hypre Reference Manual

— Version 1.14.0b —

Contents

1	Matr	ix and Vector Views (Conceptual Interfaces) —	5			
	1.1	IJ Matrix View —	5			
	1.2	IJ Vector View —				
	1.3	Struct Matrix View —	15			
	1.4	Struct Vector View —	18			
	1.5	SemiStructured Matrix View —	20			
	1.6	SemiStructured Vector View —	26			
2	Oper	rator Interface —	32			
3	Vecto	or Interface —	36			
4		rices and Vectors —	39			
	4.1	IJParCSR Matrix —	39			
	4.2	IJParCSR Vector —	50			
	4.3	Struct Matrix —	56			
	4.4	Struct Vector —				
	4.5	SemiStructured Matrix —				
	4.6	SemiStructured Vector —				
	4.7	SemiStructured ParCSR Matrix —				
	4.8	SemiStructured ParCSR Vector —	96			
5	Solve 5.9	er Interface —				
	5.10	Hybrid Solver —				
	5.10	nybrid Sorver —	110			
6		SR Matrix Solvers — Linear solvers for sparse matrix systems				
	6.1	ParCSRDiagScale Solver —				
	6.2	ParCSR BoomerAMG Solver —				
	6.3	ParCSR Euclid Solver —				
	6.4	ParCSR Schwarz Solver —				
	6.5	ParCSR ParaSails Solver —				
	6.6	ParCSR Pilut Solver —	157			
7	Struc	Structured Matrix Solvers — Linear solvers for struct matrix systems				
	7.1	StructDiagScale Solver —	163			
	7.2	Struct Jacobi Solver —	170			
	7.3	Struct PFMG Solver —	177			
	7.4	Struct SMG Solver —	183			
8		Structured Matrix Solvers — Linear solvers for semi-struct matrix systems	190			
	8.1	SemiStruct DiagScale Solver —	190			
	8.2	Struct Split Solver —	197			

hypre Reference Manual

9	Prece	onditionedSolver Interface —	204
10	Preco	PCG Preconditioned Solver —	
	10.2	GMRES Preconditioned Solver — $\hfill \ldots \hfill \ldots \hfill \ldots \hfill \ldots \hfill \hfi$	213
	10.3	BiCGSTAB Preconditioned Solver —	220
	10.4	CGNR Preconditioned Solver —	227
11	Other 11.1	r —	
12	Struc 12.1	t Grid, etc. —	
	12.2	Struct Stencil —	240
13	Semi-Structured Grid, etc. —		
	13.1	Semi-Structured Graph —	244
	13.2	${\bf Semi-Structured\ Grid\\qquad }$	249
	13.3	Semi-Structured Stencil —	254
	Class	Granh	257

Copyright (c) 2006 The Regents of the University of California. Produced at the Lawrence Livermore National Laboratory. Written by the HYPRE team. UCRL-CODE-222953. All rights reserved.

This file is part of HYPRE (see http://www.llnl.gov/CASC/hypre/). Please see the COPY-RIGHT_and_LICENSE file for the copyright notice, disclaimer, contact information and the GNU Lesser General Public License.

HYPRE is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License (as published by the Free Software Foundation) version 2.1 dated February 1999.

HYPRE is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the IMPLIED WARRANTY OF MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the terms and conditions of the GNU General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

1

Matrix and Vector Views (Conceptual Interfaces)

Names		
1.1	IJ Matrix View	
		Ę
1.2	IJ Vector View	
		10
1.3	Struct Matrix View	
		15
1.4	Struct Vector View	
		18
1.5	SemiStructured Matrix View	
		20
1.6	SemiStructured Vector View	
		26

1.1

IJ Matrix View

\mathbf{Names}		
1.1.1	struct bHYPRE_IJMatrixViewobject Symbol "bHYPRE	7
1.1.2	extern C bHYPRE_IJMatrixView bHYPRE_IJMatrixViewconnect (const char *, sidl_BaseInterface *_ex) RMI connector function for the class	7
1.1.3	SIDL_C_INLINE_DECL int32_t bHYPRE_IJMatrixView_SetLocalRange (bHYPRE_IJMatrixView self,	8
1.1.4	int32_t bHYPRE_IJMatrixView_SetValues (bHYPRE_IJMatrixView self,	8
1.1.5	$\mathrm{int}32$ _t	

	bHYPRE_IJMatrixView_AddToValues (bHYPRE_IJMatrixView self, int32_t nrows, int32_t* ncols, int32_t* rows, int32_t* cols, double* values, int32_t nnonzeros, sidl_BaseInterface *_ex)	
	Adds to values for nrows of the matrix	9
	SIDL_C_INLINE_DECL int32_t bHYPRE_IJMatrixView_GetLocalRange (bHYPRE_IJMatrixView self, int32_t* ilower, int32_t* iupper, int32_t* ilower, int32_t* iupper,	
	int32_t* jlower, int32_t* jupper, sidl_BaseInterface *_ex)	
	Gets range of rows owned by this processor and range of column partitioning for this processor	
	$\mathrm{int}32$ _{t}	
	bHYPRE_IJMatrixView_GetRowCounts (bHYPRE_IJMatrixView self, int32_t nrows, int32_t* rows, int32_t* ncols, sidl_BaseInterface *_ex)	
	Gets number of nonzeros elements for nrows rows specified in rows and returns them in ncols, which needs to be allocated by the user	
1.1.6	int32_t bHYPRE_IJMatrixView_GetValues (bHYPRE_IJMatrixView self, int32_t nrows, int32_t* ncols,	
	int32_t* rows, int32_t* cols, double* values, int32_t nnonzeros, sidl_BaseInterface *_ex)	
	Gets values for nrows rows or partial rows of the matrix	9
1.1.7	SIDL_C_INLINE_DECL int32_t bHYPRE_IJMatrixView_SetRowSizes (bHYPRE_IJMatrixView self, int32_t* sizes, int32_t nrows,	
	sidl_BaseInterface *_ex) (Optional) Set the max number of nonzeros to expect in each row	9
1.1.8	SIDL_C_INLINE_DECL int32_t	Ü
	bHYPRE_IJMatrixView_Print (bHYPRE_IJMatrixView self, const char* filename, sidl_BaseInterface *_ex)	
	Print the matrix to file	10
1.1.9	SIDL_C_INLINE_DECL_int32_t bHYPRE_IJMatrixView_Read (bHYPRE_IJMatrixView self,	
	const char* filename, bHYPRE_MPICommunicator comm, sidl_BaseInterface *_ex)	
	Read the matrix from file	10
	_ex Cast method for interface and class type conversions	
	void*	
	bHYPRE_IJMatrixViewcast2 (void* obj, const char* type, sidl_BaseInterface *_ex)	
	String cast method for interface and class type conversions	
	SIDL_C_INLINE_DECL void	

bHYPRE_IJMatrixView_exec (bHYPRE_IJMatrixView self, const char* methodName, sidl_rmi_Call inArgs, sidl_rmi_Return outArgs, sidl_BaseInterface *_ex) Select and execute a method by name SIDL_C_INLINE_DECL char* bHYPRE_IJMatrixView__getURL (bHYPRE_IJMatrixView self, sidl_BaseInterface *_ex) Get the URL of the Implementation of this object (for RMI) SIDL_C_INLINE_DECL_void bHYPRE_IJMatrixView__raddRef (bHYPRE_IJMatrixView self, sidl_BaseInterface *_ex) On a remote object, addrefs the remote instance SIDL_C_INLINE_DECL sidl_bool bHYPRE_IJMatrixView__isRemote (bHYPRE_IJMatrixView self, sidl_BaseInterface *_ex) TRUE if this object is remote, false if local

sidl_bool

bHYPRE_IJMatrixView__isLocal (bHYPRE_IJMatrixView self, sidl_BaseInterface *_ex)

TRUE if this object is remote, false if local

 $**_ex$

 $Cast\ method\ for\ interface\ and\ class\ type\ conversions$

1.1.10 **_ex

RMI connector function for the class

1.1.1

struct bHYPRE_IJMatrixView_object

Symbol "bHYPRE.IJMatrixView" (version 1.0.0)

This interface represents a linear-algebraic conceptual view of a linear system. The 'I' and 'J' in the name are meant to be mnemonic for the traditional matrix notation A(I,J).

1.1.2

extern C bHYPRE_IJMatrixView bHYPRE_IJMatrixView_connect (const char *, sidl_BaseInterface *_ex)

10

RMI connector function for the class.(addrefs)

1.1.3

SIDL_C_INLINE_DECL int32_t bHYPRE_IJMatrixView_SetLocalRange (bHYPRE_IJMatrixView self, int32_t ilower, int32_t ilower, int32_t jlower, int32_t jupper, sidl_BaseInterface *_ex)

Set the local range for a matrix object. Each process owns some unique consecutive range of rows, indicated by the global row indices ilower and iupper. The row data is required to be such that the value of ilower on any process p be exactly one more than the value of iupper on process p-1. Note that the first row of the global matrix may start with any integer value. In particular, one may use zero- or one-based indexing.

For square matrices, jlower and jupper typically should match ilower and iupper, respectively. For rectangular matrices, jlower and jupper should define a partitioning of the columns. This partitioning must be used for any vector v that will be used in matrix-vector products with the rectangular matrix. The matrix data structure may use jlower and jupper to store the diagonal blocks (rectangular in general) of the matrix separately from the rest of the matrix.

Collective.

1.1.4

int32_t bHYPRE_IJMatrixView_SetValues (bHYPRE_IJMatrixView self, int32_t nrows, int32_t* ncols, int32_t* rows, int32_t* cols, double* values, int32_t nnonzeros, sidl_BaseInterface *_ex)

Sets values for nrows of the matrix. The arrays ncols and rows are of dimension nrows and contain the number of columns in each row and the row indices, respectively. The array cols contains the column indices for each of the rows, and is ordered by rows. The data in the values array corresponds directly to the column entries in cols. The last argument is the size of the cols and values arrays, i.e. the total number of nonzeros being provided, i.e. the sum of all values in ncols. This functin erases any previous values at the specified locations and replaces them with new ones, or, if there was no value there before, inserts a new one.

Not collective.

1.1.5

 $int32_t$

bHYPRE_IJMatrixView_AddToValues (bHYPRE_IJMatrixView self, int32_t nrows, int32_t* ncols, int32_t* rows, int32_t* cols, double* values, int32_t nnonzeros, sidl_BaseInterface *_ex)

Adds to values for nrows of the matrix. Usage details are analogous to SetValues. Adds to any previous values at the specified locations, or, if there was no value there before, inserts a new one.

Not collective.

1.1.6

 $int32_t$

bHYPRE_IJMatrixView_GetValues (bHYPRE_IJMatrixView self, int32_t nrows, int32_t* ncols, int32_t* rows, int32_t* cols, double* values, int32_t nnonzeros, sidl_BaseInterface *_ex)

Gets values for nrows rows or partial rows of the matrix. Usage details are analogous to SetValues.

1.1.7

SIDL_C_INLINE_DECL int32_t bHYPRE_IJMatrixView_SetRowSizes (bHYPRE_IJMatrixView self, int32_t* sizes, int32_t nrows, sidl_BaseInterface *_ex)

(Optional) Set the max number of nonzeros to expect in each row. The array sizes contains estimated sizes for each row on this process. The integer nrows is the number of rows in the local matrix. This call can significantly improve the efficiency of matrix construction, and should always be utilized if possible.

Not collective.

1.1.8

SIDL_C_INLINE_DECL int32_t bHYPRE_IJMatrixView_Print (bHYPRE_IJMatrixView self, const char* filename, sidl_BaseInterface *_ex)

Print the matrix to file. This is mainly for debugging purposes.

1.1.9

SIDL_C_INLINE_DECL int32_t bHYPRE_IJMatrixView_Read (bHYPRE_IJMatrixView self, const char* filename, bHYPRE_MPICommunicator comm, sidl_BaseInterface *_ex)

Read the matrix from file. This is mainly for debugging purposes.

_ 1.1.10 ____

struct bHYPRE_IJMatrixView__object* bHYPRE_IJMatrixView__connectI const char * url sidl_bool ar struct sidl_BaseInterface__object **_ex

RMI connector function for the class. (no addref)

_ 1.2 __

Names

IJ Vector View

1.2.3 SIDL_C_INLINE_DECL int32_t

	bHYPRE_IJVectorView_SetLocalRange (bHYPRE_IJVectorView self, int32_t jlower, int32_t jupper, sidl_BaseInterface *_ex)	
	Set the local range for a vector object	13
1.2.4	int32_t bHYPRE_IJVectorView_SetValues (bHYPRE_IJVectorView self,	10
	Sets values in vector	13
1.2.5	int32_t bHYPRE_IJVectorView_AddToValues (bHYPRE_IJVectorView self,	
	Adds to values in vector	13
	SIDL_C_INLINE_DECL int32_t bHYPRE_IJVectorView_GetLocalRange (bHYPRE_IJVectorView self,	
1.2.6	$\mathrm{int}32$ _t	
	bHYPRE_IJVectorView_GetValues (bHYPRE_IJVectorView self, int32_t nvalues, int32_t* indices, double* values, sidl_BaseInterface *_ex) Gets values in vector	14
1.2.7	SIDL_C_INLINE_DECL int32_t bHYPRE_IJVectorView_Print (bHYPRE_IJVectorView self,	
	Print the vector to file	14
1.2.8	SIDL_C_INLINE_DECL int32_t bHYPRE_IJVectorView_Read (bHYPRE_IJVectorView self,	
	Read the vector from file	14
	_ex Cast method for interface and class type conversions	
	void* bHYPRE_IJVectorViewcast2 (void* obj, const char* type,	
	String cast method for interface and class type conversions	
	SIDL_C_INLINE_DECL void	

bHYPRE_IJVectorView_exec (bHYPRE_IJVectorView self, const char* methodName, sidl_rmi_Call inArgs, sidl_rmi_Return outArgs, sidl_BaseInterface *_ex) Select and execute a method by name SIDL_C_INLINE_DECL char* bHYPRE_IJVectorView__getURL (bHYPRE_IJVectorView self, sidl_BaseInterface *_ex) Get the URL of the Implementation of this object (for RMI) SIDL_C_INLINE_DECL_void bHYPRE_IJVectorView_raddRef (bHYPRE_IJVectorView self, sidl_BaseInterface *_ex) On a remote object, addrefs the remote instance SIDL_C_INLINE_DECL sidl_bool bHYPRE_IJVectorView__isRemote (bHYPRE_IJVectorView self, sidl_BaseInterface *_ex) TRUE if this object is remote, false if local $sidl_bool$ bHYPRE_IJVectorView__isLocal (bHYPRE_IJVectorView self, sidl_BaseInterface *_ex) TRUE if this object is remote, false if local **_ex Cast method for interface and class type conversions $**_ex$ 1.2.9

RMI connector function for the class

1.2.1

struct bHYPRE_IJVectorView__object

Symbol "bHYPRE.IJVectorView" (version 1.0.0)

1.2.2

extern C bHYPRE_IJVectorView **bHYPRE_IJVectorView_connect** (const char *, sidl_BaseInterface *_ex)

RMI connector function for the class.(addrefs)

14

1.2.3

SIDL_C_INLINE_DECL int32_t bHYPRE_IJVectorView_SetLocalRange (bHYPRE_IJVectorView self, int32_t jlower, int32_t jupper, sidl_BaseInterface *_ex)

Set the local range for a vector object. Each process owns some unique consecutive range of vector unknowns, indicated by the global indices jlower and jupper. The data is required to be such that the value of jlower on any process p be exactly one more than the value of jupper on process p-1. Note that the first index of the global vector may start with any integer value. In particular, one may use zero- or one-based indexing.

Collective.

1.2.4

bHYPRE_IJVectorView_SetValues (bHYPRE_IJVectorView self, int32_t nvalues, int32_t* indices, double* values, sidl_BaseInterface *_ex)

Sets values in vector. The arrays values and indices are of dimension nvalues and contain the vector values to be set and the corresponding global vector indices, respectively. Erases any previous values at the specified locations and replaces them with new ones.

Not collective.

1.2.5

bHYPRE_IJVectorView_AddToValues (bHYPRE_IJVectorView self, int32_t nvalues, int32_t* indices, double* values, sidl_BaseInterface *_ex)

Adds to values in vector. Usage details are analogous to SetValues.

Not collective.

1.2.6

 $int32_t$

bHYPRE_IJVectorView_GetValues (bHYPRE_IJVectorView self, int32_t nvalues, int32_t* indices, double* values, sidl_BaseInterface *_ex)

Gets values in vector. Usage details are analogous to SetValues.

Not collective.

1.2.7

SIDL_C_INLINE_DECL int32_t bHYPRE_IJVectorView_Print (bHYPRE_IJVectorView self, const char* filename, sidl_BaseInterface *_ex)

Print the vector to file. This is mainly for debugging purposes.

__ 1.2.8 __

SIDL_C_INLINE_DECL int32_t bHYPRE_IJVectorView_Read (bHYPRE_IJVectorView self, const char* filename, bHYPRE_MPICommunicator comm, sidl_BaseInterface *_ex)

Read the vector from file. This is mainly for debugging purposes.

__ 1.2.9 ____

RMI connector function for the class. (no addref)

1.3

Struct Matrix View

Names		
1.3.1	struct bHYPRE_StructMatrixView_object Symbol "bHYPRE	17
1.3.2	extern C bHYPRE_StructMatrixView bHYPRE_StructMatrixViewconnect (const char *,	
	RMI connector function for the class	17
	SIDL_C_INLINE_DECL_int32_t bHYPRE_StructMatrixView_SetGrid (bHYPRE_StructMatrixView self, bHYPRE_StructGrid grid,	
	$sidl_BaseInterface *_ex)$ $Method: SetGrid[]$	
	SIDL_C_INLINE_DECL_int32_t bHYPRE_StructMatrixView_SetStencil (bHYPRE_StructMatrixView self, bHYPRE_StructStencil stencil, sidl_BaseInterface *_ex)	
	Method: SetStencil[]	
	int32_t bHYPRE_StructMatrixView_SetValues (bHYPRE_StructMatrixView self, int32_t* index, int32_t dim, int32_t num_stencil_indices, int32_t* stencil_indices, double* values, sidl_BaseInterface *_ex) Method: SetValues[] int32_t	
	bHYPRE_StructMatrixView_SetBoxValues (bHYPRE_StructMatrixView self, int32_t* ilower, int32_t* iloper, int32_t dim, int32_t num_stencil_indices, int32_t* stencil_indices, double* values, int32_t nvalues, sidl_BaseInterface *_ex)	
	Method: SetBoxValues[]	
	SIDL_C_INLINE_DECL int32_t bHYPRE_StructMatrixView_SetNumGhost (bHYPRE_StructMatrixView self, int32_t* num_ghost, int32_t dim2, sidl_BaseInterface *_ex)	
	$Method:\ SetNumGhost[]$	
	SIDL C INLINE DECL. int 32 t	

```
bHYPRE_StructMatrixView_SetSymmetric ( bHYPRE_StructMatrixView
                                               self, int32_t symmetric,
                                               sidl_BaseInterface *_ex)
      Method: SetSymmetric[]
SIDL_C_INLINE_DECL int32_t
{\bf bHYPRE\_StructMatrixView\_SetConstantEntries}\ (
                                                     bHYPRE_StructMatrixView
                                                    self, int32_t
                                                     num_stencil_constant_points,
                                                     int32_t*
                                                     stencil_constant_points,
                                                     sidl_BaseInterface *_ex)
      Method: SetConstantEntries[]
int32_t
bHYPRE_StructMatrixView_SetConstantValues (
                                                    b HYPRE\_StructMatrixView
                                                    self, int32_t
                                                    num_stencil_indices,
                                                    int32_t* stencil_indices,
                                                    double* values,
                                                    sidl_BaseInterface *_ex)
      Method: SetConstantValues[]
_ex
       Cast method for interface and class type conversions
\mathrm{void}^*
bHYPRE_StructMatrixView__cast2 (void* obj, const char* type,
                                      sidl_BaseInterface *_ex)
      String cast method for interface and class type conversions
SIDL_C_INLINE_DECL void
bHYPRE_StructMatrixView_exec ( bHYPRE_StructMatrixView self,
                                      const char* methodName,
                                      sidl_rmi_Call inArgs,
                                     sidl_rmi_Return outArgs,
                                      sidl_BaseInterface *_ex)
      Select and execute a method by name
SIDL_C_INLINE_DECL char*
bHYPRE_StructMatrixView__getURL ( bHYPRE_StructMatrixView self,
                                         sidl_BaseInterface *_ex)
       Get the URL of the Implementation of this object (for RMI)
SIDL_C_INLINE_DECL void
bHYPRE_StructMatrixView_raddRef ( bHYPRE_StructMatrixView self,
                                          sidl_BaseInterface *_ex)
       On a remote object, addrefs the remote instance
SIDL_C_INLINE_DECL sidl_bool
bHYPRE_StructMatrixView_isRemote ( bHYPRE_StructMatrixView self,
                                           sidl_BaseInterface *_ex)
       TRUE if this object is remote, false if local
sidl_bool
```

