, [123](#)
  - ODSI\_SHORTSTATS, [124](#)
  - ODSI\_TABLEAU\_INIT\_FAIL, [124](#)
  - ODSI\_TEST\_MSG, [123](#)
  - ODSI\_UNSUPFORCEDO, [123](#)
  - ODSI\_WARM, [123](#)
- OsiDyIplSolverInterface.hpp
  - startCold, [125](#)
  - startHot, [125](#)
  - startInvalid, [125](#)
  - startWarm, [125](#)
- OsiDyIplMessageID\_enum
  - OsiDyIplMessages.hpp, [123](#)
- OsiDyIplMessages.hpp
  - OsiDyIplMessageID\_enum, [123](#)
- OsiDyIplSolverInterface, [51](#)
  - ~OsiDyIplSolverInterface, [57](#)
  - activateRowCutDebugger, [65](#)
  - addCol, [61](#)
  - addRow, [61](#)
  - applyColCut, [62](#)
  - applyRowCut, [61](#)

assignProblem, 58  
 balance, 67  
 basis, 66  
 basisIsAvailable, 65  
 branchAndBound, 66  
 canDoSimplexInterface, 64  
 clone, 57  
 condition, 67  
 deleteCols, 61  
 deleteRows, 61  
 disableFactorization, 64  
 dylp\_controlfile, 66  
 dylp\_logfile, 66  
 dylp\_outfile, 66  
 dylp\_printsoln, 66  
 enableFactorization, 64  
 getBInvACol, 65  
 getBInvARow, 65  
 getBInvCol, 65  
 getBInvRow, 65  
 getBasics, 65  
 getBasisStatus, 65  
 getColLower, 59  
 getColSolution, 64  
 getColUpper, 59  
 getDbIParam, 63  
 getDualRays, 64  
 getEmptyWarmStart, 62  
 getHintParam, 63  
 getInfinity, 63  
 getIntParam, 63  
 getIterationCount, 63  
 getMatrixByCol, 60  
 getMatrixByRow, 60  
 getNumCols, 59  
 getNumElements, 59  
 getNumIntegers, 59  
 getNumRows, 59  
 getObjCoefficients, 60  
 getObjSense, 60  
 getObjValue, 64  
 getPrimalRays, 64  
 getReducedCost, 64  
 getReducedGradient, 65  
 getRightHandSide, 59  
 getRowActivity, 64  
 getRowLower, 59  
 getRowPrice, 64  
 getRowRange, 59  
 getRowSense, 59  
 getRowUpper, 60  
 getStrParam, 63  
 getWarmStart, 62  
 initialSolve, 62  
 initialSolveOptions, 66  
 isAbandoned, 62  
 isBinary, 59  
 isContinuous, 59  
 isDualObjectiveLimitReached, 63  
 isInteger, 59  
 isIntegerNonBinary, 59  
 isIterationLimitReached, 62  
 isPrimalObjectiveLimitReached, 63  
 isProvenDualInfeasible, 62  
 isProvenOptimal, 62  
 isProvenPrimalInfeasible, 62  
 loadProblem, 58  
 markHotStart, 62  
 newLanguage, 63  
 operator=, 58  
 OsiDylpSolverInterface, 57  
 OsiDylpSolverInterfaceUnitTest, 66  
 OsiDylpSolverInterface, 57  
 readMps, 58  
 reset, 58  
 resolve, 62  
 resolveOptions, 66  
 saved\_fullsys, 67  
 setBasisStatus, 65  
 setColLower, 60  
 setColName, 60  
 setColSolution, 61  
 setColUpper, 60  
 setContinuous, 60  
 setDbIParam, 63  
 setHintParam, 63  
 setIntParam, 63  
 setInteger, 60  
 setLanguage, 64  
 setObjCoeff, 61  
 setObjName, 60  
 setObjSense, 61  
 setObjective, 61  
 setOsiDylpMessages, 66  
 setRowLower, 60  
 setRowName, 60  
 setRowPrice, 61  
 setRowType, 61  
 setRowUpper, 61  
 setStrParam, 63  
 setWarmStart, 62  
 simplex, 67  
 solveFromHotStart, 62  
 tolerances, 66  
 unmarkHotStart, 62  
 writeMps, 58  
 OsiDylpSolverInterface.hpp  
 DYLP\_INTERNAL, 125