	\mathbf{bHYF}	PRE_StructMatrixViewisLocal (bHYPRE_StructMatrixView self,	
		sidl_BaseInterface *_ex)	
		TRUE if this object is remote, false if local	
	**_ex		
		Cast method for interface and class type conversions	
1.3.3	**_ex		
		RMI connector function for the class	17

_ 1.3.1 ____

 $struct\ bHYPRE_StructMatrixView__object$

Symbol "bHYPRE.StructMatrixView" (version 1.0.0)

1.3.2

extern C bHYPRE_StructMatrixView **bHYPRE_StructMatrixView_connect** (const char *, sidl_BaseInterface *_ex)

RMI connector function for the class.(addrefs)

1.3.3

RMI connector function for the class. (no addref)

1.4

Struct Vector View

Names	
1.4.1	struct bHYPRE_StructVectorViewobject Symbol "bHYPRE
1.4.2	extern C bHYPRE_StructVectorView bHYPRE_StructVectorView_connect (const char *, sidl_BaseInterface *_ex) RMI connector function for the class
	SIDL_C_INLINE_DECL_int32_t bHYPRE_StructVectorView_SetGrid (bHYPRE_StructVectorView self, bHYPRE_StructGrid grid, sidl_BaseInterface *_ex)
	Method: SetGrid[]
	SIDL_C_INLINE_DECL int32_t bHYPRE_StructVectorView_SetNumGhost (bHYPRE_StructVectorView self, int32_t* num_ghost, int32_t dim2, sidl_BaseInterface *_ex)
	Method: SetNumGhost[]
	SIDL_C_INLINE_DECL_int32_t bHYPRE_StructVectorView_SetValue (bHYPRE_StructVectorView self, int32_t* grid_index, int32_t dim, double value, sidl_BaseInterface *_ex) Method: SetValue[]
	int32_t
	bHYPRE_StructVectorView_SetBoxValues (bHYPRE_StructVectorView self, int32_t* ilower, int32_t* iupper, int32_t dim, double* values, int32_t nvalues, sidl_BaseInterface *_ex)
	$Method:\ SetBoxValues[]$
	_ex Cast method for interface and class type conversions
	void^*
	bHYPRE_StructVectorViewcast2 (void* obj, const char* type, sidl_BaseInterface *_ex)
	String cast method for interface and class type conversions
	SIDL_C_INLINE_DECL_void

19

19

bHYPRE_StructVectorView_exec (bHYPRE_StructVectorView self, const char* methodName, sidl_rmi_Call inArgs, sidl_rmi_Return outArgs, sidl_BaseInterface *_ex) Select and execute a method by name SIDL_C_INLINE_DECL char* bHYPRE_StructVectorView__getURL (bHYPRE_StructVectorView self, sidl_BaseInterface *_ex) Get the URL of the Implementation of this object (for RMI) SIDL_C_INLINE_DECL_void bHYPRE_StructVectorView_raddRef (bHYPRE_StructVectorView self, sidl_BaseInterface *_ex) On a remote object, addrefs the remote instance SIDL_C_INLINE_DECL sidl_bool bHYPRE_StructVectorView__isRemote (bHYPRE_StructVectorView self, sidl_BaseInterface *_ex) TRUE if this object is remote, false if local sidl_bool bHYPRE_StructVectorView__isLocal (bHYPRE_StructVectorView self, sidl_BaseInterface *_ex) TRUE if this object is remote, false if local $**_ex$ Cast method for interface and class type conversions

RMI connector function for the class

1 4 1

1.4.3

 $**_ex$

struct bHYPRE_StructVectorView_object

Symbol "bHYPRE.StructVectorView" (version 1.0.0)

1.4.2

extern C bHYPRE_StructVectorView bHYPRE_StructVectorView__connect (const char *, sidl_BaseInterface *_ex)

RMI connector function for the class.(addrefs)

20

1.4.3

struct bHYPRE_StructVectorView__object* bHYPRE_StructVectorView__connectI const char * url sidl_bool ar struct sidl_BaseInterface__object **_ex

RMI connector function for the class. (no addref)

_ 1.5 _

SemiStructured Matrix View

Names		
1.5.1	struct bHYPRE_SStructMatrixView_object Symbol "bHYPRE	23
1.5.2	extern C bHYPRE_SStructMatrixView bHYPRE_SStructMatrixView_connect (const char *,	0.00
	RMI connector function for the class	23
1.5.3	SIDL_C_INLINE_DECL int32_t bHYPRE_SStructMatrixView_SetGraph (bHYPRE_SStructMatrixView	
	self, bHYPRE_SStructGraph graph, sidl_BaseInterface *_ex)	
	Set the matrix graph	23
1.5.4	int32_t bHYPRE_SStructMatrixView_SetValues (bHYPRE_SStructMatrixView self, int32_t part,	
	int32_t* index, int32_t dim, int32_t var, int32_t nentries, int32_t* entries, double* values, sidl_BaseInterface *_ex)	
	Set matrix coefficients index by index	23
1.5.5	$\mathrm{int}32$ _t	

	$bHYPRE_SStructMatrixView_SetBoxValues$ (
	bHYPRE_SStructMatrixView	
	self, int32_t part,	
	int32_t* ilower,	
	int32_t* iupper, int32_t dim,	
	int32_t var, int32_t nentries,	
	int32_t* entries,	
	double* values,	
	int32_t nvalues,	
	sidl_BaseInterface *_ex)	
	Set matrix coefficients a box at a time	24
1.5.6	int32_t	
	bHYPRE_SStructMatrixView_AddToValues (
	bHYPRE_SStructMatrixView	
	self, int32_t part,	
	$int32_t^* index, int32_t dim,$	
	int32_t var, int32_t nentries,	
	$int32_t^*$ entries,	
	double* values,	
	sidl_BaseInterface *_ex) Add to matrix coefficients index by index	24
	•	24
1.5.7	$int32_t$ bHYPRE_SStructMatrixView_AddToBoxValues (
	bHYPRE_SStructMatrixView_Add ToBox values (
	self, int32_t part,	
	int32_t* ilower,	
	int32_t* iupper,	
	int32_t dim, int32_t var,	
	int32_t dmi, int32_t var,	
	int32_t* entries,	
	double* values,	
	int32_t nvalues,	
	sidl_BaseInterface *_ex)	
	Add to matrix coefficients a box at a time	25
1 5 0	•	20
1.5.8	SIDL_C_INLINE_DECL int32_t bHYPRE_SStructMatrixView_SetSymmetric (
	bHYPRE_SStructMatrixView_SetSymmetric (bHYPRE_SStructMatrixView	
	self, int32_t part,	
	int32_t var, int32_t to_var,	
	int32_t symmetric, sidl_BaseInterface *_ex)	
	Define symmetry properties for the stencil entries in the matrix	25
		20
	SIDL_C_INLINE_DECL int32_t	
	bHYPRE_SStructMatrixView_SetNSSymmetric (
	bHYPRE_SStructMatrixView	
	self, int32_t symmetric,	
	sidl_BaseInterface *_ex)	
	Define symmetry properties for all non-stencil matrix entries	
	SIDL_C_INLINE_DECL int32_t	

	bHYPRE_SStructMatrixView_SetComplex (bHYPRE_SStructMatrixView self, sidl_BaseInterface *_ex)	
	Set the matrix to be complex	
1.5.9	SIDL_C_INLINE_DECL_int32_t	
1.0.0	bHYPRE_SStructMatrixView_Print (bHYPRE_SStructMatrixView self, const char* filename, int32_t all, sidl_BaseInterface *_ex)	
	Print the matrix to file	25
	_ex	
	Cast method for interface and class type conversions	
	void^*	
	bHYPRE_SStructMatrixViewcast2 (void* obj, const char* type, sidl_BaseInterface *_ex)	
	String cast method for interface and class type conversions	
	SIDL_C_INLINE_DECL_void	
	bHYPRE_SStructMatrixViewexec (bHYPRE_SStructMatrixView self, const char* methodName, sidl_rmi_Call inArgs, sidl_rmi_Return outArgs, sidl_BaseInterface *_ex)	
	Select and execute a method by name	
	SIDL_C_INLINE_DECL_char* bHYPRE_SStructMatrixViewgetURL (bHYPRE_SStructMatrixView self, sidl_BaseInterface *_ex) Get the URL of the Implementation of this object (for RMI)	
	SIDL_C_INLINE_DECL void bHYPRE_SStructMatrixViewraddRef (bHYPRE_SStructMatrixView self, sidl_BaseInterface *_ex) On a remote object, addrefs the remote instance	
	SIDL_C_INLINE_DECL_sidl_bool	
	bHYPRE_SStructMatrixViewisRemote (bHYPRE_SStructMatrixView self, sidl_BaseInterface *_ex)	
	TRUE if this object is remote, false if local	
	sidl_bool bHYPRE_SStructMatrixViewisLocal (bHYPRE_SStructMatrixView self, sidl_BaseInterface *_ex)	
	TRUE if this object is remote, false if local	
	**_ex	
	Cast method for interface and class type conversions	
1.5.10	**_ex RMI connector function for the class	26

 $_$ 1.5.1 $_$

struct bHYPRE_SStructMatrixView_object

Symbol "bHYPRE.SStructMatrixView" (version 1.0.0)

___ 1.5.2 __

extern C bHYPRE_SStructMatrixView bHYPRE_SStructMatrixView __connect (const char *, sidl_BaseInterface *_ex)

RMI connector function for the class.(addrefs)

_ 1.5.3 _

SIDL_C_INLINE_DECL int32_t bHYPRE_SStructMatrixView_SetGraph (bHYPRE_SStructMatrixView self, bHYPRE_SStructGraph graph, sidl_BaseInterface *_ex)

Set the matrix graph. DEPRECATED Use Create

1.5.4

 $int32_t$

bHYPRE_SStructMatrixView_SetValues (bHYPRE_SStructMatrixView self, int32_t part, int32_t* index, int32_t dim, int32_t var, int32_t nentries, int32_t* entries, double* values, sidl_BaseInterface *_ex)

Set matrix coefficients index by index.

NOTE: Users are required to set values on all processes that own the associated variables. This means that some data will be multiply defined.

NOTE: The entries in this routine must all be of the same type: either stencil or non-stencil, but not both. Also, if they are stencil entries, they must all represent couplings to the same variable type (there are no such restrictions for non-stencil entries).

If the matrix is complex, then values consists of pairs of doubles representing the real and imaginary parts of each complex value.

1.5.5

$int32_t$

bHYPRE_SStructMatrixView_SetBoxValues (bHYPRE_SStructMatrixView self, int32_t part, int32_t* ilower, int32_t* iupper, int32_t dim, int32_t var, int32_t nentries, int32_t* entries, double* values, int32_t nvalues, sidl_BaseInterface *_ex)

Set matrix coefficients a box at a time.

NOTE: Users are required to set values on all processes that own the associated variables. This means that some data will be multiply defined.

NOTE: The entries in this routine must all be of the same type: either stencil or non-stencil, but not both. Also, if they are stencil entries, they must all represent couplings to the same variable type (there are no such restrictions for non-stencil entries).

If the matrix is complex, then values consists of pairs of doubles representing the real and imaginary parts of each complex value.

1.5.6

int32 t

bHYPRE_SStructMatrixView_AddToValues (bHYPRE_SStructMatrixView self, int32_t part, int32_t* index, int32_t dim, int32_t var, int32_t nentries, int32_t* entries, double* values, sidl_BaseInterface *_ex)

Add to matrix coefficients index by index.

NOTE: Users are required to set values on all processes that own the associated variables. This means that some data will be multiply defined.

NOTE: The entries in this routine must all be of the same type: either stencil or non-stencil, but not both. Also, if they are stencil entries, they must all represent couplings to the same variable type.

If the matrix is complex, then values consists of pairs of doubles representing the real and imaginary parts of each complex value.

1.5.7

$int32_t$

bHYPRE_SStructMatrixView_AddToBoxValues (

bHYPRE_SStructMatrixView self, int32_t part, int32_t* ilower, int32_t* iupper, int32_t dim, int32_t var, int32_t nentries, int32_t* entries, double* values, int32_t nvalues, sidl_BaseInterface *_ex)

Add to matrix coefficients a box at a time.

NOTE: Users are required to set values on all processes that own the associated variables. This means that some data will be multiply defined.

NOTE: The entries in this routine must all be of stencil type. Also, they must all represent couplings to the same variable type.

If the matrix is complex, then values consists of pairs of doubles representing the real and imaginary parts of each complex value.

$_{-}$ 1.5.8 $_{-}$

SIDL_C_INLINE_DECL int32_t

bHYPRE_SStructMatrixView_SetSymmetric (bHYPRE_SStructMatrixView self, int32_t part, int32_t var, int32_t to_var, int32_t symmetric, sidl_BaseInterface *_ex)

Define symmetry properties for the stencil entries in the matrix. The boolean argument symmetric is applied to stencil entries on part part that couple variable var to variable to_var. A value of -1 may be used for part, var, or to_var to specify "all". For example, if part and to_var are set to -1, then the boolean is applied to stencil entries on all parts that couple variable var to all other variables.

By default, matrices are assumed to be nonsymmetric. Significant storage savings can be made if the matrix is symmetric.

1.5.9

SIDL_C_INLINE_DECL int32_t

bHYPRE_SStructMatrixView_Print (bHYPRE_SStructMatrixView self, const char* filename, int32_t all, sidl_BaseInterface *_ex)

Print the matrix to file. This is mainly for debugging purposes.

1.5.10

RMI connector function for the class. (no addref)

_ 1.6 _

SemiStructured Vector View

Names		
1.6.1	struct bHYPRE_SStructVectorViewobject Symbol "bHYPRE	28
1.6.2	extern C bHYPRE_SStructVectorView bHYPRE_SStructVectorViewconnect (const char *,	20
	RMI connector function for the class	28
	SIDL_C_INLINE_DECL int32_t bHYPRE_SStructVectorView_SetGrid (bHYPRE_SStructVectorView self, bHYPRE_SStructGrid grid, sidl_BaseInterface *_ex)	
	Set the vector grid	
1.6.3	SIDL_C_INLINE_DECL int32_t bHYPRE_SStructVectorView_SetValues (bHYPRE_SStructVectorView	
	self, int32_t part, int32_t* index, int32_t dim, int32_t var, double value, sidl_BaseInterface *_ex)	
	Set vector coefficients index by index	29
1.6.4	int32_t bHYPRE_SStructVectorView_SetBoxValues (
	bHYPRE_SStructVectorView self, int32_t part, int32_t* ilower, int32_t* iupper, int32_t dim, int32_t var, double* values, int32_t nvalues, sidl_BaseInterface *_ex)	
	Set vector coefficients a box at a time	29
1.6.5	SIDL_C_INLINE_DECL int32_t	

	$\mathbf{bHYPRE_SStructVectorView_AddToValues} \ (\ \ \mathbf{bHYPRE_SStructVectorView} \\$	
	self, int32_t part,	
	$int32_{t}$ index, $int32_{t}$ dim,	
	int32_t var, double value,	
	sidl_BaseInterface *_ex)	
	Set vector coefficients index by index	29
1.6.6	$\mathrm{int}32$ _{t}	
	$b HYPRE_SStructVectorView_AddToBoxValues~($	
	bHYPRE_SStructVectorView	
	$self, int 32_t part,$	
	$int32_{-}t^{*}$ ilower,	
	$int32_{-}t^{*}$ iupper,	
	$int32_{-}t dim$, $int32_{-}t var$,	
	double* values,	
	int32_t nvalues,	
	sidl_BaseInterface *_ex)	
	Set vector coefficients a box at a time	30
	SIDL_C_INLINE_DECL int32_t	
	bHYPRE_SStructVectorView_Gather (bHYPRE_SStructVectorView self, sidl_BaseInterface *_ex)	
	Gather vector data before calling GetValues	
1.6.7	SIDL_C_INLINE_DECL_int32_t	
1.0.7	bHYPRE_SStructVectorView_GetValues (bHYPRE_SStructVectorView	
	self, int32_t part,	
	int32_t* index, int32_t dim,	
	int32_t var, double* value,	
	sidl_BaseInterface *_ex)	
	Get vector coefficients index by index	30
1.6.8	${ m int}32$ _t	
1.0.0	bHYPRE_SStructVectorView_GetBoxValues (
	bHYPRE_SStructVectorView	
	self, int32_t part,	
	$int32$ _t* ilower,	
	int32_t* iupper, int32_t dim,	
	int32_t var, double* values,	
	int32 _{-t} nvalues,	
	sidl_BaseInterface *_ex)	
	Get vector coefficients a box at a time	31
	SIDL_C_INLINE_DECL int32_t	
	bHYPRE_SStructVectorView_SetComplex (bHYPRE_SStructVectorView	
	self, sidl_BaseInterface *_ex)	
	Set the vector to be complex	
1.6.9	SIDL_C_INLINE_DECL int32_t	
	bHYPRE_SStructVectorView_Print (bHYPRE_SStructVectorView self,	
	const char* filename, int32_t all,	
	sidl_BaseInterface *_ex)	
	Print the vector to file	31
	_ex	
	Cast method for interface and class type conversions	
	· · · · · · · · · · · · · · · · · · ·	
	void^*	

```
bHYPRE_SStructVectorView__cast2 (void* obj, const char* type,
                                                  sidl_BaseInterface *_ex)
                   String cast method for interface and class type conversions
            SIDL_C_INLINE_DECL_void
            bHYPRE_SStructVectorView__exec ( bHYPRE_SStructVectorView self,
                                                  const char* methodName,
                                                  sidl_rmi_Call inArgs,
                                                  sidl_rmi_Return outArgs,
                                                  sidl_BaseInterface *_ex)
                   Select and execute a method by name
            SIDL_C_INLINE_DECL char*
            bHYPRE_SStructVectorView__getURL ( bHYPRE_SStructVectorView self,
                                                     sidl_BaseInterface *_ex)
                   Get the URL of the Implementation of this object (for RMI)
            SIDL_C_INLINE_DECL_void
            bHYPRE_SStructVectorView_raddRef ( bHYPRE_SStructVectorView self,
                                                      sidl_BaseInterface *_ex)
                   On a remote object, addrefs the remote instance
            SIDL_C_INLINE_DECL sidl_bool
            bHYPRE_SStructVectorView__isRemote ( bHYPRE_SStructVectorView
                                                       self, sidl_BaseInterface *_ex)
                   TRUE if this object is remote, false if local
            sidl_bool
            bHYPRE_SStructVectorView__isLocal ( bHYPRE_SStructVectorView self,
                                                    sidl_BaseInterface *_ex)
                   TRUE if this object is remote, false if local
            **_ex
                   Cast method for interface and class type conversions
            **_ex
1.6.10
                   RMI connector function for the class .....
                                                                                             31
```

1.6.1

struct bHYPRE_SStructVectorView__object

Symbol "bHYPRE.SStructVectorView" (version 1.0.0)

1.6.2

extern C bHYPRE_SStructVectorView
bHYPRE_SStructVectorView_connect (const char *, sidl_BaseInterface *_ex)

RMI connector function for the class.(addrefs)

1.6.3

SIDL_C_INLINE_DECL int32_t

bHYPRE_SStructVectorView_SetValues (bHYPRE_SStructVectorView self, int32_t part, int32_t* index, int32_t dim, int32_t var, double value, sidl_BaseInterface *_ex)

Set vector coefficients index by index.

NOTE: Users are required to set values on all processes that own the associated variables. This means that some data will be multiply defined.

If the vector is complex, then value consists of a pair of doubles representing the real and imaginary parts of the complex value.

1.6.4

 $int32_t$

bHYPRE_SStructVectorView_SetBoxValues (bHYPRE_SStructVectorView self, int32_t part, int32_t* ilower, int32_t* iupper, int32_t dim, int32_t var, double* values, int32_t nvalues, sidl_BaseInterface *_ex)

Set vector coefficients a box at a time.

NOTE: Users are required to set values on all processes that own the associated variables. This means that some data will be multiply defined.

If the vector is complex, then values consists of pairs of doubles representing the real and imaginary parts of each complex value.