- ODSI\_start\_enum, 125
- OsiDyLpSolverInterfaceUnitTest
  - OsiDyLpSolverInterface, 66
- OsiDyLpWarmStartBasis, 67
  - ~OsiDyLpWarmStartBasis, 69
  - applyDiff, 70
  - assignBasisStatus, 71
  - checkBasis, 71
  - clone, 71
  - compressRows, 70
  - deleteRows, 70
  - generateDiff, 70
  - getConStatus, 69
  - getConstraintStatus, 69
  - getPhase, 70
  - mergeBasis, 70
  - numberActiveConstraints, 69
  - operator=, 71
  - OsiDyLpWarmStartBasis, 69
  - OsiDyLpWarmStartBasis, 69
  - print, 71
  - resize, 70
  - setConStatus, 69
  - setPhase, 70
  - setSize, 70
- OsiDyLpWarmStartBasis.hpp
  - DYLP\_INTERNAL, 126
- OsiDyLpWarmStartBasis::applyDiff
  - OsiDyLpWarmStartBasisDiff, 73
- OsiDyLpWarmStartBasis::generateDiff
  - OsiDyLpWarmStartBasisDiff, 73
- OsiDyLpWarmStartBasisDiff, 71
  - ~OsiDyLpWarmStartBasisDiff, 72
  - clone, 72
  - operator=, 72
  - OsiDyLpWarmStartBasis::applyDiff, 73
  - OsiDyLpWarmStartBasis::generateDiff, 73
- owner
  - lpprob\_struct, 37
- P
- P
  - bnfdef\_any, 11
  - bnfref\_any, 16
- p0\_col
  - INV, 28
- p0\_row
  - INV, 27
- p1
  - lpstats\_struct, 43
- p2
  - lpstats\_struct, 43
- p2d
  - lpopts\_struct, 35
- POOL, 75
  - avail, 75
  - count, 75
  - glplib.h, 99
  - link, 75
  - size, 75
  - stock, 75
  - used, 75
- parm1
  - bnfTdef\_struct, 18
- parse
  - dylib\_bnfdr.h, 113
- parse\_any, 73
  - c, 73
  - g, 73
- parts
  - consys\_struct, 22
- patch
  - lpopts\_struct, 33
- pchk
  - lptols\_struct, 44
- pdef
  - dylib\_bnfdr.h, 108
- pdegen
  - lpstats\_struct, 43
- pfeas
  - lptols\_struct, 44
- pfeas\_scale
  - lptols\_struct, 44
- phase
  - lpprob\_struct, 37
- phase1
  - lpopts\_struct, 35
- phase2
  - lpopts\_struct, 36
- phasecnts
  - lpstats\_struct, 40
- piv\_lim
  - LUF, 48
- piv\_tol
  - LUF, 48
- pivot
  - lptols\_struct, 45
- pivoting
  - lpopts\_struct, 35
- pivrej
  - lpstats\_struct, 42
- pivreject
  - lpopts\_struct, 35
- pivrnks
  - lpstats\_struct, 42
- pivs
  - lpstats\_struct, 43
- pivtol\_red

- lpstats\_struct, 42
- pkcoeff\_struct, 73
  - ndx, 74
  - val, 74
- pkvec\_2norm
  - dy\_vector.h, 85
- pkvec\_check
  - dy\_vector.h, 85
- pkvec\_dotexvec
  - dy\_vector.h, 86
- pkvec\_free
  - dy\_vector.h, 85
- pkvec\_new
  - dy\_vector.h, 85
- pkvec\_resize
  - dy\_vector.h, 85
- pkvec\_struct, 74
  - cnt, 74
  - coeffs, 75
  - dflt, 74
  - dim, 74
  - ndx, 74
  - nme, 74
  - size, 74
- pmulti
  - lpstats\_struct, 43
- pp\_col
  - LUF, 47
- pp\_row
  - LUF, 47
- ppsel
  - lpopts\_struct, 32
- pref
  - dylib\_bnrdr.h, 110
- prev
  - MEM, 51
- prevpiv
  - lpstats\_struct, 41
- pricing
  - lpopts\_struct, 35
- print
  - glplib.h, 98, 100
  - lpopts\_struct, 36
  - OsiDyIpWarmStartBasis, 71
- promote
  - lpstats\_struct, 42
- prtbfnfdef
  - dylib\_bnrdr.h, 113
- prtbfnfref
  - dylib\_bnrdr.h, 113
- puntcall
  - lpstats\_struct, 42
- puntret
  - lpstats\_struct, 42
- purge
  - lptols\_struct, 45
- purgevar
  - lptols\_struct, 45
- pvecclst
  - attvhdr\_struct\_tag, 9
- Q
- qechr
  - bnfTdef\_struct, 18
- qq\_col
  - LUF, 47
- qq\_row
  - LUF, 47
- qschr
  - bnfTdef\_struct, 18
- R
- REALLOC
  - dylib\_std.h, 121
- rank
  - LUF, 49
- rays
  - lpopts\_struct, 36
- rbdef
  - dylib\_bnrdr.h, 109
- rbref
  - dylib\_bnrdr.h, 110
- rbrefdbg
  - dylib\_bnrdr.h, 111
- rdrclear
  - dylib\_bnrdr.h, 113
- rdrinit
  - dylib\_bnrdr.h, 113
- read\_pointer
  - glplib.h, 98, 99
- readMps
  - OsiDyIpSolverInterface, 58
- real
  - dylib\_fortran.h, 115
- reframe
  - lptols\_struct, 45
- reset
  - OsiDyIpSolverInterface, 58
- resize
  - OsiDyIpWarmStartBasis, 70
- resolve
  - OsiDyIpSolverInterface, 62
- resolveOptions
  - OsiDyIpSolverInterface, 66
- revs
  - conbnd\_struct, 20
- rfdef
  - dylib\_bnrdr.h, 109
- rfref

- dylib\_bnfdr.h, 110
- rfrefdbg
  - dylib\_bnfdr.h, 111
- rhs
  - consys\_struct, 24
- rhslow
  - consys\_struct, 24
- rowhdr
  - coeff\_struct\_tag, 19
- rowhdr\_struct
  - dy\_consys.h, 82
- rowhdr\_struct\_tag, 76
  - coeffs, 76
  - len, 76
  - ndx, 76
  - nme, 76
- rownxt
  - coeff\_struct\_tag, 19
- rows
  - conmtx\_struct, 21
- rowscale
  - consys\_struct, 24
- rowsez
  - consys\_struct, 23
  - lpprob\_struct, 38
- rs\_head
  - LUF\_WA, 50
- rs\_max
  - LUF\_WA, 50
- rs\_next
  - LUF\_WA, 50
- rs\_prev
  - LUF\_WA, 50
- S
- STRALLOC
  - dylib\_strrtns.h, 122
- STRFREE
  - dylib\_strrtns.h, 122
- save\_pointer
  - glplib.h, 98, 99
- saved\_fullsys
  - OsiDyIpSolverInterface, 67
- savnd
  - bnfLBdef\_struct, 14
- savnm
  - bnfLBdef\_struct, 13
- scaling
  - lpopts\_struct, 33
- scan
  - lpopts\_struct, 32
- sep
  - bnfref\_type3, 17
- setBasisStatus
  - OsiDyIpSolverInterface, 65
- setColLower
  - OsiDyIpSolverInterface, 60
- setColName
  - OsiDyIpSolverInterface, 60
- setColSolution
  - OsiDyIpSolverInterface, 61
- setColUpper
  - OsiDyIpSolverInterface, 60
- setConStatus
  - OsiDyIpWarmStartBasis, 69
- setContinuous
  - OsiDyIpSolverInterface, 60
- setDblParam
  - OsiDyIpSolverInterface, 63
- setHintParam
  - OsiDyIpSolverInterface, 63
- setIntParam
  - OsiDyIpSolverInterface, 63
- setInteger
  - OsiDyIpSolverInterface, 60
- setLanguage
  - OsiDyIpSolverInterface, 64
- setObjCoeff
  - OsiDyIpSolverInterface, 61
- setObjName
  - OsiDyIpSolverInterface, 60
- setObjSense
  - OsiDyIpSolverInterface, 61
- setObjective
  - OsiDyIpSolverInterface, 61
- setOsiDyIpMessages
  - OsiDyIpSolverInterface, 66
- setPhase
  - OsiDyIpWarmStartBasis, 70
- setRowLower
  - OsiDyIpSolverInterface, 60
- setRowName
  - OsiDyIpSolverInterface, 60
- setRowPrice
  - OsiDyIpSolverInterface, 61
- setRowType
  - OsiDyIpSolverInterface, 61
- setRowUpper
  - OsiDyIpSolverInterface, 61
- setSize
  - OsiDyIpWarmStartBasis, 70
- setStrParam
  - OsiDyIpSolverInterface, 63
- setWarmStart
  - OsiDyIpSolverInterface, 62
- setflg
  - dylib\_std.h, 120
- setup