1.6.5

 $SIDL_C_INLINE_DECL int 32_t$

bHYPRE_SStructVectorView_AddToValues (bHYPRE_SStructVectorView self, int32_t part, int32_t* index, int32_t dim, int32_t var, double value, sidl_BaseInterface *_ex)

Set vector coefficients index by index.

NOTE: Users are required to set values on all processes that own the associated variables. This means that some data will be multiply defined.

If the vector is complex, then value consists of a pair of doubles representing the real and imaginary parts of the complex value.

1.6.6

$int32_t$

bHYPRE_SStructVectorView_AddToBoxValues (

bHYPRE_SStructVectorView self, int32_t part, int32_t* ilower, int32_t* iupper, int32_t dim, int32_t var, double* values, int32_t nvalues, sidl_BaseInterface *_ex)

Set vector coefficients a box at a time.

NOTE: Users are required to set values on all processes that own the associated variables. This means that some data will be multiply defined.

If the vector is complex, then values consists of pairs of doubles representing the real and imaginary parts of each complex value.

1.6.7

SIDL_C_INLINE_DECL int32_t

bHYPRE_SStructVectorView_GetValues (bHYPRE_SStructVectorView self, int32_t part, int32_t* index, int32_t dim, int32_t var, double* value, sidl_BaseInterface *_ex)

Get vector coefficients index by index.

NOTE: Users may only get values on processes that own the associated variables.

If the vector is complex, then value consists of a pair of doubles representing the real and imaginary parts of the complex value.

1.6.8

 $int32_t$

bHYPRE_SStructVectorView_GetBoxValues (bHYPRE_SStructVectorView self, int32_t part, int32_t* ilower, int32_t* iupper, int32_t dim, int32_t var, double* values, int32_t nvalues, sidl_BaseInterface *_ex)

Get vector coefficients a box at a time.

NOTE: Users may only get values on processes that own the associated variables.

If the vector is complex, then values consists of pairs of doubles representing the real and imaginary parts of each complex value.

1.6.9

SIDL_C_INLINE_DECL int32_t bHYPRE_SStructVectorView_Print (bHYPRE_SStructVectorView self, const char* filename, int32_t all, sidl_BaseInterface *_ex)

Print the vector to file. This is mainly for debugging purposes.

_ 1.6.10 __

struct bHYPRE_SStructVectorView__object* bHYPRE_SStructVectorView__connectI const char * url sidl_bool ar struct sidl_BaseInterface__object ** ex

RMI connector function for the class. (no addref)

 $\mathbf{2}$

Operator Interface

Names		
2.1	struct bHYPRE_Operator_object Symbol "bHYPRE	34
2.2	extern C bHYPRE_Operator bHYPRE_Operatorconnect (const char *, sidl_BaseInterface *_ex) RMI connector function for the class	35
2.3	SIDL_C_INLINE_DECL_int32_t bHYPRE_Operator_SetCommunicator (bHYPRE_Operator self,	
	Set the MPI Communicator	35
2.4	SIDL_C_INLINE_DECL_void bHYPRE_Operator_Destroy (bHYPRE_Operator self,	35
	SIDL_C_INLINE_DECL_int32_t bHYPRE_Operator_SetIntParameter (bHYPRE_Operator self,	
	SIDL_C_INLINE_DECL_int32_t bHYPRE_Operator_SetDoubleParameter (bHYPRE_Operator self,	
	SIDL_C_INLINE_DECL_int32_t bHYPRE_Operator_SetStringParameter (bHYPRE_Operator self,	
	Set the string parameter associated with name	
	SIDL_C_INLINE_DECL int32_t bHYPRE_Operator_SetIntArray1Parameter (bHYPRE_Operator self,	
	Set the int 1-D array parameter associated with name	
	SIDL_C_INLINE_DECL_int32_t	

```
bHYPRE_Operator_SetIntArray2Parameter ( bHYPRE_Operator self,
                                               const char* name,
                                              struct sidl_int_array* value,
                                              sidl_BaseInterface *_ex)
      Set the int 2-D array parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_Operator_SetDoubleArray1Parameter ( bHYPRE_Operator self,
                                                   const char* name,
                                                   double* value,
                                                   int32_t nvalues.
                                                   sidl_BaseInterface *_ex)
      Set the double 1-D array parameter associated with name
SIDL_C_INLINE_DECL_int32_t
bHYPRE_Operator_SetDoubleArray2Parameter ( bHYPRE_Operator self,
                                                   const char* name,
                                                   struct sidl_double_array*
                                                   value,
                                                   sidl_BaseInterface *_ex)
      Set the double 2-D array parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_Operator_GetIntValue ( bHYPRE_Operator self,
                                   const char* name, int32_t* value,
                                   sidl_BaseInterface *_ex)
      Set the int parameter associated with name
SIDL_C_INLINE_DECL_int32_t
bHYPRE_Operator_GetDoubleValue ( bHYPRE_Operator self,
                                       const char* name, double* value,
                                       sidl_BaseInterface *_ex)
      Get the double parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_Operator_Setup ( bHYPRE_Operator self, bHYPRE_Vector b,
                            bHYPRE_Vector x, sidl_BaseInterface *_ex)
      (Optional) Do any preprocessing that may be necessary in order to execute
      Apply
SIDL_C_INLINE_DECL int32_t
bHYPRE_Operator_Apply (bHYPRE_Operator self, bHYPRE_Vector b,
                            bHYPRE_Vector* x, sidl_BaseInterface *_ex)
      Apply the operator to b, returning x
SIDL_C_INLINE_DECL int32_t
bHYPRE_Operator_ApplyAdjoint ( bHYPRE_Operator self,
                                    bHYPRE_Vector b,
                                    bHYPRE_Vector* x,
                                    sidl_BaseInterface *_ex)
      Apply the adjoint of the operator to b, returning x
_ex
      Cast method for interface and class type conversions
void*
```

```
bHYPRE_Operator_cast2 (void* obj, const char* type,
                                        sidl_BaseInterface *_ex)
                   String cast method for interface and class type conversions
            SIDL_C_INLINE_DECL_void
            bHYPRE_Operator_exec ( bHYPRE_Operator self,
                                        const char* methodName, sidl_rmi_Call inArgs,
                                       sidl_rmi_Return outArgs, sidl_BaseInterface *_ex)
                   Select and execute a method by name
            SIDL_C_INLINE_DECL_char*
            bHYPRE_Operator__getURL ( bHYPRE_Operator self,
                                           sidl_BaseInterface *_ex)
                   Get the URL of the Implementation of this object (for RMI)
            SIDL_C_INLINE_DECL void
            bHYPRE_Operator_raddRef ( bHYPRE_Operator self,
                                           sidl_BaseInterface *_ex)
                   On a remote object, addrefs the remote instance
            SIDL_C_INLINE_DECL sidl_bool
            bHYPRE_Operator__isRemote ( bHYPRE_Operator self,
                                             sidl_BaseInterface *_ex)
                   TRUE if this object is remote, false if local
            sidl_bool
            bHYPRE_Operator__isLocal ( bHYPRE_Operator self,
                                          sidl_BaseInterface *_ex)
                   TRUE if this object is remote, false if local
            **_ex
                   Cast method for interface and class type conversions
2.5
            **_ex
                   RMI connector function for the class .....
                                                                                             35
```

2.1

struct bHYPRE_Operator_object

Symbol "bHYPRE.Operator" (version 1.0.0)

An Operator is anything that maps one Vector to another. The terms Setup and Apply are reserved for Operators. The implementation is allowed to assume that supplied parameter arrays will not be destroyed.

_ 2.2 .

extern C bHYPRE_Operator **bHYPRE_Operator__connect** (const char *, sidl_BaseInterface *_ex)

RMI connector function for the class.(addrefs)

_ 2.3 _

SIDL_C_INLINE_DECL int32_t
bHYPRE_Operator_SetCommunicator (bHYPRE_Operator self,
bHYPRE_MPICommunicator mpi_comm, sidl_BaseInterface *_ex)

Set the MPI Communicator. DEPRECATED, use Create:

2.4

 $\begin{array}{l} {\rm SIDL_C_INLINE_DECL~void} \\ {\bf bHYPRE_Operator_Destroy} \; (\ \, {\rm bHYPRE_Operator~self, ~sidl_BaseInterface~*_ex}) \end{array}$

The Destroy function doesn't necessarily destroy anything. It is just another name for deleteRef. Thus it decrements the object's reference count. The Babel memory management system will destroy the object if the reference count goes to zero.

_ 2.5 _

RMI connector function for the class. (no addref)

. 3

Vector Interface

Names		
3.1	struct bHYPRE_Vector_object Symbol "bHYPRE	37
3.2	extern C bHYPRE_Vector bHYPRE_Vector_connect (const char *, sidl_BaseInterface *_ex) RMI connector function for the class	37
	SIDL_C_INLINE_DECL int32_t bHYPRE_Vector_Clear (bHYPRE_Vector self, sidl_BaseInterface *_ex) Set self to 0	
	SIDL_C_INLINE_DECL int32_t bHYPRE_Vector_Copy (bHYPRE_Vector self, bHYPRE_Vector x,	
3.3	SIDL_C_INLINE_DECL int32_t bHYPRE_Vector_Clone (bHYPRE_Vector self, bHYPRE_Vector* x,	37
	SIDL_C_INLINE_DECL int32_t bHYPRE_Vector_Scale (bHYPRE_Vector self, double a,	
	SIDL_C_INLINE_DECL int32_t bHYPRE_Vector_Dot (bHYPRE_Vector self, bHYPRE_Vector x,	
	SIDL_C_INLINE_DECL int32_t bHYPRE_Vector_Axpy (bHYPRE_Vector self, double a, bHYPRE_Vector x, sidl_BaseInterface *_ex) Add ax to self	
	_ex Cast method for interface and class type conversions	
	void* bHYPRE_Vectorcast2 (void* obj, const char* type, sidl_BaseInterface *_ex) String cast method for interface and class type conversions	
	SIDL_C_INLINE_DECL void bHYPRE_Vector_exec (bHYPRE_Vector self, const char* methodName, sidl_rmi_Call inArgs, sidl_rmi_Return outArgs, sidl_BaseInterface *_ex) Select and execute a method by name	
	SIDL C INLINE DECL, char*	

bHYPRE_Vector_getURL (bHYPRE_Vector self, sidl_BaseInterface *_ex) Get the URL of the Implementation of this object (for RMI) SIDL_C_INLINE_DECL void bHYPRE_Vector_raddRef (bHYPRE_Vector self, sidl_BaseInterface *_ex) On a remote object, addrefs the remote instance SIDL_C_INLINE_DECL sidl_bool bHYPRE_Vector_isRemote (bHYPRE_Vector self, sidl_BaseInterface *_ex) TRUE if this object is remote, false if local $sidl_bool$ bHYPRE_Vector_isLocal (bHYPRE_Vector self, sidl_BaseInterface *_ex) TRUE if this object is remote, false if local $**_ex$ Cast method for interface and class type conversions **_ex 3.4 RMI connector function for the class 38

3.1

struct bHYPRE_Vector_object

Symbol "bHYPRE.Vector" (version 1.0.0)

3.2 .

extern C bHYPRE_Vector bHYPRE_Vector_connect (const char *, sidl_BaseInterface *_ex)

RMI connector function for the class.(addrefs)

_ 3.3 _

SIDL_C_INLINE_DECL int32_t bHYPRE_Vector_Clone (bHYPRE_Vector self, bHYPRE_Vector* x, sidl_BaseInterface *_ex)

Create an x compatible with self. The new vector's data is not specified.

NOTE: When this method is used in an inherited class, the cloned Vector object can be cast to an object with the inherited class type.

3 4

RMI connector function for the class. (no addref)

_	4
	-

Matrices and Vectors

Names		
4.1	IJParCSR Matrix	
		39
4.2	IJParCSR Vector	Ε.
		5(
4.3	Struct Matrix	۲,
		56
4.4	Struct Vector	62
		02
4.5	SemiStructured Matrix	67
4.0	0 10 1 17	01
4.6	SemiStructured Vector	77
4.7	SemiStructured ParCSR Matrix	•
4.7	Semistructured Farcsa Matrix	8
4.8	SemiStructured ParCSR Vector	
1.0	Semistructured Lareste vector	96

4.1 .

IJParCSR Matrix

Names

Constructor function for the class

 $bHYPRE_IJParCSRMatrix$

bHYPRE_IJParCSRMatrix__createRemote (const char * url,

sidl_BaseInterface *_ex)

RMI constructor function for the class

 $b HYPRE_IJParCSRMatrix$

bHYPRE_IJParCSRMatrix_wrapObj (void * data, sidl_BaseInterface *_ex)

Wraps up the private data struct pointer (struct
bHYPRE_IJParCSRMatrix_data) passed in rather than running the

constructor

4.1.2 bHYPRE_IJParCSRMatrix

	bHYPRE_IJParCSRMatrix	
	bHYPRE_IJParCSRMatrix_Create (bHYPRE_MPICommunicator	
	${ m mpi_comm}, { m int} 32_{ m t} \; { m ilower},$	
	int32_t iupper, int32_t jlower,	
	int32_t jupper, sidl_BaseInterface *_ex)	
	Method: Create[]	
	bHYPRE_IJParCSRMatrix	
	${\bf bHYPRE_IJParCSRMatrix_GenerateLaplacian} \ ($	
	bHYPRE_MPICommunicator	
	mpi_comm , $int32_t$ nx ,	
	$int32_{-t} ny$, $int32_{-t} nz$,	
	$int32_{-t} Px$, $int32_{-t} Py$,	
	$int32_{-t}$ Pz, $int32_{-t}$ p,	
	$int32_{-}t q$, $int32_{-}t r$,	
	double* values,	
	int32_t nvalues,	
	int32_t discretization, sidl_BaseInterface *_ex)	
	Method: GenerateLaplacian[]	
4.1.3	int32_t bHYPRE_IJParCSRMatrix_SetDiagOffdSizes (bHYPRE_IJParCSRMatrix self, int32_t* diag_sizes, int32_t* offdiag_sizes, int32_t local_nrows, sidl_BaseInterface *_ex)	
	(Optional) Set the max number of nonzeros to expect in each row of the diagonal and off-diagonal blocks	46
4.1.4	SIDL_C_INLINE_DECL int32_t	
	bHYPRE_IJParCSRMatrix_SetLocalRange (bHYPRE_IJParCSRMatrix	
	$self, int 32_t ilower,$	
	int32_t iupper, int32_t jlower,	
	${ m int}32$ _t jupper,	
	sidl_BaseInterface *_ex)	
	Set the local range for a matrix object	46
4.1.5	$\mathrm{int}32$ _t	
	bHYPRE_IJParCSRMatrix_SetValues (bHYPRE_IJParCSRMatrix self, int32_t nrows, int32_t* ncols, int32_t* rows, int32_t* cols, double* values, int32_t nnonzeros,	
	sidl_BaseInterface *_ex) Sets values for nrows of the matrix	46
		40
4.1.6	$\mathrm{int}32$ _{t}	

	bHYPRE_IJParCSRMatrix_AddToValues (bHYPRE_IJParCSRMatrix	
	self, int32_t nrows, int32_t* ncols, int32_t* rows,	
	int32_t* cols, double* values,	
	int32_t nnonzeros,	
	sidl_BaseInterface *_ex)	
	Adds to values for nrows of the matrix	47
	SIDL_C_INLINE_DECL int32_t	
	bHYPRE_IJParCSRMatrix_GetLocalRange (bHYPRE_IJParCSRMatrix	
	self, int 32_{-} t* ilower,	
	int32_t* iupper,	
	$int32_{-}t^{*}$ jlower,	
	$int32_{-}t^{*}$ jupper,	
	sidl_BaseInterface *_ex)	
	Gets range of rows owned by this processor and range of column partitioning for this processor	
	$int32_t$	
	bHYPRE_IJParCSRMatrix_GetRowCounts (bHYPRE_IJParCSRMatrix self, int32_t nrows,	
	$int32_{-}t^* rows$, $int32_{-}t^* ncols$,	
	sidl_BaseInterface *_ex)	
	Gets number of nonzeros elements for nrows rows specified in rows and	
	returns them in ncols, which needs to be allocated by the user	
4.1.7	$int32_t$	
	bHYPRE_IJParCSRMatrix_GetValues (bHYPRE_IJParCSRMatrix self,	
	int32_t nrows, int32_t* ncols,	
	int32_t* rows, int32_t* cols,	
	double* values, int32_t nnonzeros, sidl_BaseInterface *_ex)	
	Gets values for nrows rows or partial rows of the matrix	47
4.1.0	-	
4.1.8	SIDL_C_INLINE_DECL int32_t bHYPRE_IJParCSRMatrix_SetRowSizes (bHYPRE_IJParCSRMatrix self,	
	int32_t* sizes, int32_t nrows,	
	sidl_BaseInterface *_ex)	
	(Optional) Set the max number of nonzeros to expect in each row	47
4.1.9	SIDL_C_INLINE_DECL int32_t	
4.1.9	bHYPRE_IJParCSRMatrix_Print (bHYPRE_IJParCSRMatrix self,	
	const char* filename,	
	sidl_BaseInterface *_ex)	
	Print the matrix to file	48
4.1.10	SIDL_C_INLINE_DECL int32_t	
	$\mathbf{bHYPRE_IJParCSRMatrix_Read} \ (\ \ \mathbf{bHYPRE_IJParCSRMatrix} \ \mathbf{self},$	
	const char* filename,	
	bHYPRE_MPICommunicator comm,	
	sidl_BaseInterface *_ex)	48
4 1 1 1	Read the matrix from file	48
4.1.11	SIDL_C_INLINE_DECL int32_t	

	${\bf bHYPRE_IJParCSRMatrix_SetCommunicator} \ ($	
	bHYPRE_IJParCSRMatrix	
	$\operatorname{self},$	
	bHYPRE_MPICommunicator	
	mpi_comm,	
	sidl_BaseInterface *_ex) Set the MPI Communicator	48
		40
4.1.12	SIDL_C_INLINE_DECL void	
	bHYPRE_IJParCSRMatrix_Destroy (bHYPRE_IJParCSRMatrix self, sidl_BaseInterface *_ex)	
	The Destroy function doesn't necessarily destroy anything	49
		10
	SIDL_C_INLINE_DECL int32_t bHYPRE_IJParCSRMatrix_Initialize (bHYPRE_IJParCSRMatrix self,	
	sidl_BaseInterface *_ex)	
	Prepare an object for setting coefficient values, whether for the first time or	
	subsequently	
4.1.13	SIDL_C_INLINE_DECL int32_t	
1.1.10	bHYPRE_IJParCSRMatrix_Assemble (bHYPRE_IJParCSRMatrix self,	
	sidl_BaseInterface *_ex)	
	Finalize the construction of an object before using, either for the first time	
	or on subsequent uses	49
	SIDL_C_INLINE_DECL int32_t	
	$\mathbf{bHYPRE_IJParCSRMatrix_SetIntParameter} \ (\ \mathbf{bHYPRE_IJParCSRMatrix}$	
	self, const char* name,	
	int32_t value,	
	sidl_BaseInterface *_ex) Set the int parameter associated with name	
	SIDL_C_INLINE_DECL int32_t	
	bHYPRE_IJParCSRMatrix_SetDoubleParameter (bHYPRE_IJParCSRMatrix	
	self, const char* name,	
	double value,	
	sidl_BaseInterface *_ex)	
	Set the double parameter associated with name	
	SIDL_C_INLINE_DECL int32_t	
	bHYPRE_IJParCSRMatrix_SetStringParameter (
	bHYPRE_IJParCSRMatrix	
	self, const char* name,	
	const char* value,	
	sidl_BaseInterface *_ex)	
	Set the string parameter associated with name	
	SIDL_C_INLINE_DECL int32_t	

```
bHYPRE_IJParCSRMatrix_SetIntArray1Parameter (
                                                       bHYPRE_IJParCSRMatrix
                                                       const char* name,
                                                       int32_t* value,
                                                       int32_t nvalues,
                                                       sidl_BaseInterface
                                                       *_ex)
      Set the int 1-D array parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_IJParCSRMatrix_SetIntArray2Parameter (
                                                       bHYPRE_IJParCSRMatrix
                                                       self,
                                                       const char* name,
                                                       struct
                                                       sidl_int_array*
                                                       value,
                                                       sidl_BaseInterface
                                                       *_ex)
      Set the int 2-D array parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_IJParCSRMatrix_SetDoubleArray1Parameter (
                                                           bHYPRE_IJParCSRMatrix
                                                           self, const
                                                           char* name,
                                                           double* value,
                                                           int32_t nvalues.
                                                           sidl\_BaseInterface
                                                            *_ex)
      Set the double 1-D array parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE\_IJParCSRMatrix\_SetDoubleArray2Parameter\ (
                                                           bHYPRE_IJParCSRMatrix
                                                           self, const
                                                           char* name,
                                                           struct
                                                           sidl_double_array*
                                                           value,
                                                           sidl_BaseInterface
                                                            *_ex)
      Set the double 2-D array parameter associated with name
SIDL_C_INLINE_DECL_int32_t
bHYPRE_IJParCSRMatrix_GetIntValue ( bHYPRE_IJParCSRMatrix self,
                                           const char* name.
                                           int32_t* value,
                                           sidl_BaseInterface *_ex)
      Set the int parameter associated with name
```

SIDL_C_INLINE_DECL int32_t

bHYPRE_IJParCSRMatrix_GetDoubleValue (bHYPRE_IJParCSRMatrix

```
self, const char* name,
                                                          double* value.
                                                          sidl_BaseInterface *_ex)
                  Get the double parameter associated with name
            SIDL_C_INLINE_DECL int32_t
            bHYPRE_IJParCSRMatrix_Setup ( bHYPRE_IJParCSRMatrix self,
                                               bHYPRE_Vector b, bHYPRE_Vector x,
                                               sidl_BaseInterface *_ex)
                  (Optional) Do any preprocessing that may be necessary in order to execute
                  Apply
            SIDL_C_INLINE_DECL_int32_t
            bHYPRE_IJParCSRMatrix_Apply ( bHYPRE_IJParCSRMatrix self,
                                                bHYPRE_Vector b.
                                                bHYPRE_Vector* x,
                                                sidl_BaseInterface *_ex)
                  Apply the operator to b, returning x
            SIDL_C_INLINE_DECL int32_t
            bHYPRE_IJParCSRMatrix_ApplyAdjoint ( bHYPRE_IJParCSRMatrix
                                                       self, bHYPRE_Vector b,
                                                       bHYPRE_Vector* x,
                                                       sidl_BaseInterface *_ex)
                  Apply the adjoint of the operator to b, returning x
            SIDL_C_INLINE_DECL int32_t
4.1.14
            bHYPRE_IJParCSRMatrix_GetRow ( bHYPRE_IJParCSRMatrix self,
                                                  int32_t row, int32_t* size,
                                                  struct sidl_int_array** col_ind,
                                                  struct sidl_double_array** values,
                                                  sidl_BaseInterface *_ex)
                  The GetRow method will allocate space for its two output arrays on the first
                  call .....
                                                                                           49
            _{-}ex
                  Cast method for interface and class type conversions
            *void
            bHYPRE_IJParCSRMatrix_cast2 (void* obj, const char* type,
                                               sidl_BaseInterface *_ex)
                  String cast method for interface and class type conversions
            SIDL_C_INLINE_DECL void
            bHYPRE_IJParCSRMatrix_exec ( bHYPRE_IJParCSRMatrix self,
                                               const char* methodName,
                                               sidl_rmi_Call inArgs,
                                               sidl_rmi_Return outArgs,
                                               sidl_BaseInterface *_ex)
                  Select and execute a method by name
            SIDL_C_INLINE_DECL char*
            bHYPRE_IJParCSRMatrix__getURL ( bHYPRE_IJParCSRMatrix self,
                                                  sidl_BaseInterface *_ex)
                  Get the URL of the Implementation of this object (for RMI)
            SIDL_C_INLINE_DECL_void
```

bHYPRE_IJParCSRMatrix_raddRef (bHYPRE_IJParCSRMatrix self, sidl_BaseInterface *_ex)

On a remote object, addrefs the remote instance

SIDL_C_INLINE_DECL_sidl_bool

bHYPRE_IJParCSRMatrix_isRemote (bHYPRE_IJParCSRMatrix self, sidl_BaseInterface *_ex)

TRUE if this object is remote, false if local

 $sidl_bool$

bHYPRE_IJParCSRMatrix__isLocal (bHYPRE_IJParCSRMatrix self, sidl_BaseInterface *_ex)

TRUE if this object is remote, false if local

 $**_{-}ex$

Cast method for interface and class type conversions

4.1.15 **_**ex**

4.1.1

struct bHYPRE_IJParCSRMatrix_object

Symbol "bHYPRE.IJParCSRMatrix" (version 1.0.0)

The IJParCSR matrix class.

Objects of this type can be cast to IJMatrixView, Operator, or CoefficientAccess objects using the <code>__cast</code> methods.

4.1.2

bHYPRE_IJParCSRMatrix bHYPRE_IJParCSRMatrix_connect (const char *, sidl_BaseInterface *_ex)

RMI connector function for the class.(addrefs)

4.1.3

 $int32_t$

bHYPRE_IJParCSRMatrix_SetDiagOffdSizes (bHYPRE_IJParCSRMatrix self, int32_t* diag_sizes, int32_t* offdiag_sizes, int32_t local_nrows, sidl_BaseInterface *_ex)

(Optional) Set the max number of nonzeros to expect in each row of the diagonal and off-diagonal blocks. The diagonal block is the submatrix whose column numbers correspond to rows owned by this process, and the off-diagonal block is everything else. The arrays diag_sizes and offdiag_sizes contain estimated sizes for each row of the diagonal and off-diagonal blocks, respectively. This routine can significantly improve the efficiency of matrix construction, and should always be utilized if possible.

Not collective.

4.1.4

SIDL_C_INLINE_DECL int32_t bHYPRE_IJParCSRMatrix_SetLocalRange (bHYPRE_IJParCSRMatrix self, int32_t ilower, int32_t ilower, int32_t jlower, int32_t jupper, sidl_BaseInterface *_ex)

Set the local range for a matrix object. Each process owns some unique consecutive range of rows, indicated by the global row indices ilower and iupper. The row data is required to be such that the value of ilower on any process p be exactly one more than the value of iupper on process p-1. Note that the first row of the global matrix may start with any integer value. In particular, one may use zero- or one-based indexing.

For square matrices, jlower and jupper typically should match ilower and iupper, respectively. For rectangular matrices, jlower and jupper should define a partitioning of the columns. This partitioning must be used for any vector v that will be used in matrix-vector products with the rectangular matrix. The matrix data structure may use jlower and jupper to store the diagonal blocks (rectangular in general) of the matrix separately from the rest of the matrix.

Collective.

$_{-}$ 4.1.5 $_{-}$

 $int32_t$

bHYPRE_IJParCSRMatrix_SetValues (bHYPRE_IJParCSRMatrix self, int32_t nrows, int32_t* ncols, int32_t* rows, int32_t* cols, double* values, int32_t nnonzeros, sidl_BaseInterface *_ex)

Sets values for nrows of the matrix. The arrays ncols and rows are of dimension nrows and contain the number of columns in each row and the row indices, respectively. The array cols contains the column indices for each of the rows, and is ordered by rows. The data in the values array corresponds directly to the column entries in cols. The last argument is the size of the cols and values arrays, i.e. the total number of nonzeros being provided, i.e. the sum of all values in ncols. This functin erases any previous values at the specified locations and replaces them with new ones, or, if there was no value there before, inserts a new one.

Not collective.

4.1.6

 $int32_t$

bHYPRE_IJParCSRMatrix_AddToValues (bHYPRE_IJParCSRMatrix self, int32_t nrows, int32_t* ncols, int32_t* cols, double* values, int32_t nnonzeros, sidl_BaseInterface *_ex)

Adds to values for **nrows** of the matrix. Usage details are analogous to **SetValues**. Adds to any previous values at the specified locations, or, if there was no value there before, inserts a new one.

Not collective.

4.1.7

 $int32_t$

bHYPRE_IJParCSRMatrix_GetValues (bHYPRE_IJParCSRMatrix self, int32_t nrows, int32_t* ncols, int32_t* rows, int32_t* cols, double* values, int32_t nnonzeros, sidl_BaseInterface *_ex)

Gets values for nrows rows or partial rows of the matrix. Usage details are analogous to SetValues.

4.1.8

SIDL_C_INLINE_DECL int32_t bHYPRE_IJParCSRMatrix_SetRowSizes (bHYPRE_IJParCSRMatrix self, int32_t* sizes, int32_t nrows, sidl_BaseInterface *_ex)

(Optional) Set the max number of nonzeros to expect in each row. The array sizes contains estimated sizes for each row on this process. The integer nrows is the number of rows in the local matrix. This call can significantly improve the efficiency of matrix construction, and should always be utilized if possible.

Not collective.

4.1.9

SIDL_C_INLINE_DECL int32_t **bHYPRE_IJParCSRMatrix_Print** (bHYPRE_IJParCSRMatrix self, const char* filename, sidl_BaseInterface *_ex)

Print the matrix to file. This is mainly for debugging purposes.

__ 4.1.10 _____

SIDL_C_INLINE_DECL int32_t bHYPRE_IJParCSRMatrix_Read (bHYPRE_IJParCSRMatrix self, const char* filename, bHYPRE_MPICommunicator comm, sidl_BaseInterface *_ex)

Read the matrix from file. This is mainly for debugging purposes.

4.1.11

SIDL_C_INLINE_DECL int32_t bHYPRE_IJParCSRMatrix_SetCommunicator (bHYPRE_IJParCSRMatrix self, bHYPRE_MPICommunicator mpi_comm, sidl_BaseInterface *_ex)

Set the MPI Communicator. DEPRECATED, Use Create()

4.1.12

SIDL_C_INLINE_DECL_void bHYPRE_IJParCSRMatrix_Destroy (bHYPRE_IJParCSRMatrix self, sidl_BaseInterface *_ex)

The Destroy function doesn't necessarily destroy anything. It is just another name for deleteRef. Thus it decrements the object's reference count. The Babel memory management system will destroy the object if the reference count goes to zero.

4.1.13

SIDL_C_INLINE_DECL int32_t bHYPRE_IJParCSRMatrix_Assemble (bHYPRE_IJParCSRMatrix self, sidl_BaseInterface *_ex)

Finalize the construction of an object before using, either for the first time or on subsequent uses. Initialize and Assemble always appear in a matched set, with Initialize preceding Assemble. Values can only be set in between a call to Initialize and Assemble.

4.1.14

SIDL_C_INLINE_DECL int32_t bHYPRE_IJParCSRMatrix_GetRow (bHYPRE_IJParCSRMatrix self, int32_t row, int32_t* size, struct sidl_int__array** col_ind, struct sidl_double_array** values, sidl_BaseInterface *_ex)

The GetRow method will allocate space for its two output arrays on the first call. The space will be reused on subsequent calls. Thus the user must not delete them, yet must not depend on the data from GetRow to persist beyond the next GetRow call.

4.1.15

struct bHYPRE_IJParCSRMatrix__object* bHYPRE_IJParCSRMatrix__connectI const char * url sidl_bool ar struct sidl_BaseInterface__object **_ex

 RMI connector function for the class. (no addref)

__ 4.2 _

IJParCSR Vector

Names		
4.2.1	struct bHYPRE_IJParCSRVectorobject Symbol "bHYPRE	53
	_ex Constructor function for the class	
	bHYPRE_IJParCSRVector bHYPRE_IJParCSRVectorcreateRemote (const char * url, sidl_BaseInterface *_ex)	
	RMI constructor function for the class	
	bHYPRE_IJParCSRVector bHYPRE_IJParCSRVectorwrapObj (void * data, sidl_BaseInterface *_ex) Wraps up the private data struct pointer (struct bHYPRE_IJParCSRVectordata) passed in rather than running the constructor	
4.2.2	bHYPRE_IJParCSRVector bHYPRE_IJParCSRVectorconnect (const char *, sidl_BaseInterface *_ex) RMI connector function for the class	53
	bHYPRE_IJParCSRVector bHYPRE_IJParCSRVector_Create (bHYPRE_MPICommunicator mpi_comm, int32_t jlower, int32_t jupper, sidl_BaseInterface *_ex) Method: Create[]	
4.2.3	SIDL_C_INLINE_DECL int32_t bHYPRE_IJParCSRVector_SetLocalRange (bHYPRE_IJParCSRVector self, int32_t jlower, int32_t jupper, sidl_BaseInterface *_ex)	
	Set the local range for a vector object	53
4.2.4	int32_t bHYPRE_IJParCSRVector_SetValues (bHYPRE_IJParCSRVector self,	
	Sets values in vector	54
4.2.5	$\mathrm{int}32_\mathrm{t}$	

	bHYPRE_IJParCSRVector_AddToValues (bHYPRE_IJParCSRVector self, int32_t nvalues, int32_t* indices, double* values, sidl_BaseInterface *_ex)
	Adds to values in vector
	SIDL_C_INLINE_DECL int32_t bHYPRE_IJParCSRVector_GetLocalRange (bHYPRE_IJParCSRVector
	self, int32_t* jlower, int32_t* jupper, sidl_BaseInterface *_ex) Returns range of the part of the vector owned by this processor
4.2.6	int32_t bHYPRE_IJParCSRVector_GetValues (bHYPRE_IJParCSRVector self,
	Gets values in vector
4.2.7	SIDL_C_INLINE_DECL int32_t bHYPRE_IJParCSRVector_Print (bHYPRE_IJParCSRVector self,
	Print the vector to file
4.2.8	SIDL_C_INLINE_DECL int32_t bHYPRE_IJParCSRVector_Read (bHYPRE_IJParCSRVector self,
4.2.9	SIDL_C_INLINE_DECL int32_t bHYPRE_IJParCSRVector_SetCommunicator (bHYPRE_IJParCSRVector self, bHYPRE_MPICommunicator mpi_comm, sidl_BaseInterface *_ex)
	Set the MPI Communicator
4.2.10	SIDL_C_INLINE_DECL void bHYPRE_IJParCSRVector_Destroy (bHYPRE_IJParCSRVector self, sidl_BaseInterface *_ex)
	The Destroy function doesn't necessarily destroy anything
	SIDL_C_INLINE_DECL int32_t bHYPRE_IJParCSRVector_Initialize (bHYPRE_IJParCSRVector self, sidl_BaseInterface *_ex)
	Prepare an object for setting coefficient values, whether for the first time or subsequently
4.2.11	SIDL_C_INLINE_DECL int32_t bHYPRE_IJParCSRVector_Assemble (bHYPRE_IJParCSRVector self, sidl_BaseInterface *_ex)
	Finalize the construction of an object before using, either for the first time or on subsequent uses
	SIDL_C_INLINE_DECL_int32_t

```
bHYPRE_IJParCSRVector_Clear ( bHYPRE_IJParCSRVector self,
                                               sidl_BaseInterface *_ex)
                  Set \ {\tt self} \ to \ \theta
            SIDL_C_INLINE_DECL int32_t
            bHYPRE_IJParCSRVector_Copy ( bHYPRE_IJParCSRVector self,
                                               bHYPRE_Vector x,
                                               sidl_BaseInterface *_ex)
                   Copy data from x into self
4.2.12
            SIDL_C_INLINE_DECL int32_t
            bHYPRE_IJParCSRVector_Clone ( bHYPRE_IJParCSRVector self,
                                               bHYPRE_Vector* x,
                                               sidl_BaseInterface *_ex)
                   Create an x compatible with self .....
                                                                                            56
            SIDL_C_INLINE_DECL int32_t
            bHYPRE_IJParCSRVector_Scale ( bHYPRE_IJParCSRVector self,
                                              double a, sidl_BaseInterface *_ex)
                  Scale  self by a
            SIDL_C_INLINE_DECL int32_t
            bHYPRE_IJParCSRVector_Dot ( bHYPRE_IJParCSRVector self,
                                             bHYPRE_Vector x, double* d,
                                             sidl_BaseInterface *_ex)
                   Compute d, the inner-product of self and x
            SIDL_C_INLINE_DECL int32_t
            bHYPRE_IJParCSRVector_Axpy ( bHYPRE_IJParCSRVector self,
                                               double a, bHYPRE_Vector x,
                                               sidl_BaseInterface *_ex)
                   Add ax to self
            _ex
                   Cast method for interface and class type conversions
            void*
            bHYPRE_IJParCSRVector__cast2 (void* obj, const char* type,
                                               sidl_BaseInterface *_ex)
                  String cast method for interface and class type conversions
            SIDL_C_INLINE_DECL void
            bHYPRE_IJParCSRVector_exec ( bHYPRE_IJParCSRVector self,
                                               const char* methodName,
                                              sidl_rmi_Call inArgs,
                                              sidl_rmi_Return outArgs,
                                              sidl_BaseInterface *_ex)
                  Select and execute a method by name
            SIDL_C_INLINE_DECL char*
            bHYPRE_IJParCSRVector__getURL ( bHYPRE_IJParCSRVector self,
                                                  sidl_BaseInterface *_ex)
                   Get the URL of the Implementation of this object (for RMI)
            SIDL_C_INLINE_DECL_void
            bHYPRE_IJParCSRVector__raddRef ( bHYPRE_IJParCSRVector self,
                                                  sidl_BaseInterface *_ex)
                   On a remote object, addrefs the remote instance
            SIDL_C_INLINE_DECL_sidl_bool
```

 $\label{eq:bhypre_ij} \begin{tabular}{ll} \mathbf{b} \mathbf{HYPRE_IJParCSRVector_isRemote} & \mathbf{b} \mathbf{HYPRE_IJParCSRVector_self}, \\ \mathbf{sidl_BaseInterface} & \mathbf{*_ex}) \\ \end{tabular}$

TRUE if this object is remote, false if local

sidl_bool

bHYPRE_IJParCSRVector__isLocal (bHYPRE_IJParCSRVector self, sidl_BaseInterface *_ex)

TRUE if this object is remote, false if local

 $**_ex$

Cast method for interface and class type conversions

4.2.13 **_ex

RMI connector function for the class

4.2.1

struct bHYPRE_IJParCSRVector_object

Symbol "bHYPRE.IJParCSRVector" (version 1.0.0)

The IJParCSR vector class.

Objects of this type can be cast to IJVectorView or Vector objects using the __cast methods.

4.2.2

bHYPRE_IJParCSRVector bHYPRE_IJParCSRVector__connect (const char *, sidl_BaseInterface *_ex)

RMI connector function for the class.(addrefs)

 $_$ 4.2.3 $_$

SIDL_C_INLINE_DECL int32_t bHYPRE_IJParCSRVector_SetLocalRange (bHYPRE_IJParCSRVector self, int32_t jlower, int32_t jupper, sidl_BaseInterface *_ex)

Set the local range for a vector object. Each process owns some unique consecutive range of vector unknowns, indicated by the global indices <code>jlower</code> and <code>jupper</code>. The data is required to be such that the value of <code>jlower</code>

56

on any process p be exactly one more than the value of jupper on process p-1. Note that the first index of the global vector may start with any integer value. In particular, one may use zero- or one-based indexing.

Collective.

 $_{-}$ 4.2.4 $_{-}$

bHYPRE_IJParCSRVector_SetValues (bHYPRE_IJParCSRVector self, int32_t nvalues, int32_t* indices, double* values, sidl_BaseInterface *_ex)

Sets values in vector. The arrays values and indices are of dimension nvalues and contain the vector values to be set and the corresponding global vector indices, respectively. Erases any previous values at the specified locations and replaces them with new ones.

Not collective.

4.2.5

 $int32_t$

 $bHYPRE_IJParCSRVector_AddToValues (bHYPRE_IJParCSRVector self, int 32_t nvalues, int 32_t^* indices, double* values, sidl_BaseInterface *_ex)$

Adds to values in vector. Usage details are analogous to SetValues.

Not collective.

4.2.6

 $int32_t$

bHYPRE_IJParCSRVector_GetValues (bHYPRE_IJParCSRVector self, int32_t nvalues, int32_t* indices, double* values, sidl_BaseInterface *_ex)

Gets values in vector. Usage details are analogous to SetValues.

Not collective.

4.2.7

SIDL_C_INLINE_DECL int32_t bHYPRE_IJParCSRVector_Print (bHYPRE_IJParCSRVector self, const char* filename, sidl_BaseInterface *_ex)

Print the vector to file. This is mainly for debugging purposes.

4.2.8

SIDL_C_INLINE_DECL int32_t bHYPRE_IJParCSRVector_Read (bHYPRE_IJParCSRVector self, const char* filename, bHYPRE_MPICommunicator comm, sidl_BaseInterface *_ex)

Read the vector from file. This is mainly for debugging purposes.