- lpopts\_struct, 35
- simplex
  - OsiDyIpSolverInterface, 67
- sing
  - lpstats\_struct, 42
- size
  - bnfGdef\_struct, 12
  - MEM, 51
  - POOL, 75
- soln
  - lpopts\_struct, 36
- solveFromHotStart
  - OsiDyIpSolverInterface, 62
- startCold
  - OsiDyIpSolverInterface.hpp, 125
- startHot
  - OsiDyIpSolverInterface.hpp, 125
- startInvalid
  - OsiDyIpSolverInterface.hpp, 125
- startWarm
  - OsiDyIpSolverInterface.hpp, 125
- status
  - lpprob\_struct, 37
- stock
  - POOL, 75
- stralloc
  - dylib\_strrtns.h, 122
- strat
  - lpopts\_struct, 32
- strfree
  - dylib\_strrtns.h, 122
- string
  - lex\_struct, 29
- strsave
  - dylib\_strrtns.h, 122
- suhl
  - LUF, 48
- sv\_beg
  - LUF, 48
- sv\_end
  - LUF, 48
- sv\_head
  - LUF, 48
- sv\_ndx
  - LUF, 48
- sv\_next
  - LUF, 48
- sv\_prev
  - LUF, 48
- sv\_size
  - LUF, 47
- sv\_tail
  - LUF, 48
- sv\_val
  - LUF, 48
- swing
  - lptols\_struct, 45
- size
  - lpstats\_struct, 40
  - pkvec\_struct, 74
- T
  - bnfdef\_any, 11
  - bnfref\_any, 16
- t1
  - bnfref\_any, 15
- t2
  - bnfref\_any, 16
- t3
  - bnfref\_any, 16
- TRUE
  - dylib\_std.h, 120
- TRUEL
  - dylib\_fortran.h, 114
- tableau
  - lpopts\_struct, 36
- tdef
  - dylib\_bnfrdr.h, 108
- tiny
  - consys\_struct, 22
- token
  - keytab\_entry\_internal, 29
- tolerances
  - OsiDyIpSolverInterface, 66
- toobig
  - lptols\_struct, 45
- tot
  - lpstats\_struct, 43
- totpivs
  - lpstats\_struct, 43
- tqdef
  - dylib\_bnfrdr.h, 108
- tref
  - dylib\_bnfrdr.h, 110
- ttype
  - bnfTdef\_struct, 18
- txt
  - bnfLdef\_struct, 14
- U
  - UNUSED
    - dylib\_std.h, 120
- ucalloc
  - glplib.h, 98, 100
- ufree
  - glplib.h, 98, 100
- umalloc
  - glplib.h, 98, 100



- unmarkHotStart
  - OsiDyIpSolverInterface, 62
- upd\_tol
  - INV, 28
- used
  - POOL, 75
- usedual
  - lpopts\_struct, 33
- V
- VALID\_ATTVMTYPE
  - dy\_consys.h, 81
- VALID\_CONTYPE
  - dy\_consys.h, 81
- VALID\_STATUS
  - dylp.h, 90
- VALID\_VARTYPE
  - dy\_consys.h, 81
- val
  - bnfTdef\_struct, 18
  - coeff\_struct\_tag, 19
  - pkcoeff\_struct, 74
- valid
  - INV, 27
  - LUF, 46
- varcnt
  - consys\_struct, 22
- varmgmt
  - lpopts\_struct, 36
- varname\_code
  - dylib\_fortran.h, 116
- vars
  - lpopts\_struct, 34, 35
  - lpstats\_struct, 41
- vartypBIN
  - dy\_consys.h, 83
- vartypCON
  - dy\_consys.h, 83
- vartypINT
  - dy\_consys.h, 83
- vartypINV
  - dy\_consys.h, 83
- vartyp\_enum
  - dy\_consys.h, 83
- vc\_cap
  - LUF, 47
- vc\_len
  - LUF, 47
- vc\_ptr
  - LUF, 47
- vec
  - attvhdr\_struct\_tag, 9
- vib
  - consys\_struct, 24
- vndx
  - basisel\_struct, 10
- vr\_cap
  - LUF, 47
- vr\_len
  - LUF, 47
- vr\_piv
  - LUF, 47
- vr\_ptr
  - LUF, 47
- vstatB
  - dylp.h, 89
- vstatBASIC
  - dylp.h, 90
- vstatBFR
  - dylp.h, 89
- vstatBFX
  - dylp.h, 89
- vstatBLB
  - dylp.h, 89
- vstatBLLB
  - dylp.h, 90
- vstatBUB
  - dylp.h, 89
- vstatBUUB
  - dylp.h, 90
- vstatEXOTIC
  - dylp.h, 90
- vstatINV
  - dylp.h, 89
- vstatNBFR
  - dylp.h, 89
- vstatNBFX
  - dylp.h, 89
- vstatNBLB
  - dylp.h, 89
- vstatNBUB
  - dylp.h, 89
- vstatNOLOAD
  - dylp.h, 90
- vstatNONBASIC
  - dylp.h, 90
- vstatNOPER
  - dylp.h, 90
- vstatNOPIVOT
  - dylp.h, 90
- vstatQUALS
  - dylp.h, 90
- vstatSB
  - dylp.h, 90
- vstatSTATUS
  - dylp.h, 90
- vtyp
  - consys\_struct, 24

vub

consys\_struct, [24](#)

W

warn

dylib\_errs.h, [113](#)

watch

gplib.h, [98](#), [100](#)

what

attvhdr\_struct\_tag, [9](#)

work

LUF, [48](#)

writeMps

OsiDyIpSolverInterface, [58](#)

X

x

lpprob\_struct, [38](#)

xzndx

consys\_struct, [24](#)

Y

y

lpprob\_struct, [38](#)

Z

zero

lptols\_struct, [44](#)