_ 4.2.9 _

SIDL_C_INLINE_DECL int32_t bHYPRE_IJParCSRVector_SetCommunicator (bHYPRE_IJParCSRVector self, bHYPRE_MPICommunicator mpi_comm, sidl_BaseInterface *_ex)

Set the MPI Communicator. DEPRECATED, Use Create()

_ 4.2.10 ____

SIDL_C_INLINE_DECL void bHYPRE_IJParCSRVector_Destroy (bHYPRE_IJParCSRVector self, sidl_BaseInterface *_ex)

The Destroy function doesn't necessarily destroy anything. It is just another name for deleteRef. Thus it decrements the object's reference count. The Babel memory management system will destroy the object if the reference count goes to zero.

4.2.11

SIDL_C_INLINE_DECL int32_t bHYPRE_IJParCSRVector_Assemble (bHYPRE_IJParCSRVector self, sidl_BaseInterface *_ex)

Finalize the construction of an object before using, either for the first time or on subsequent uses. Initialize and Assemble always appear in a matched set, with Initialize preceding Assemble. Values can only be set in between a call to Initialize and Assemble.

__ 4.2.12 _____

SIDL_C_INLINE_DECL int32_t bHYPRE_IJParCSRVector_Clone (bHYPRE_IJParCSRVector self, bHYPRE_Vector* x, sidl_BaseInterface *_ex)

Create an x compatible with self. The new vector's data is not specified.

NOTE: When this method is used in an inherited class, the cloned Vector object can be cast to an object with the inherited class type.

_ 4.2.13 ___

RMI connector function for the class. (no addref)

4.3

Struct Matrix

Names

_ex

Constructor function for the class bHYPRE_StructMatrix bHYPRE_StructMatrix_createRemote (const char * url, sidl_BaseInterface *_ex) RMI constructor function for the class bHYPRE_StructMatrix bHYPRE_StructMatrix_wrapObj (void * data, sidl_BaseInterface *_ex) theprivatedatastructpointer(struct bHYPRE_StructMatrix_data) passed in rather than running the constructor4.3.2bHYPRE_StructMatrix **bHYPRE_StructMatrix__connect** (const char *, sidl_BaseInterface *_ex) RMI connector function for the class 61 bHYPRE_StructMatrix bHYPRE_StructMatrix_Create (bHYPRE_MPICommunicator mpi_comm, bHYPRE_StructGrid grid, bHYPRE_StructStencil stencil, sidl_BaseInterface *_ex) Method: Create[] SIDL_C_INLINE_DECL int32_t bHYPRE_StructMatrix_SetGrid (bHYPRE_StructMatrix self, bHYPRE_StructGrid grid, sidl_BaseInterface *_ex) Method: SetGrid[] SIDL_C_INLINE_DECL int32_t bHYPRE_StructMatrix_SetStencil (bHYPRE_StructMatrix self, bHYPRE_StructStencil stencil, sidl_BaseInterface *_ex) Method: SetStencil/ int32 t bHYPRE_StructMatrix_SetValues (bHYPRE_StructMatrix self, int32_t* index, int32_t dim, int32_t num_stencil_indices, int32_t* stencil_indices, double* values, sidl_BaseInterface *_ex) Method: SetValues// $int32_t$ bHYPRE_StructMatrix_SetBoxValues (bHYPRE_StructMatrix self, int32_t* ilower, int32_t* iupper, int32_t dim, int32_t num_stencil_indices, int32_t* stencil_indices, double* values, int32_t nvalues, sidl_BaseInterface *_ex) Method: SetBoxValues[]

SIDL_C_INLINE_DECL int32_t

	bHYPRE_StructMatrix_SetNumGhost (bHYPRE_StructMatrix self, int32_t* num_ghost, int32_t dim2,	
	sidl_BaseInterface *_ex)	
	Method: SetNumGhost[]	
	SIDL_C_INLINE_DECL int32_t bHYPRE_StructMatrix_SetSymmetric (bHYPRE_StructMatrix self,	
	sidl_BaseInterface *_ex)	
	$Method:\ SetSymmetric[]$	
	SIDL_C_INLINE_DECL int32_t bHYPRE_StructMatrix_SetConstantEntries (bHYPRE_StructMatrix self, int32_t num_stencil_constant_points,	
	int32_t* stencil_constant_points, sidl_BaseInterface *_ex)	
	$Method:\ SetConstantEntries[]$	
	$\mathrm{int}32_\mathrm{t}$	
	bHYPRE_StructMatrix_SetConstantValues (bHYPRE_StructMatrix self, int32_t num_stencil_indices, int32_t* stencil_indices, double* values, sidl_BaseInterface *_ex)	
	Method: SetConstantValues[]	
1.3.3	SIDL_C_INLINE_DECL int32_t bHYPRE_StructMatrix_SetCommunicator (bHYPRE_StructMatrix self, bHYPRE_MPICommunicator mpi_comm, sidl_BaseInterface *_ex)	
	Set the MPI Communicator	61
1.3.4	SIDL_C_INLINE_DECL_void bHYPRE_StructMatrix_Destroy (bHYPRE_StructMatrix self, sidl_BaseInterface *_ex)	
	The Destroy function doesn't necessarily destroy anything	62
	SIDL_C_INLINE_DECL_int32_t bHYPRE_StructMatrix_Initialize (bHYPRE_StructMatrix self,	
	Prepare an object for setting coefficient values, whether for the first time or subsequently	
1.3.5	SIDL_C_INLINE_DECL int32_t bHYPRE_StructMatrix_Assemble (bHYPRE_StructMatrix self, sidl_BaseInterface *_ex)	
	Finalize the construction of an object before using, either for the first time or on subsequent uses	62
	SIDL_C_INLINE_DECL int32_t bHYPRE_StructMatrix_SetIntParameter (bHYPRE_StructMatrix self,	
	Set the int parameter associated with name	
	SIDL_C_INLINE_DECL int32_t	

```
bHYPRE_StructMatrix_SetDoubleParameter ( bHYPRE_StructMatrix
                                                self, const char* name,
                                                double value,
                                                sidl_BaseInterface *_ex)
      Set the double parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_StructMatrix_SetStringParameter ( bHYPRE_StructMatrix self,
                                               const char* name,
                                               const char* value,
                                               sidl_BaseInterface *_ex)
      Set the string parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_StructMatrix_SetIntArray1Parameter ( bHYPRE_StructMatrix
                                                   self, const char* name,
                                                   int32_t* value,
                                                   int32_t nvalues.
                                                   sidl_BaseInterface *_ex)
      Set the int 1-D array parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_StructMatrix_SetIntArray2Parameter ( bHYPRE_StructMatrix
                                                   self, const char* name,
                                                   struct sidl_int_array*
                                                   value.
                                                   sidl_BaseInterface *_ex)
      Set the int 2-D array parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_StructMatrix_SetDoubleArray1Parameter (
                                                        bHYPRE_StructMatrix
                                                        self.
                                                        const char* name.
                                                        double* value,
                                                        int32_t nvalues,
                                                        sidl\_BaseInterface
                                                        *_ex)
      Set the double 1-D array parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_StructMatrix_SetDoubleArray2Parameter (
                                                        bHYPRE_StructMatrix
                                                        self.
                                                        const char* name,
                                                        struct
                                                        sidl_double_array*
                                                        value.
                                                        sidl_BaseInterface
                                                        *_ex)
      Set the double 2-D array parameter associated with name
```

SIDL_C_INLINE_DECL int32_t

```
bHYPRE_StructMatrix_GetIntValue ( bHYPRE_StructMatrix self,
                                        const char* name, int32_t* value,
                                        sidl_BaseInterface *_ex)
      Set the int parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_StructMatrix_GetDoubleValue ( bHYPRE_StructMatrix self,
                                             const char* name,
                                             double* value,
                                             sidl_BaseInterface *_ex)
       Get the double parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_StructMatrix_Setup ( bHYPRE_StructMatrix self,
                                 bHYPRE_Vector b, bHYPRE_Vector x,
                                 sidl_BaseInterface *_ex)
       (Optional) Do any preprocessing that may be necessary in order to execute
      Apply
SIDL_C_INLINE_DECL int32_t
bHYPRE_StructMatrix_Apply ( bHYPRE_StructMatrix self,
                                 bHYPRE_Vector b, bHYPRE_Vector* x,
                                 sidl_BaseInterface *_ex)
      Apply the operator to b, returning x
SIDL_C_INLINE_DECL int32_t
\mathbf{bHYPRE\_StructMatrix\_ApplyAdjoint} \ ( \ \ \mathbf{bHYPRE\_StructMatrix} \ \mathbf{self},
                                         bHYPRE_Vector b,
                                         bHYPRE_Vector* x.
                                         sidl_BaseInterface *_ex)
      Apply the adjoint of the operator to b, returning x
_ex
       Cast method for interface and class type conversions
void*
bHYPRE_StructMatrix_cast2 (void* obj, const char* type,
                                 sidl_BaseInterface *_ex)
       String cast method for interface and class type conversions
SIDL_C_INLINE_DECL_void
bHYPRE_StructMatrix_exec ( bHYPRE_StructMatrix self,
                                const char* methodName,
                                sidl_rmi_Call inArgs,
                                sidl_rmi_Return outArgs,
                                sidl_BaseInterface *_ex)
      Select and execute a method by name
SIDL_C_INLINE_DECL char*
bHYPRE_StructMatrix_getURL ( bHYPRE_StructMatrix self,
                                    sidl_BaseInterface *_ex)
       Get the URL of the Implementation of this object (for RMI)
SIDL_C_INLINE_DECL void
bHYPRE_StructMatrix_raddRef ( bHYPRE_StructMatrix self,
                                     sidl_BaseInterface *_ex)
       On a remote object, address the remote instance
```

SIDL_C_INLINE_DECL_sidl_bool

bHYPRE_StructMatrix__isRemote (bHYPRE_StructMatrix self, sidl_BaseInterface *_ex)

TRUE if this object is remote, false if local

sidl_bool

bHYPRE_StructMatrix__isLocal (bHYPRE_StructMatrix self,

bHYPRE_StructMatrix__isLocal (bHYPRE_StructMatrix self, sidl_BaseInterface *_ex)

TRUE if this object is remote, false if local

 $**_ex$

Cast method for interface and class type conversions

4.3.6 **_ex

4.3.1

struct bHYPRE_StructMatrix_object

Symbol "bHYPRE.StructMatrix" (version 1.0.0)

A single class that implements both a view interface and an operator interface. A StructMatrix is a matrix on a structured grid. One function unique to a StructMatrix is SetConstantEntries. This declares that matrix entries corresponding to certain stencil points (supplied as stencil element indices) will be constant throughout the grid.

4.3.2

bHYPRE_StructMatrix bHYPRE_StructMatrix_connect (const char *, sidl_BaseInterface *_ex)

RMI connector function for the class.(addrefs)

4.3.3

SIDL_C_INLINE_DECL int32_t bHYPRE_StructMatrix_SetCommunicator (bHYPRE_StructMatrix self, bHYPRE_MPICommunicator mpi_comm, sidl_BaseInterface *_ex)

Set the MPI Communicator. DEPRECATED, Use Create()

4.3.4

SIDL_C_INLINE_DECL_void bHYPRE_StructMatrix_Destroy (bHYPRE_StructMatrix self, sidl_BaseInterface *_ex)

The Destroy function doesn't necessarily destroy anything. It is just another name for deleteRef. Thus it decrements the object's reference count. The Babel memory management system will destroy the object if the reference count goes to zero.

4.3.5

Finalize the construction of an object before using, either for the first time or on subsequent uses. Initialize and Assemble always appear in a matched set, with Initialize preceding Assemble. Values can only be set in between a call to Initialize and Assemble.

4.3.6

struct bHYPRE_StructMatrix_object* bHYPRE_StructMatrix_connectI const char * url sidl_bool ar struct sidl_BaseInterface_object ** ex

RMI connector function for the class. (no addref)

4.4

Struct Vector

Names

 $_{\mathbf{ex}}$

	Constructor function for the class	
	bHYPRE_StructVector	
	bHYPRE_StructVectorcreateRemote (const char * url, sidl_BaseInterface *_ex)	
	RMI constructor function for the class	
	bHYPRE_StructVector	
	bHYPRE_StructVectorwrapObj (void * data, sidl_BaseInterface *_ex) Wraps up the private data struct pointer (struct bHYPRE_StructVectordata) passed in rather than running the con- structor	
4.4.2	bHYPRE_StructVector bHYPRE_StructVectorconnect (const char *, sidl_BaseInterface *_ex) RMI connector function for the class	65
	bHYPRE_StructVector bHYPRE_MPICommunicator mpi_comm, bHYPRE_StructGrid grid, sidl_BaseInterface *_ex)	
	Method: Create[]	
	SIDL_C_INLINE_DECL int32_t bHYPRE_StructVector_SetGrid (bHYPRE_StructVector self,	
	Method: SetGrid[]	
	SIDL_C_INLINE_DECL int32_t	
	bHYPRE_StructVector_SetNumGhost (bHYPRE_StructVector self, int32_t* num_ghost, int32_t dim2, sidl_BaseInterface *_ex)	
	$Method:\ SetNumGhost[]$	
	SIDL_C_INLINE_DECL int32_t bHYPRE_StructVector_SetValue (bHYPRE_StructVector self,	
	Method: SetValue[]	
	int32_t bHYPRE_StructVector_SetBoxValues (bHYPRE_StructVector self,	
	$Method:\ SetBoxValues[]$	
4.4.3	SIDL_C_INLINE_DECL int32_t bHYPRE_StructVector_SetCommunicator (bHYPRE_StructVector self, bHYPRE_MPICommunicator mpi_comm, sidl_BaseInterface *_ex)	
	Set the MPI Communicator	66
4.4.4	SIDL_C_INLINE_DECL_void	

	bHYPRE_StructVector_Destroy (bHYPRE_StructVector self,	
	sidl_BaseInterface *_ex)	
	The Destroy function doesn't necessarily destroy anything	66
	SIDL_C_INLINE_DECL int32_t bHYPRE_StructVector_Initialize (bHYPRE_StructVector self,	
	sidl_BaseInterface *_ex)	
	Prepare an object for setting coefficient values, whether for the first time or subsequently	
4.4.5	SIDL_C_INLINE_DECL int32_t bHYPRE_StructVector_Assemble (bHYPRE_StructVector self,	
	sidl_BaseInterface *_ex) Finalize the construction of an object before using, either for the first time	C
	or on subsequent uses	66
	SIDL_C_INLINE_DECL int32_t bHYPRE_StructVector_Clear (bHYPRE_StructVector self,	
	Set self to 0	
	SIDL_C_INLINE_DECL int32_t	
	bHYPRE_StructVector_Copy (bHYPRE_StructVector self, bHYPRE_Vector x, sidl_BaseInterface *_ex)	
	$Copy\ data\ from\ x\ into\ exttt{self}$	
1.4.6	SIDL_C_INLINE_DECL int32_t	
	bHYPRE_StructVector_Clone (bHYPRE_StructVector self, bHYPRE_Vector* x, sidl_BaseInterface *_ex)	
	Create an x compatible with self	66
	SIDL_C_INLINE_DECL int32_t bHYPRE_StructVector_Scale (bHYPRE_StructVector self, double a, sidl_BaseInterface *_ex)	
	Scale self by a	
	SIDL_C_INLINE_DECL int32_t	
	bHYPRE_StructVector_Dot (bHYPRE_StructVector self, bHYPRE_Vector x, double* d,	
	sidl_BaseInterface *_ex) Compute d, the inner-product of self and x	
	SIDL_C_INLINE_DECL int32_t	
	bHYPRE_StructVector_Axpy (bHYPRE_StructVector self, double a, bHYPRE_Vector x, sidl_BaseInterface *_ex)	
	Add ax to self	
	_ex Cast method for interface and class type conversions	
	void*	
	bHYPRE_StructVectorcast2 (void* obj, const char* type, sidl_BaseInterface *_ex)	
	String cast method for interface and class type conversions	
	SIDL_C_INLINE_DECL_void	

```
bHYPRE_StructVector_exec ( bHYPRE_StructVector self.
                                            const char* methodName,
                                            sidl_rmi_Call inArgs, sidl_rmi_Return outArgs,
                                            sidl_BaseInterface *_ex)
                   Select and execute a method by name
            SIDL_C_INLINE_DECL char*
            bHYPRE_StructVector__getURL ( bHYPRE_StructVector self,
                                               sidl_BaseInterface *_ex)
                   Get the URL of the Implementation of this object (for RMI)
            SIDL_C_INLINE_DECL void
            bHYPRE_StructVector_raddRef ( bHYPRE_StructVector self,
                                               sidl_BaseInterface *_ex)
                   On a remote object, addrefs the remote instance
            SIDL_C_INLINE_DECL sidl_bool
            bHYPRE_StructVector_isRemote ( bHYPRE_StructVector self,
                                                sidl_BaseInterface *_ex)
                   TRUE if this object is remote, false if local
            sidl\_bool
            bHYPRE_StructVector__isLocal ( bHYPRE_StructVector self,
                                              sidl_BaseInterface *_ex)
                   TRUE if this object is remote, false if local
            **_{-ex}
                   Cast method for interface and class type conversions
            **_ex
4.4.7
                   RMI connector function for the class .....
                                                                                             67
```

4.4.1 _

struct bHYPRE_StructVector_object

Symbol "bHYPRE.StructVector" (version 1.0.0)

4.4.2

bHYPRE_StructVector bHYPRE_StructVector__connect (const char *, sidl_BaseInterface *_ex)

RMI connector function for the class.(addrefs)

4.4.3

SIDL_C_INLINE_DECL int32_t bHYPRE_StructVector_SetCommunicator (bHYPRE_StructVector self, bHYPRE_MPICommunicator mpi_comm, sidl_BaseInterface *_ex)

Set the MPI Communicator. DEPRECATED, Use Create()

4.4.4

SIDL_C_INLINE_DECL void bHYPRE_StructVector_Destroy (bHYPRE_StructVector self, sidl_BaseInterface *_ex)

The Destroy function doesn't necessarily destroy anything. It is just another name for deleteRef. Thus it decrements the object's reference count. The Babel memory management system will destroy the object if the reference count goes to zero.

4.4.5

SIDL_C_INLINE_DECL int32_t bHYPRE_StructVector_Assemble (bHYPRE_StructVector self, sidl_BaseInterface *_ex)

Finalize the construction of an object before using, either for the first time or on subsequent uses. Initialize and Assemble always appear in a matched set, with Initialize preceding Assemble. Values can only be set in between a call to Initialize and Assemble.

4.4.6

SIDL_C_INLINE_DECL int32_t bHYPRE_StructVector_Clone (bHYPRE_StructVector self, bHYPRE_Vector* x, sidl_BaseInterface *_ex)

Create an x compatible with self. The new vector's data is not specified.

NOTE: When this method is used in an inherited class, the cloned Vector object can be cast to an object with the inherited class type.

 $_$ 4.4.7 $_$

struct bHYPRE_StructVector_object* bHYPRE_StructVector_connectI const char * url sidl_bool ar struct sidl_BaseInterface_object **_ex

RMI connector function for the class. (no addref)

-4.5 .

SemiStructured Matrix

```
Names
4.5.1
            struct\ bHYPRE\_SStructMatrix\_object
                 Symbol "bHYPRE .....
                                                                                       72
           _{\mathbf{ex}}
                  Constructor function for the class
           bHYPRE_SStructMatrix
           bHYPRE_SStructMatrix_createRemote (const char * url,
                                                   sidl_BaseInterface *_ex)
                 RMI constructor function for the class
           bHYPRE\_SStructMatrix
           bHYPRE_SStructMatrix__wrapObj (void * data, sidl_BaseInterface *_ex)
                                       private
                                the
                                                data
                                                        struct
                 bHYPRE_SStructMatrix_data) passed in rather than running the con-
                 structor
           bHYPRE\_SStructMatrix
4.5.2
           bHYPRE_SStructMatrix__connect (const char *, sidl_BaseInterface *_ex)
                 RMI connector function for the class ......
                                                                                       72
           bHYPRE_SStructMatrix
           bHYPRE_SStructMatrix_Create ( bHYPRE_MPICommunicator mpi_comm,
                                           bHYPRE_SStructGraph graph,
                                           sidl_BaseInterface *_ex)
                 Method: Create[]
           SIDL_C_INLINE_DECL int32_t
           bHYPRE_SStructMatrix_SetObjectType ( bHYPRE_SStructMatrix self,
                                                   int32_t type,
                                                   sidl_BaseInterface *_ex)
                 Method: SetObjectType[]
```

4.5.3

SIDL_C_INLINE_DECL int32_t

	bHYPRE_SStructMatrix_SetGraph (bHYPRE_SStructMatrix self, bHYPRE_SStructGraph graph, sidl_BaseInterface *_ex)	
	Set the matrix graph	72
4.5.4	$\mathrm{int}32$ _{t}	
-10-1	bHYPRE_SStructMatrix_SetValues (bHYPRE_SStructMatrix self, int32_t part, int32_t* index, int32_t dim, int32_t var, int32_t nentries, int32_t* entries, double* values, sidl_BaseInterface *_ex)	
	Set matrix coefficients index by index	73
4.5.5	int32_t bHYPRE_SStructMatrix_SetBoxValues (bHYPRE_SStructMatrix self,	
	Set matrix coefficients a box at a time	73
4.5.6	int32_t bHYPRE_SStructMatrix_AddToValues (bHYPRE_SStructMatrix self, int32_t part, int32_t* index, int32_t dim, int32_t var, int32_t nentries, int32_t* entries, double* values, sidl_BaseInterface *_ex)	
	Add to matrix coefficients index by index	74
4.5.7	int32_t bHYPRE_SStructMatrix_AddToBoxValues (bHYPRE_SStructMatrix self, int32_t part, int32_t* ilower, int32_t* iupper, int32_t dim, int32_t var, int32_t nentries, int32_t* entries, double* values, int32_t nvalues, sidl_BaseInterface *_ex)	
	Add to matrix coefficients a box at a time	74
4.5.8	SIDL_C_INLINE_DECL int32_t bHYPRE_SStructMatrix_SetSymmetric (bHYPRE_SStructMatrix self,	
	Define symmetry properties for the stencil entries in the matrix	75
	SIDL_C_INLINE_DECL int32_t bHYPRE_SStructMatrix_SetNSSymmetric (bHYPRE_SStructMatrix self, int32_t symmetric, sidl_BaseInterface *_ex) Define symmetry properties for all non-stencil matrix entries	
	SIDL_C_INLINE_DECL int32_t	

	bHYPRE_SStructMatrix_SetComplex (bHYPRE_SStructMatrix self, sidl_BaseInterface *_ex)	
	Set the matrix to be complex	
4.5.9	SIDL_C_INLINE_DECL int32_t bHYPRE_SStructMatrix_Print (bHYPRE_SStructMatrix self,	
	Print the matrix to file	75
4.5.10	SIDL_C_INLINE_DECL int32_t bHYPRE_SStructMatrix_GetObject (bHYPRE_SStructMatrix self,	
	A semi-structured matrix or vector contains a Struct or IJ matrix or vector	
		75
4.5.11	SIDL_C_INLINE_DECL int32_t bHYPRE_SStructMatrix_SetCommunicator (bHYPRE_SStructMatrix self, bHYPRE_MPICommunicator mpi_comm, sidl_BaseInterface *_ex)	
	Set the MPI Communicator	76
4.5.12	SIDL_C_INLINE_DECL void bHYPRE_SStructMatrix_Destroy (bHYPRE_SStructMatrix self,	76
	SIDL_C_INLINE_DECL int32_t bHYPRE_SStructMatrix_Initialize (bHYPRE_SStructMatrix self,	
4.5.13	SIDL_C_INLINE_DECL int32_t bHYPRE_SStructMatrix_Assemble (bHYPRE_SStructMatrix self,	7.0
	or on subsequent uses	76
	SIDL_C_INLINE_DECL int32_t bHYPRE_SStructMatrix_SetIntParameter (bHYPRE_SStructMatrix self,	
	SIDL_C_INLINE_DECL int32_t	
	bHYPRE_SStructMatrix_SetDoubleParameter (bHYPRE_SStructMatrix self, const char* name, double value, sidl_BaseInterface *_ex)	
	Set the double parameter associated with name	
	SIDL_C_INLINE_DECL int32_t	

```
bHYPRE_SStructMatrix_SetStringParameter ( bHYPRE_SStructMatrix
                                                 self, const char* name,
                                                 const char* value,
                                                 sidl_BaseInterface *_ex)
      Set the string parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_SStructMatrix_SetIntArray1Parameter (
                                                     bHYPRE_SStructMatrix
                                                     self, const char* name,
                                                     int32_t* value,
                                                     int32_t nvalues,
                                                     sidl_BaseInterface *_ex)
      Set the int 1-D array parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_SStructMatrix_SetIntArray2Parameter (
                                                     bHYPRE_SStructMatrix
                                                     self, const char* name,
                                                     struct sidl_int_array*
                                                     value,
                                                     sidl_BaseInterface *_ex)
      Set the int 2-D array parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_SStructMatrix_SetDoubleArray1Parameter (
                                                         bHYPRE_SStructMatrix
                                                         self,
                                                         const char* name,
                                                         double* value,
                                                         int32_t nvalues.
                                                         sidl_BaseInterface
                                                         *_ex)
      Set the double 1-D array parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE\_SStructMatrix\_SetDoubleArray2Parameter (
                                                         bHYPRE_SStructMatrix
                                                         self.
                                                         const char* name,
                                                         struct
                                                         sidl_double_array*
                                                         value,
                                                         sidl_BaseInterface
                                                         *_ex)
      Set the double 2-D array parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_SStructMatrix_GetIntValue ( bHYPRE_SStructMatrix self,
                                         const char* name, int32_t* value,
                                         sidl_BaseInterface *_ex)
      Set the int parameter associated with name
SIDL_C_INLINE_DECL int32_t
```

```
bHYPRE_SStructMatrix_GetDoubleValue ( bHYPRE_SStructMatrix self,
                                             const char* name.
                                             double* value.
                                             sidl_BaseInterface *_ex)
       Get the double parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_SStructMatrix_Setup ( bHYPRE_SStructMatrix self,
                                  bHYPRE_Vector b, bHYPRE_Vector x,
                                  sidl_BaseInterface *_ex)
       (Optional) Do any preprocessing that may be necessary in order to execute
      Apply
SIDL_C_INLINE_DECL_int32_t
bHYPRE_SStructMatrix_Apply ( bHYPRE_SStructMatrix self,
                                  bHYPRE_Vector b, bHYPRE_Vector* x,
                                  sidl_BaseInterface *_ex)
      Apply the operator to b, returning x
SIDL_C_INLINE_DECL int32_t
bHYPRE_SStructMatrix_ApplyAdjoint ( bHYPRE_SStructMatrix self,
                                          bHYPRE_Vector b,
                                          bHYPRE_Vector* x,
                                          sidl_BaseInterface *_ex)
      Apply the adjoint of the operator to b, returning x
_{\mathbf{ex}}
       Cast method for interface and class type conversions
void*
bHYPRE_SStructMatrix_cast2 (void* obj, const char* type,
                                  sidl_BaseInterface *_ex)
      String cast method for interface and class type conversions
SIDL_C_INLINE_DECL_void
bHYPRE_SStructMatrix_exec ( bHYPRE_SStructMatrix self,
                                 const char* methodName,
                                 sidl_rmi_Call inArgs,
                                 sidl_rmi_Return outArgs,
                                 sidl_BaseInterface *_ex)
      Select and execute a method by name
SIDL_C_INLINE_DECL_char*
bHYPRE_SStructMatrix_getURL ( bHYPRE_SStructMatrix self,
                                     sidl_BaseInterface *_ex)
       Get the URL of the Implementation of this object (for RMI)
SIDL_C_INLINE_DECL void
bHYPRE_SStructMatrix_raddRef ( bHYPRE_SStructMatrix self,
                                     sidl_BaseInterface *_ex)
       On a remote object, addrefs the remote instance
SIDL_C_INLINE_DECL sidl_bool
bHYPRE_SStructMatrix_isRemote ( bHYPRE_SStructMatrix self,
                                      sidl_BaseInterface *_ex)
       TRUE if this object is remote, false if local
sidl_bool
```

bHYPRE_SStructMatrix__isLocal (bHYPRE_SStructMatrix self, sidl_BaseInterface *_ex)

TRUE if this object is remote, false if local

**_ex

Cast method for interface and class type conversions

4.5.14 **_ex

RMI connector function for the class

 $_$ 4.5.1 $_$

struct bHYPRE_SStructMatrix_object

Symbol "bHYPRE.SStructMatrix" (version 1.0.0)

The semi-structured grid matrix class.

Objects of this type can be cast to SStructMatrixView or Operator objects using the **__cast** methods.

4.5.2

bHYPRE_SStructMatrix bHYPRE_SStructMatrix_connect (const char *, sidl_BaseInterface *_ex)

RMI connector function for the class.(addrefs)

 $_$ 4.5.3 $_$

SIDL_C_INLINE_DECL int32_t bHYPRE_SStructMatrix_SetGraph (bHYPRE_SStructMatrix self, bHYPRE_SStructGraph graph, sidl_BaseInterface *_ex)

Set the matrix graph. DEPRECATED Use Create $\,$

76

4.5.4

 $int32_t$

bHYPRE_SStructMatrix_SetValues (bHYPRE_SStructMatrix self, int32_t part, int32_t* index, int32_t dim, int32_t var, int32_t nentries, int32_t* entries, double* values, sidl_BaseInterface *_ex)

Set matrix coefficients index by index.

NOTE: Users are required to set values on all processes that own the associated variables. This means that some data will be multiply defined.

NOTE: The entries in this routine must all be of the same type: either stencil or non-stencil, but not both. Also, if they are stencil entries, they must all represent couplings to the same variable type (there are no such restrictions for non-stencil entries).

If the matrix is complex, then values consists of pairs of doubles representing the real and imaginary parts of each complex value.

4.5.5

 $int32_t$

bHYPRE_SStructMatrix_SetBoxValues (bHYPRE_SStructMatrix self, int32_t part, int32_t* ilower, int32_t* iupper, int32_t dim, int32_t var, int32_t nentries, int32_t* entries, double* values, int32_t nvalues, sidl_BaseInterface *_ex)

Set matrix coefficients a box at a time.

NOTE: Users are required to set values on all processes that own the associated variables. This means that some data will be multiply defined.

NOTE: The entries in this routine must all be of the same type: either stencil or non-stencil, but not both. Also, if they are stencil entries, they must all represent couplings to the same variable type (there are no such restrictions for non-stencil entries).

If the matrix is complex, then values consists of pairs of doubles representing the real and imaginary parts of each complex value.

4.5.6

$int32_t$

bHYPRE_SStructMatrix_AddToValues (bHYPRE_SStructMatrix self, int32_t part, int32_t* index, int32_t dim, int32_t var, int32_t nentries, int32_t* entries, double* values, sidl_BaseInterface *_ex)

Add to matrix coefficients index by index.

NOTE: Users are required to set values on all processes that own the associated variables. This means that some data will be multiply defined.

NOTE: The entries in this routine must all be of the same type: either stencil or non-stencil, but not both. Also, if they are stencil entries, they must all represent couplings to the same variable type.

If the matrix is complex, then values consists of pairs of doubles representing the real and imaginary parts of each complex value.

4.5.7

$int32_t$

bHYPRE_SStructMatrix_AddToBoxValues (bHYPRE_SStructMatrix self, int32_t part, int32_t* ilower, int32_t* iupper, int32_t dim, int32_t var, int32_t nentries, int32_t* entries, double* values, int32_t nvalues, sidl_BaseInterface *_ex)

Add to matrix coefficients a box at a time.

NOTE: Users are required to set values on all processes that own the associated variables. This means that some data will be multiply defined.

NOTE: The entries in this routine must all be of stencil type. Also, they must all represent couplings to the same variable type.

If the matrix is complex, then values consists of pairs of doubles representing the real and imaginary parts of each complex value.

$_$ 4.5.8 $_$

SIDL_C_INLINE_DECL int32_t bHYPRE_SStructMatrix_SetSymmetric (bHYPRE_SStructMatrix self, int32_t part, int32_t var, int32_t to_var, int32_t symmetric, sidl_BaseInterface *_ex)

Define symmetry properties for the stencil entries in the matrix. The boolean argument symmetric is applied to stencil entries on part part that couple variable var to variable to_var. A value of -1 may be used for part, var, or to_var to specify "all". For example, if part and to_var are set to -1, then the boolean is applied to stencil entries on all parts that couple variable var to all other variables.

By default, matrices are assumed to be nonsymmetric. Significant storage savings can be made if the matrix is symmetric.

$_$ 4.5.9 $_$

SIDL_C_INLINE_DECL int32_t bHYPRE_SStructMatrix_Print (bHYPRE_SStructMatrix self, const char* filename, int32_t all, sidl_BaseInterface *_ex)

Print the matrix to file. This is mainly for debugging purposes.

4.5.10

SIDL_C_INLINE_DECL int32_t bHYPRE_SStructMatrix_GetObject (bHYPRE_SStructMatrix self, sidl_BaseInterface* A, sidl_BaseInterface*_ex)

A semi-structured matrix or vector contains a Struct or IJ matrix or vector. GetObject returns it. The returned type is a sidl.BaseInterface. A cast must be used on the returned object to convert it into a known type.

4.5.11

SIDL_C_INLINE_DECL int32_t bHYPRE_SStructMatrix_SetCommunicator (bHYPRE_SStructMatrix self, bHYPRE_MPICommunicator mpi_comm, sidl_BaseInterface *_ex)

Set the MPI Communicator. DEPRECATED, Use Create()

4.5.12

SIDL_C_INLINE_DECL void **bHYPRE_SStructMatrix_Destroy** (bHYPRE_SStructMatrix self, sidl_BaseInterface *_ex)

The Destroy function doesn't necessarily destroy anything. It is just another name for deleteRef. Thus it decrements the object's reference count. The Babel memory management system will destroy the object if the reference count goes to zero.

4.5.13

SIDL_C_INLINE_DECL int32_t bHYPRE_SStructMatrix_Assemble (bHYPRE_SStructMatrix self, sidl_BaseInterface *_ex)

Finalize the construction of an object before using, either for the first time or on subsequent uses. Initialize and Assemble always appear in a matched set, with Initialize preceding Assemble. Values can only be set in between a call to Initialize and Assemble.

_ 4.5.14 _

struct bHYPRE_SStructMatrix__object* bHYPRE_SStructMatrix__connectI const char * url sidl_bool ar struct sidl_BaseInterface__object **_ex

RMI connector function for the class. (no addref)

_ 4.6 _

SemiStructured Vector

Names		
4.6.1	struct bHYPRE_SStructVectorobject Symbol "bHYPRE	80
	_ex	
	Constructor function for the class	
	bHYPRE_SStructVector bHYPRE_SStructVectorcreateRemote (const char * url, sidl_BaseInterface *_ex)	
	RMI constructor function for the class	
	bHYPRE_SStructVector bHYPRE_SStructVectorwrapObj (void * data, sidl_BaseInterface *_ex) Wraps up the private data struct pointer (struct bHYPRE_SStructVectordata) passed in rather than running the constructor	
4.6.2	bHYPRE_SStructVector bHYPRE_SStructVectorconnect (const char *, sidl_BaseInterface *_ex) RMI connector function for the class	81
	bHYPRE_SStructVector bHYPRE_MPICommunicator mpi_comm, bHYPRE_SStructGrid grid, sidl_BaseInterface *_ex)	
	Method: Create[]	
	SIDL_C_INLINE_DECL int32_t bHYPRE_SStructVector_SetObjectType (bHYPRE_SStructVector self, int32_t type, sidl_BaseInterface *_ex)	
	Method: SetObjectType[]	
	SIDL_C_INLINE_DECL_int32_t bHYPRE_SStructVector_SetGrid (bHYPRE_SStructVector self, bHYPRE_SStructGrid grid, sidl_BaseInterface *_ex)	
	Set the vector grid	
4.6.3	SIDL_C_INLINE_DECL int32_t bHYPRE_SStructVector_SetValues (bHYPRE_SStructVector self,	
	Set vector coefficients index by index	81
464	int32 t	

	bHYPRE_SStructVector_SetBoxValues (bHYPRE_SStructVector self,	
	int32_t part, int32_t* ilower,	
	int32_t* iupper, int32_t dim,	
	int32_t var, double* values, int32_t nvalues,	
	sidl-BaseInterface *_ex)	
	Set vector coefficients a box at a time	81
		01
4.6.5	SIDL_C_INLINE_DECL_int32_t	
	bHYPRE_SStructVector_AddToValues (bHYPRE_SStructVector self,	
	$int32_t part$, $int32_t^* index$, $int32_t dim$, $int32_t var$,	
	double value,	
	sidl_BaseInterface *_ex)	
	Set vector coefficients index by index	82
4.6.6	int32_t	
	bHYPRE_SStructVector_AddToBoxValues (bHYPRE_SStructVector self,	
	int32_t part, int32_t* ilower,	
	int32_t* iupper, int32_t dim, int32_t var, double* values,	
	int32_t var, double varies,	
	sidl_BaseInterface *_ex)	
	Set vector coefficients a box at a time	82
	SIDL_C_INLINE_DECL int32_t	
	bHYPRE_SStructVector_Gather (bHYPRE_SStructVector self,	
	sidl_BaseInterface *_ex)	
	Gather vector data before calling GetValues	
4.6.7	SIDL_C_INLINE_DECL int32_t	
	bHYPRE_SStructVector_GetValues (bHYPRE_SStructVector self,	
	int32_t part, int32_t* index,	
	$int32_{-}t dim$, $int32_{-}t var$,	
	double* value, sidl_BaseInterface *_ex)	
	Get vector coefficients index by index	82
4.6.8	$int32_t$	
	bHYPRE_SStructVector_GetBoxValues (bHYPRE_SStructVector self,	
	int32_t part, int32_t* ilower,	
	int32_t* iupper, int32_t dim,	
	int32_t var, double* values, int32_t nvalues,	
	sidl_BaseInterface *_ex)	
	Get vector coefficients a box at a time	83
	•	00
	SIDL_C_INLINE_DECL_int32_t bHYPRE_SStructVector_SetComplex (bHYPRE_SStructVector self,	
	sidl_BaseInterface *_ex)	
	Set the vector to be complex	
1.0.0	•	
4.6.9	SIDL_CINLINE_DECL int32_t	
	bHYPRE_SStructVector_Print (bHYPRE_SStructVector self, const char* filename, int32_t all,	
	sidl_BaseInterface *_ex)	
	Print the vector to file	83
4.6.10	SIDL_C_INLINE_DECL_int32_t	
_, _,		

	bHYPRE_SStructVector_GetObject (bHYPRE_SStructVector self, sidl_BaseInterface* A, sidl_BaseInterface *_ex)
	A semi-structured matrix or vector contains a Struct or IJ matrix or vector
4.6.11	SIDL_C_INLINE_DECL int32_t bHYPRE_SStructVector_SetCommunicator (bHYPRE_SStructVector self, bHYPRE_MPICommunicator mpi_comm,
	sidl_BaseInterface *_ex)
	Set the MPI Communicator
4.6.12	SIDL_C_INLINE_DECL void bHYPRE_SStructVector_Destroy (bHYPRE_SStructVector self, sidl_BaseInterface *_ex)
	The Destroy function doesn't necessarily destroy anything
	SIDL_C_INLINE_DECL int32_t bHYPRE_SStructVector_Initialize (bHYPRE_SStructVector self,
4.6.13	SIDL_C_INLINE_DECL int32_t bHYPRE_SStructVector_Assemble (bHYPRE_SStructVector self, sidl_BaseInterface *_ex)
	Finalize the construction of an object before using, either for the first time or on subsequent uses
	SIDL_C_INLINE_DECL int32_t bHYPRE_SStructVector_Clear (bHYPRE_SStructVector self, sidl_BaseInterface *_ex)
	$Set \; \mathtt{self} \; to \; 0$
	SIDL_C_INLINE_DECL int32_t bHYPRE_SStructVector_Copy (bHYPRE_SStructVector self,
4.6.14	SIDL_C_INLINE_DECL int32_t bHYPRE_SStructVector_Clone (bHYPRE_SStructVector self,
	Create an x compatible with self
	SIDL_C_INLINE_DECL_int32_t bHYPRE_SStructVector_Scale (bHYPRE_SStructVector self, double a, sidl_BaseInterface *_ex)
	$Scale \; { t self} \; by \; { t a}$
	SIDL_C_INLINE_DECL int32_t bHYPRE_SStructVector_Dot (bHYPRE_SStructVector self,
	Compute d, the inner-product of self and x
	SIDL_C_INLINE_DECL int32_t

```
bHYPRE_SStructVector_Axpy ( bHYPRE_SStructVector self, double a,
                                 bHYPRE_Vector x, sidl_BaseInterface *_ex)
       Add ax to self
_{\mathbf{ex}}
       Cast method for interface and class type conversions
void*
bHYPRE_SStructVector_cast2 (void* obj, const char* type,
                                  sidl_BaseInterface *_ex)
      String cast method for interface and class type conversions
SIDL_C_INLINE_DECL void
bHYPRE_SStructVector_exec ( bHYPRE_SStructVector self,
                                 const char* methodName,
                                 sidl_rmi_Call inArgs,
                                 sidl_rmi_Return outArgs,
                                 sidl_BaseInterface *_ex)
      Select and execute a method by name
SIDL_C_INLINE_DECL char*
bHYPRE_SStructVector__getURL ( bHYPRE_SStructVector self,
                                    sidl_BaseInterface *_ex)
       Get the URL of the Implementation of this object (for RMI)
SIDL_C_INLINE_DECL void
bHYPRE_SStructVector_raddRef ( bHYPRE_SStructVector self,
                                     sidl_BaseInterface *_ex)
       On a remote object, addrefs the remote instance
SIDL_C_INLINE_DECL sidl_bool
bHYPRE_SStructVector__isRemote ( bHYPRE_SStructVector self,
                                      sidl_BaseInterface *_ex)
       TRUE if this object is remote, false if local
sidl_bool
bHYPRE_SStructVector__isLocal ( bHYPRE_SStructVector self,
                                   sidl_BaseInterface *_ex)
       TRUE if this object is remote, false if local
**_ex
       Cast method for interface and class type conversions
**_ex
       RMI connector function for the class ......
                                                                                  85
```

4.6.1

4.6.15

struct bHYPRE_SStructVector_object

Symbol "bHYPRE.SStructVector" (version 1.0.0)

The semi-structured grid vector class.

Objects of this type can be cast to SStructVectorView or Vector objects using the __cast methods.

4.6.2

bHYPRE_SStructVector bHYPRE_SStructVector_connect (const char *, sidl_BaseInterface *_ex)

RMI connector function for the class.(addrefs)

 $_{-}$ 4.6.3 $_{-}$

SIDL_C_INLINE_DECL int32_t bHYPRE_SStructVector_SetValues (bHYPRE_SStructVector self, int32_t part, int32_t* index, int32_t dim, int32_t var, double value, sidl_BaseInterface *_ex)

Set vector coefficients index by index.

NOTE: Users are required to set values on all processes that own the associated variables. This means that some data will be multiply defined.

If the vector is complex, then value consists of a pair of doubles representing the real and imaginary parts of the complex value.

4.6.4

int32_t bHYPRE_SStructVector_SetBoxValues (bHYPRE_SStructVector self, int32_t part, int32_t* ilower, int32_t* iupper, int32_t dim, int32_t var, double* values, int32_t nvalues, sidl_BaseInterface *_ex)

Set vector coefficients a box at a time.

NOTE: Users are required to set values on all processes that own the associated variables. This means that some data will be multiply defined.

If the vector is complex, then values consists of pairs of doubles representing the real and imaginary parts of each complex value.

4.6.5

SIDL_C_INLINE_DECL int32_t bHYPRE_SStructVector_AddToValues (bHYPRE_SStructVector self, int32_t part, int32_t* index, int32_t dim, int32_t var, double value, sidl_BaseInterface *_ex)

Set vector coefficients index by index.

NOTE: Users are required to set values on all processes that own the associated variables. This means that some data will be multiply defined.

If the vector is complex, then value consists of a pair of doubles representing the real and imaginary parts of the complex value.

4.6.6

int32_t
bHYPRE_SStructVector_AddToBoxValues (bHYPRE_SStructVector self,
int32_t_part__int32_t* ilogor__int32_t* iupper__int32_t_dim__int32_t_var__double*

int32_t part, int32_t* ilower, int32_t* iupper, int32_t dim, int32_t var, double* values, int32_t nvalues, sidl_BaseInterface *_ex)

Set vector coefficients a box at a time.

NOTE: Users are required to set values on all processes that own the associated variables. This means that some data will be multiply defined.

If the vector is complex, then values consists of pairs of doubles representing the real and imaginary parts of each complex value.

4.6.7

SIDL_C_INLINE_DECL int32_t

bHYPRE_SStructVector_GetValues (bHYPRE_SStructVector self, int32_t part, int32_t* index, int32_t dim, int32_t var, double* value, sidl_BaseInterface *_ex)

Get vector coefficients index by index.

NOTE: Users may only get values on processes that own the associated variables.

If the vector is complex, then value consists of a pair of doubles representing the real and imaginary parts of the complex value.

4.6.8

int32_t bHYPRE_SStructVector_GetBoxValues (bHYPRE_SStructVector self, int32_t part, int32_t* ilower, int32_t* iupper, int32_t dim, int32_t var, double* values, int32_t nvalues, sidl_BaseInterface *_ex)

Get vector coefficients a box at a time.

NOTE: Users may only get values on processes that own the associated variables.

If the vector is complex, then values consists of pairs of doubles representing the real and imaginary parts of each complex value.

4.6.9

SIDL_C_INLINE_DECL int32_t bHYPRE_SStructVector_Print (bHYPRE_SStructVector self, const char* filename, int32_t all, sidl_BaseInterface *_ex)

Print the vector to file. This is mainly for debugging purposes.

4.6.10

SIDL_C_INLINE_DECL int32_t bHYPRE_SStructVector_GetObject (bHYPRE_SStructVector self, sidl_BaseInterface* A, sidl_BaseInterface*_ex)

A semi-structured matrix or vector contains a Struct or IJ matrix or vector. GetObject returns it. The returned type is a sidl.BaseInterface. A cast must be used on the returned object to convert it into a known type.

4.6.11

SIDL_C_INLINE_DECL int32_t bHYPRE_SStructVector_SetCommunicator (bHYPRE_SStructVector self, bHYPRE_MPICommunicator mpi_comm, sidl_BaseInterface *_ex)

Set the MPI Communicator. DEPRECATED, Use Create()

4.6.12

The Destroy function doesn't necessarily destroy anything. It is just another name for deleteRef. Thus it decrements the object's reference count. The Babel memory management system will destroy the object if the reference count goes to zero.

4.6.13

SIDL_C_INLINE_DECL int32_t bHYPRE_SStructVector_Assemble (bHYPRE_SStructVector self, sidl_BaseInterface *_ex)

Finalize the construction of an object before using, either for the first time or on subsequent uses. Initialize and Assemble always appear in a matched set, with Initialize preceding Assemble. Values can only be set in between a call to Initialize and Assemble.

4.6.14

SIDL_C_INLINE_DECL int32_t bHYPRE_SStructVector_Clone (bHYPRE_SStructVector self, bHYPRE_Vector* x, sidl_BaseInterface *_ex)

Create an x compatible with self. The new vector's data is not specified.

NOTE: When this method is used in an inherited class, the cloned Vector object can be cast to an object with the inherited class type.

4.6.15

RMI connector function for the class. (no addref)

4.7

SemiStructured ParCSR Matrix

Names struct bHYPRE_SStructParCSRMatrix_object 4.7.1 Symbol "bHYPRE 91 _ex Constructor function for the class bHYPRE_SStructParCSRMatrix bHYPRE_SStructParCSRMatrix_createRemote (const char * url, sidl_BaseInterface *_ex) $RMI\ constructor\ function\ for\ the\ class$ bHYPRE_SStructParCSRMatrix bHYPRE_SStructParCSRMatrix__wrapObj (void * data, sidl_BaseInterface *_ex) Wrapstheprivatedatastructpointer(struct upbHYPRE_SStructParCSRMatrix_data) passed in rather than running the constructor 4.7.2bHYPRE_SStructParCSRMatrix bHYPRE_SStructParCSRMatrix_connect (const char *, sidl_BaseInterface *_ex) RMI connector function for the class 92 bHYPRE_SStructParCSRMatrix bHYPRE_SStructParCSRMatrix_Create (bHYPRE_MPICommunicator mpi_comm, bHYPRE_SStructGraph graph, sidl_BaseInterface *_ex) Method: Create[]

4.7.3

SIDL_C_INLINE_DECL int32_t

	${ m bHYPRE_SStructParCSRMatrix_SetGraph}$ (
	bHYPRE_SStructParCSRMatrix	
	$\operatorname{self}, \operatorname{bHYPRE_SStructGraph}$	
	$\operatorname{graph},$	
	sidl_BaseInterface *_ex)	
	Set the matrix graph	92
4.7.4	$\mathrm{int}32$ _{t}	
	bHYPRE_SStructParCSRMatrix_SetValues (
	bHYPRE_SStructParCSRMatrix	
	self, int32_t part,	
	int32_t* index, int32_t dim,	
	int32_t var, int32_t nentries,	
	$int32_{-}t^{*}$ entries,	
	double* values,	
	sidl_BaseInterface *_ex)	
	Set matrix coefficients index by index	92
4.7.5	${ m int}32$ _t	
4.1.0	bHYPRE_SStructParCSRMatrix_SetBoxValues (
	bHYPRE_SStructParCSRMatrix	
	self, int32_t part,	
	$int32_{-}t^*$ ilower,	
	$\inf 32_{-}t^* \text{ into } 32_{-}t^*$	
	int32_t dim, int32_t var,	
	$int 32_t$ nentries,	
	$int 32t^*$ entries,	
	double* values,	
	$int 32_{-t}$ nvalues,	
	sidl_BaseInterface *_ex)	
	Set matrix coefficients a box at a time	93
4.7.6	$\mathrm{int}32$ _t	
4.7.0	bHYPRE_SStructParCSRMatrix_AddToValues (
	bHYPRE_SStructParCSRMatrix	
	self, int32_t part,	
	$\frac{3cH}{10022}$ part, $\frac{3cH}{10022}$ int32_t* index,	
	int32_t dim, int32_t var,	
	int32_t nentries,	
	$int32_t^*$ entries,	
	double* values,	
	sidl_BaseInterface *_ex)	
	Add to matrix coefficients index by index	93
4 77 77	•	
4.7.7	$\mathrm{int}32$ _t	

	$b HYPRE_SStructParCSRMatrix_AddToBoxValu$	es (
		bHYPRE_SStructParCSRMatrix
		self, int32_t part,
		$int32_{-}t^*$ ilower,
		int32_t* iupper,
		$int32_{-}t dim,$
		$int32_{-}t var,$
		$int32_{-}t$ nentries,
		$int32_{-}t^*$ entries,
		double* values,
		int32_t nvalues,
		$sidl_BaseInterface$
		*_ex)
	Add to matrix coefficients a box at a time	94
1.7.8	SIDL_C_INLINE_DECL int32_t	
	${\bf bHYPRE_SStructParCSRMatrix_SetSymmetric}\ ($	
		${\it bHYPRE_SStructParCSRMatrix}$
		self, int32_t part,
		int32_t var,
		int32_t to_var,
		int32_t symmetric,
		sidl_BaseInterface *_ex)
	Define symmetry properties for the stencil entries	$s in the matrix \dots 94$
	SIDL_C_INLINE_DECL int32_t	
	$b {\bf HYPRE_SStructParCSRMatrix_SetNSSymmetr}$	ic (
		bHYPRE_SStructParCSRMatrix
		self,
		int32_t symmetric,
		$sidl_BaseInterface$
		$*_{ex}$
	Define symmetry properties for all non-stencil m	atrix entries
	SIDL_C_INLINE_DECL int32_t	
	bHYPRE_SStructParCSRMatrix_SetComplex (
		IYPRE_SStructParCSRMatrix
	se	ıf,
	sic	ll_BaseInterface *_ex)
	Set the matrix to be complex	
1.7.9	SIDL_C_INLINE_DECL int32_t	
	bHYPRE_SStructParCSRMatrix_Print (bHYPRI	E_SStructParCSRMatrix
		st char* filename,
		sidl_BaseInterface *_ex)
	Print the matrix to file	94
1.7.10	SIDL_C_INLINE_DECL int32_t	
1.1.10	bHYPRE_SStructParCSRMatrix_GetObject (
	· ·	PRE_SStructParCSRMatrix
		sidl_BaseInterface* A,
	·	BaseInterface *_ex)
	A semi-structured matrix or vector contains a Str	
		95
4.7.11	SIDL_C_INLINE_DECL int32_t	
1.1.11	01P1_0_111111111_P1O1 1H004_0	

${f bHYPRE_SStructParCSRMatrix_SetCommunicator}$	
bHYPRE_SStructParCSRM	Iatrix
self,	
bHYPRE_MPICommunicate	or
${ m mpi_comm},$	
$*_{ex}$	
Set the MPI Communicator	95
SIDL_C_INLINE_DECL void	
${\bf bHYPRE_SStructParCSRMatrix_Destroy}\ ($	
bHYPRE_SStructParCSRMatrix	
The Destroy function doesn't necessarily destroy anything	95
SIDL_C_INLINE_DECL int32_t	
${\bf bHYPRE_SStructParCSRMatrix_Initialize} \ ($	
bHYPRE_SStructParCSRMatrix	
self, sidl_BaseInterface *_ex)	
Prepare an object for setting coefficient values, whether for the first time or	
subsequently	
SIDL_C_INLINE_DECL_int32_t	
bHYPRE_SStructParCSRMatrix	
self, sidl_BaseInterface *_ex)	
Finalize the construction of an object before using, either for the first time	
or on subsequent uses	95
SIDL C INLINE DECL. int32 t	
· ·	atrix
$\operatorname{sidl_BaseInterface}$	
$*_{ex}$	
Set the int parameter associated with name	
SIDL C_INLINE_DECL_int32_t	
	RMatrix
const char* name,	
double value,	
sidl_BaseInterface	
$*_{-}\mathrm{ex})$	
Set the double parameter associated with name	
	bHYPRE_SStructParCSRN self, bHYPRE_MPICommunicat mpi_comm, sidl_BaseInterface *_ex) SIDL_C_INLINE_DECL_void bHYPRE_SStructParCSRMatrix_Destroy (

 $SIDL_C_INLINE_DECL \ int 32_t$

```
bHYPRE\_SStructParCSRMatrix\_SetStringParameter \ (
                                                         bHYPRE_SStructParCSRMatrix
                                                         const char* name,
                                                         const char* value,
                                                         sidl\_BaseInterface
                                                         *_ex)
      Set the string parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_SStructParCSRMatrix_SetIntArray1Parameter (
                                                             bHYPRE\_SStructParCSRMatrix
                                                             self, const
                                                             char* name,
                                                             int32_t* value,
                                                             int32_{-}t
                                                             nvalues,
                                                             sidl_BaseInterface
                                                             *_ex)
      Set the int 1-D array parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_SStructParCSRMatrix_SetIntArray2Parameter (
                                                             bHYPRE\_SStructParCSRMatrix
                                                             self, const
                                                             char* name,
                                                             struct
                                                             sidl_int_array*
                                                             value,
                                                             sidl\_BaseInterface
                                                              *_ex)
      Set the int 2-D array parameter associated with name
SIDL_C_INLINE_DECL int32_t
b HYPRE\_SStructParCSRMatrix\_SetDoubleArray1Parameter\ (
                                                                  bHYPRE_SStructParCSRMatrix
                                                                  self.
                                                                  const
                                                                  char*
                                                                  name,
                                                                  double*
                                                                  value.
                                                                  int32_t
                                                                  nvalues.
                                                                  sidl\_BaseInterface
                                                                  *_ex)
```

Set the double 1-D array parameter associated with name SIDL_C_INLINE_DECL int32_t

```
bHYPRE_SStructParCSRMatrix_SetDoubleArray2Parameter (
                                                                bHYPRE_SStructParCSRMatrix
                                                                self.
                                                                const
                                                                char*
                                                                name,
                                                                struct
                                                                sidl_double_array*
                                                                value.
                                                                sidl\_BaseInterface
                                                                *_ex)
      Set the double 2-D array parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_SStructParCSRMatrix_GetIntValue (
                                                bHYPRE_SStructParCSRMatrix
                                                self. const char* name.
                                                int32_t* value,
                                                sidl_BaseInterface *_ex)
      Set the int parameter associated with name
SIDL_C_INLINE_DECL int32_t
{\bf bHYPRE\_SStructParCSRMatrix\_GetDoubleValue}\ (
                                                     bHYPRE_SStructParCSRMatrix
                                                     self.
                                                     const char* name,
                                                     double* value,
                                                     sidl_BaseInterface
                                                     *_ex)
      Get the double parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_SStructParCSRMatrix_Setup ( bHYPRE_SStructParCSRMatrix
                                          self, bHYPRE_Vector b,
                                         bHYPRE_Vector x,
                                          sidl_BaseInterface *_ex)
      (Optional) Do any preprocessing that may be necessary in order to execute
      Apply
SIDL_C_INLINE_DECL_int32_t
bHYPRE_SStructParCSRMatrix_Apply ( bHYPRE_SStructParCSRMatrix
                                          self, bHYPRE_Vector b,
                                          bHYPRE_Vector* x,
                                          sidl_BaseInterface *_ex)
      Apply the operator to b, returning x
SIDL_C_INLINE_DECL int32_t
bHYPRE_SStructParCSRMatrix_ApplyAdjoint (
                                                  bHYPRE_SStructParCSRMatrix
                                                  self, bHYPRE_Vector b,
                                                  bHYPRE_Vector* x,
                                                  sidl_BaseInterface *_ex)
      Apply the adjoint of the operator to b, returning x
_ex
```

Cast method for interface and class type conversions void* bHYPRE_SStructParCSRMatrix_cast2 (void* obj, const char* type, sidl_BaseInterface *_ex) String cast method for interface and class type conversions SIDL_C_INLINE_DECL void bHYPRE_SStructParCSRMatrix_exec (bHYPRE_SStructParCSRMatrix self, const char* methodName, sidl_rmi_Call inArgs, sidl_rmi_Return outArgs, sidl_BaseInterface *_ex) Select and execute a method by name SIDL_C_INLINE_DECL char* bHYPRE_SStructParCSRMatrix__getURL (bHYPRE_SStructParCSRMatrix self, sidl_BaseInterface *_ex) Get the URL of the Implementation of this object (for RMI) SIDL_C_INLINE_DECL void bHYPRE_SStructParCSRMatrix_raddRef ($bHYPRE_SStructParCSRMatrix$ self, sidl_BaseInterface *_ex) On a remote object, addrefs the remote instance SIDL_C_INLINE_DECL sidl_bool bHYPRE_SStructParCSRMatrix__isRemote ($bHYPRE_SStructParCSRMatrix$ self, sidl_BaseInterface *_ex) TRUE if this object is remote, false if local $sidl_bool$ bHYPRE_SStructParCSRMatrix__isLocal (bHYPRE_SStructParCSRMatrix self, sidl_BaseInterface *_ex) TRUE if this object is remote, false if local $**_ex$

RMI connector function for the class

__ 4.7.1

4.7.14

struct bHYPRE_SStructParCSRMatrix_object

Cast method for interface and class type conversions

Symbol "bHYPRE.SStructParCSRMatrix" (version 1.0.0)

The SStructParCSR matrix class.

 $**_ex$

96

Objects of this type can be cast to SStructMatrixView or Operator objects using the __cast methods.

$_$ 4.7.2 $_$

bHYPRE_SStructParCSRMatrix bHYPRE_SStructParCSRMatrix_connect (const char *, sidl_BaseInterface *_ex)

RMI connector function for the class.(addrefs)

_ 4.7.3 _

SIDL_C_INLINE_DECL int32_t bHYPRE_SStructParCSRMatrix_SetGraph (bHYPRE_SStructParCSRMatrix self, bHYPRE_SStructGraph graph, sidl_BaseInterface *_ex)

Set the matrix graph. DEPRECATED Use Create

4.7.4

int32_t bHYPRE_SStructParCSRMatrix_SetValues (

bHYPRE_SStructParCSRMatrix self, int32_t part, int32_t* index, int32_t dim, int32_t var, int32_t nentries, int32_t* entries, double* values, sidl_BaseInterface *_ex)

Set matrix coefficients index by index.

NOTE: Users are required to set values on all processes that own the associated variables. This means that some data will be multiply defined.

NOTE: The entries in this routine must all be of the same type: either stencil or non-stencil, but not both. Also, if they are stencil entries, they must all represent couplings to the same variable type (there are no such restrictions for non-stencil entries).

If the matrix is complex, then values consists of pairs of doubles representing the real and imaginary parts of each complex value.

4.7.5

$int32_t$

bHYPRE_SStructParCSRMatrix_SetBoxValues (

bHYPRE_SStructParCSRMatrix self, int32_t part, int32_t* ilower, int32_t* iupper, int32_t dim, int32_t var, int32_t nentries, int32_t* entries, double* values, int32_t nvalues, sidl_BaseInterface *_ex)

Set matrix coefficients a box at a time.

NOTE: Users are required to set values on all processes that own the associated variables. This means that some data will be multiply defined.

NOTE: The entries in this routine must all be of the same type: either stencil or non-stencil, but not both. Also, if they are stencil entries, they must all represent couplings to the same variable type (there are no such restrictions for non-stencil entries).

If the matrix is complex, then values consists of pairs of doubles representing the real and imaginary parts of each complex value.

4.7.6

int32_t

${\bf bHYPRE_SStructParCSRMatrix_AddToValues} \ ($

bHYPRE_SStructParCSRMatrix self, int32_t part, int32_t* index, int32_t dim, int32_t var, int32_t nentries, int32_t* entries, double* values, sidl_BaseInterface *_ex)

Add to matrix coefficients index by index.

NOTE: Users are required to set values on all processes that own the associated variables. This means that some data will be multiply defined.

NOTE: The entries in this routine must all be of the same type: either stencil or non-stencil, but not both. Also, if they are stencil entries, they must all represent couplings to the same variable type.

If the matrix is complex, then values consists of pairs of doubles representing the real and imaginary parts of each complex value.

4.7.7

$int32_t$

$bHYPRE_SStructParCSRMatrix_AddToBoxValues \ ($

bHYPRE_SStructParCSRMatrix self, int32_t part, int32_t* ilower, int32_t* iupper, int32_t dim, int32_t var, int32_t nentries, int32_t* entries, double* values, int32_t nvalues, sidl_BaseInterface *_ex)

Add to matrix coefficients a box at a time.

NOTE: Users are required to set values on all processes that own the associated variables. This means that some data will be multiply defined.

NOTE: The entries in this routine must all be of stencil type. Also, they must all represent couplings to the same variable type.

If the matrix is complex, then values consists of pairs of doubles representing the real and imaginary parts of each complex value.

_ 4.7.8 _

SIDL_C_INLINE_DECL int32_t

bHYPRE_SStructParCSRMatrix_SetSymmetric (

bHYPRE_SStructParCSRMatrix self, int32_t part, int32_t var, int32_t to_var, int32_t symmetric, sidl_BaseInterface *_ex)

Define symmetry properties for the stencil entries in the matrix. The boolean argument symmetric is applied to stencil entries on part part that couple variable var to variable to_var. A value of -1 may be used for part, var, or to_var to specify "all". For example, if part and to_var are set to -1, then the boolean is applied to stencil entries on all parts that couple variable var to all other variables.

By default, matrices are assumed to be nonsymmetric. Significant storage savings can be made if the matrix is symmetric.

4.7.9

SIDL_C_INLINE_DECL int32_t

bHYPRE_SStructParCSRMatrix_Print (bHYPRE_SStructParCSRMatrix self, const char* filename, int32_t all, sidl_BaseInterface *_ex)

Print the matrix to file. This is mainly for debugging purposes.

4.7.10

SIDL_C_INLINE_DECL int32_t bHYPRE_SStructParCSRMatrix_GetObject (bHYPRE_SStructParCSRMatrix self, sidl_BaseInterface* A, sidl_BaseInterface*_ex)

A semi-structured matrix or vector contains a Struct or IJ matrix or vector. GetObject returns it. The returned type is a sidl.BaseInterface. A cast must be used on the returned object to convert it into a known type.

__ 4.7.11 ____

SIDL_C_INLINE_DECL int32_t bHYPRE_SStructParCSRMatrix_SetCommunicator (bHYPRE_SStructParCSRMatrix self, bHYPRE_MPICommunicator mpi_comm, sidl_BaseInterface *_ex)

Set the MPI Communicator. DEPRECATED, Use Create()

_ 4.7.12 __

SIDL_C_INLINE_DECL void bHYPRE_SStructParCSRMatrix_Destroy (bHYPRE_SStructParCSRMatrix self, sidl_BaseInterface *_ex)

The Destroy function doesn't necessarily destroy anything. It is just another name for deleteRef. Thus it decrements the object's reference count. The Babel memory management system will destroy the object if the reference count goes to zero.

_ 4.7.13 _

SIDL_C_INLINE_DECL int32_t bHYPRE_SStructParCSRMatrix_Assemble (bHYPRE_SStructParCSRMatrix self, sidl_BaseInterface *_ex) Finalize the construction of an object before using, either for the first time or on subsequent uses. Initialize and Assemble always appear in a matched set, with Initialize preceding Assemble. Values can only be set in between a call to Initialize and Assemble.

4.7.14

struct bHYPRE_SStructParCSRMatrix_object* bHYPRE_SStructParCSRMatrix_connectI const char * url sidl_bool ar struct sidl_BaseInterface_object

**_ex

RMI connector function for the class. (no addref)

_ 4.8 _

SemiStructured ParCSR Vector

Names 4.8.1 struct bHYPRE_SStructParCSRVector_object Symbol "bHYPRE 100 ex Constructor function for the class bHYPRE_SStructParCSRVector bHYPRE_SStructParCSRVector__createRemote (const char * url, sidl_BaseInterface *_ex) RMI constructor function for the class bHYPRE_SStructParCSRVector bHYPRE_SStructParCSRVector_wrapObj (void * data, sidl_BaseInterface *_ex) privatestructWrapsthedatapointer(struct bHYPRE_SStructParCSRVector__data) passed in rather than running the constructor 4.8.2bHYPRE_SStructParCSRVector $\mathbf{bHYPRE_SStructParCSRVector__connect} \ (\mathrm{const} \ \mathrm{char} \ ^*,$ sidl_BaseInterface *_ex) RMI connector function for the class 101 bHYPRE_SStructParCSRVector bHYPRE_SStructParCSRVector_Create (bHYPRE_MPICommunicator mpi_comm, bHYPRE_SStructGrid grid, sidl_BaseInterface *_ex) Method: Create[]

SIDL_C_INLINE_DECL int32_t

	bHYPRE_SStructParCSRVector_SetGrid (bHYPRE_SStructParCSRVector self, bHYPRE_SStructGrid grid, sidl_BaseInterface *_ex)	
	Set the vector grid	
4.8.3	SIDL_C_INLINE_DECL int32_t	
	bHYPRE_SStructParCSRVector_SetValues (bHYPRE_SStructParCSRVector self, int32_t part, int32_t* index, int32_t dim, int32_t var, double value, sidl_BaseInterface *_ex)	
	Set vector coefficients index by index	.01
4.8.4	$\mathrm{int}32_\mathrm{t}$	
	bHYPRE_SStructParCSRVector_SetBoxValues (
	bHYPRE_SStructParCSRVector self, int32_t part, int32_t* ilower, int32_t* iupper, int32_t dim, int32_t var, double* values, int32_t nvalues,	
	sidl_BaseInterface *_ex) Set vector coefficients a box at a time	.01
	**	.01
4.8.5	SIDL_C_INLINE_DECL int32_t bHYPRE_SStructParCSRVector_AddToValues (
	bHYPRE_SStructParCSRVector self, int32_t part, int32_t* index, int32_t dim, int32_t var, double value, sidl_BaseInterface *_ex) Set vector coefficients index by index	.02
4.8.6	$\mathrm{int}32_\mathrm{t}$	
	bHYPRE_SStructParCSRVector_AddToBoxValues (
	bHYPRE_SStructParCSRVectorself, int32_t part, int32_t* ilower, int32_t* ilower, int32_t* iupper, int32_t dim, int32_t var, double* values, int32_t nvalues, sidl_BaseInterface *_ex)	or .02
		.02
	SIDL_C_INLINE_DECL int32_t bHYPRE_SStructParCSRVector_Gather (bHYPRE_SStructParCSRVector	
	self, sidl_BaseInterface *_ex) Gather vector data before calling GetValues	
4.8.7	SIDL_C_INLINE_DECL int32_t	

	$bHYPRE_SStructParCSRVector_GetValues$ (
	bHYPRE_SStructParCSRVector	
	$self, int 32_t part,$	
	$int32_{t}$ index, $int32_{t}$ dim,	
	int32_t var, double* value,	
	sidl_BaseInterface *_ex)	400
	Get vector coefficients index by index	102
4.8.8	$\mathrm{int}32_\mathrm{t}$	
	${\bf bHYPRE_SStructParCSRVector_GetBoxValues} \ ($	
	bHYPRE_SStructParCSRVector	
	self, int32_t part,	
	int32_t* ilower,	
	$int32_t^* iupper, \\ int32_t dim, int32_t var,$	
	double* values,	
	int32 _{-t} nvalues,	
	sidl_BaseInterface *_ex)	
	Get vector coefficients a box at a time	103
	SIDL_C_INLINE_DECL_int32_t	
	bHYPRE_SStructParCSRVector_SetComplex (
	bHYPRE_SStructParCSRVector	
	self,	
	sidl_BaseInterface *_ex)	
	Set the vector to be complex	
4.8.9	SIDL_C_INLINE_DECL int32_t	
	bHYPRE_SStructParCSRVector_Print (bHYPRE_SStructParCSRVector	
	self, const char* filename,	
	int32_t all, sidl_BaseInterface *_ex) Print the vector to file	103
4040		100
4.8.10	SIDL_C_INLINE_DECL int32_t	
	${\bf bHYPRE_SStructParCSRVector_GetObject} \ (\\ {\bf bHYPRE_SStructParCSRVector}$	
	self, sidl_BaseInterface* A,	
	sidl_BaseInterface *_ex)	
	A semi-structured matrix or vector contains a Struct or IJ matrix or vector	
		103
4.8.11	SIDL_C_INLINE_DECL_int32_t	
	bHYPRE_SStructParCSRVector_SetCommunicator (
	bHYPRE_SStructParCSRV	ector
	$\operatorname{self},$	
	bHYPRE_MPICommunicat	or
	mpi_comm,	
	sidl_BaseInterface *	
	*_ex) Set the MPI Communicator	104
4010		104
4.8.12	SIDL_C_INLINE_DECL_void	
	bHYPRE_SStructParCSRVector_Destroy (bHYPRE_SStructParCSRVector self, sidl_BaseInterface *_ex)	
	The Destroy function doesn't necessarily destroy anything	104
	SIDL C INLINE DECL. int32 t	-01
	STELL OF INTERNET DEATE MESA E	

	bHYPRE_SStructParCSRVector_Initialize (
	bHYPRE_SStructParCSRVector self, sidl_BaseInterface *_ex)	
	Prepare an object for setting coefficient values, whether for the first time or subsequently	
4.8.13	SIDL_C_INLINE_DECL int32_t bHYPRE_SStructParCSRVector_Assemble (
	bHYPRE_SStructParCSRVector	
	self, sidl_BaseInterface *_ex) Finalize the construction of an object before using, either for the first time or on subsequent uses	104
	SIDL_C_INLINE_DECL int32_t bHYPRE_SStructParCSRVector_Clear (bHYPRE_SStructParCSRVector	
	self , $\operatorname{sidl_BaseInterface *_ex}$)	
	SIDL_C_INLINE_DECL int32_t	
	bHYPRE_SStructParCSRVector_Copy (bHYPRE_SStructParCSRVector self, bHYPRE_Vector x, sidl_BaseInterface *_ex)	
	$Copy \ data \ from \ x \ into \ {\tt self}$	
4.8.14	SIDL_C_INLINE_DECL int32_t bHYPRE_SStructParCSRVector_Clone (bHYPRE_SStructParCSRVector	
	self, bHYPRE_Vector* x, sidl_BaseInterface *_ex)	101
	Create an x compatible with self	104
	SIDL_C_INLINE_DECL int32_t bHYPRE_SStructParCSRVector_Scale (bHYPRE_SStructParCSRVector	
	self, double a, sidl_BaseInterface *_ex)	
	$Scale \; { t self} \; by \; { t a}$	
	SIDL_C_INLINE_DECL int32_t bHYPRE_SStructParCSRVector_Dot (bHYPRE_SStructParCSRVector self,	
	bHYPRE_Vector x, double* d, sidl_BaseInterface *_ex) Compute d, the inner-product of self and x	
	SIDL_C_INLINE_DECL int32_t	
	bHYPRE_SStructParCSRVector_Axpy (bHYPRE_SStructParCSRVector self, double a, bHYPRE_Vector x,	
	$sidl_BaseInterface *_ex)$ $Add ax to self$	
	_ex	
	Cast method for interface and class type conversions	
	void* bHYPRE_SStructParCSRVectorcast2 (void* obj, const char* type, sidl_BaseInterface *_ex)	
	String cast method for interface and class type conversions	
	SIDL_C_INLINE_DECL void	

 $b HYPRE_SStructParCSRVector__exec \ (\ bHYPRE_SStructParCSRVector$ self, const char* methodName, sidl_rmi_Call inArgs, sidl_rmi_Return outArgs, sidl_BaseInterface *_ex) Select and execute a method by name SIDL_C_INLINE_DECL char* bHYPRE_SStructParCSRVector__getURL ($bHYPRE_SStructParCSRVector$ self, sidl_BaseInterface *_ex) Get the URL of the Implementation of this object (for RMI) SIDL_C_INLINE_DECL void bHYPRE_SStructParCSRVector__raddRef ($bHYPRE_SStructParCSRVector$ self, sidl_BaseInterface *_ex) On a remote object, addrefs the remote instance SIDL_C_INLINE_DECL sidl_bool bHYPRE_SStructParCSRVector__isRemote ($b HYPRE_SStructParCSRVector$ self, sidl_BaseInterface *_ex) TRUE if this object is remote, false if local sidl_bool bHYPRE_SStructParCSRVector__isLocal (bHYPRE_SStructParCSRVector self, sidl_BaseInterface *_ex) TRUE if this object is remote, false if local **_ex Cast method for interface and class type conversions 4.8.15**_ex RMI connector function for the class 105

4.8.1

struct bHYPRE_SStructParCSRVector__object

Symbol "bHYPRE.SStructParCSRVector" (version 1.0.0)

The SStructParCSR vector class.

Objects of this type can be cast to SStructVectorView or Vector objects using the __cast methods.

4.8.2

 $bHYPRE_SStructParCSRVector\\ bHYPRE_SStructParCSRVector__connect\ (const\ char\ *,\ sidl_BaseInterface\ *_ex)$

RMI connector function for the class.(addrefs)

4.8.3

SIDL_C_INLINE_DECL int32_t bHYPRE_SStructParCSRVector_SetValues (bHYPRE_SStructParCSRVector self, int32_t part, int32_t* index, int32_t dim, int32_t var, double value,

sidl_BaseInterface *_ex)

Set vector coefficients index by index.

NOTE: Users are required to set values on all processes that own the associated variables. This means that some data will be multiply defined.

If the vector is complex, then value consists of a pair of doubles representing the real and imaginary parts of the complex value.

4.8.4

$int32_t$

bHYPRE_SStructParCSRVector_SetBoxValues (

bHYPRE_SStructParCSRVector self, int32_t part, int32_t* ilower, int32_t* iupper, int32_t dim, int32_t var, double* values, int32_t nvalues, sidl_BaseInterface *_ex)

Set vector coefficients a box at a time.

NOTE: Users are required to set values on all processes that own the associated variables. This means that some data will be multiply defined.

If the vector is complex, then values consists of pairs of doubles representing the real and imaginary parts of each complex value.

4.8.5

SIDL_C_INLINE_DECL_int32_t

$bHYPRE_SStructParCSRVector_AddToValues \ ($

bHYPRE_SStructParCSRVector self, int32_t part, int32_t* index, int32_t dim, int32_t var, double value, sidl_BaseInterface *_ex)

Set vector coefficients index by index.

NOTE: Users are required to set values on all processes that own the associated variables. This means that some data will be multiply defined.

If the vector is complex, then value consists of a pair of doubles representing the real and imaginary parts of the complex value.

4.8.6

$int32_t$

bHYPRE_SStructParCSRVector_AddToBoxValues (

bHYPRE_SStructParCSRVector self, int32_t part, int32_t* ilower, int32_t* iupper, int32_t dim, int32_t var, double* values, int32_t nvalues, sidl_BaseInterface *_ex)

Set vector coefficients a box at a time.

NOTE: Users are required to set values on all processes that own the associated variables. This means that some data will be multiply defined.

If the vector is complex, then values consists of pairs of doubles representing the real and imaginary parts of each complex value.

_ 4.8.7 _

SIDL_C_INLINE_DECL int32_t

bHYPRE_SStructParCSRVector_GetValues (

bHYPRE_SStructParCSRVector self, int32_t part, int32_t* index, int32_t dim, int32_t var, double* value, sidl_BaseInterface *_ex)

Get vector coefficients index by index.

NOTE: Users may only get values on processes that own the associated variables.

If the vector is complex, then value consists of a pair of doubles representing the real and imaginary parts of the complex value.

4.8.8

$int32_t$

bHYPRE_SStructParCSRVector_GetBoxValues (

bHYPRE_SStructParCSRVector self, int32_t part, int32_t* ilower, int32_t* iupper, int32_t dim, int32_t var, double* values, int32_t nvalues, sidl_BaseInterface *_ex)

Get vector coefficients a box at a time.

NOTE: Users may only get values on processes that own the associated variables.

If the vector is complex, then values consists of pairs of doubles representing the real and imaginary parts of each complex value.

4.8.9

SIDL_C_INLINE_DECL int32_t

bHYPRE_SStructParCSRVector_Print (bHYPRE_SStructParCSRVector self, const char* filename, int32_t all, sidl_BaseInterface *_ex)

Print the vector to file. This is mainly for debugging purposes.

4 8 10

SIDL_C_INLINE_DECL int32_t

 ${\bf bHYPRE_SStructParCSRVector_GetObject}\ ($

bHYPRE_SStructParCSRVector self, sidl_BaseInterface* A, sidl_BaseInterface*_ex)

A semi-structured matrix or vector contains a Struct or IJ matrix or vector. GetObject returns it. The returned type is a sidl.BaseInterface. A cast must be used on the returned object to convert it into a known type.

4.8.11

SIDL_C_INLINE_DECL int32_t bHYPRE_SStructParCSRVector_SetCommunicator (bHYPRE_SStructParCSRVector self, bHYPRE_MPICommunicator mpi_comm, sidl_BaseInterface *_ex)

Set the MPI Communicator. DEPRECATED, Use Create()

4.8.12

SIDL_C_INLINE_DECL void bHYPRE_SStructParCSRVector_Destroy (bHYPRE_SStructParCSRVector self, sidl_BaseInterface *_ex)

The Destroy function doesn't necessarily destroy anything. It is just another name for deleteRef. Thus it decrements the object's reference count. The Babel memory management system will destroy the object if the reference count goes to zero.

4.8.13

SIDL_C_INLINE_DECL int32_t bHYPRE_SStructParCSRVector_Assemble (bHYPRE_SStructParCSRVector self, sidl_BaseInterface *_ex)

Finalize the construction of an object before using, either for the first time or on subsequent uses. Initialize and Assemble always appear in a matched set, with Initialize preceding Assemble. Values can only be set in between a call to Initialize and Assemble.

4.8.14

SIDL_C_INLINE_DECL int32_t bHYPRE_SStructParCSRVector_Clone (bHYPRE_SStructParCSRVector self, bHYPRE_Vector* x, sidl_BaseInterface *_ex)

Create an x compatible with self. The new vector's data is not specified.

NOTE: When this method is used in an inherited class, the cloned Vector object can be cast to an object with the inherited class type.

_ 4.8.15 ___

struct bHYPRE_SStructParCSRVector_object* bHYPRE_SStructParCSRVector_connectI const char * url sidl_bool ar struct sidl_BaseInterface_object **_ex

RMI connector function for the class. (no addref)

. 5

Solver Interface

Names		
5.1	struct bHYPRE_Solver_object Symbol "bHYPRE	107
5.2	extern C bHYPRE_Solver bHYPRE_Solverconnect (const char *, sidl_BaseInterface *_ex) RMI connector function for the class	107
5.3	SIDL_C_INLINE_DECL int32_t bHYPRE_Solver_SetOperator (bHYPRE_Solver self,	108
5.4	SIDL_C_INLINE_DECL int32_t bHYPRE_Solver_SetTolerance (bHYPRE_Solver self, double tolerance, sidl_BaseInterface *_ex) (Optional) Set the convergence tolerance	108
5.5	SIDL_C_INLINE_DECL int32_t bHYPRE_Solver_SetMaxIterations (bHYPRE_Solver self,	100
	(Optional) Set maximum number of iterations	108
5.6	SIDL_C_INLINE_DECL int32_t bHYPRE_Solver_SetLogging (bHYPRE_Solver self, int32_t level,	108
5.7	SIDL_C_INLINE_DECL int32_t bHYPRE_Solver_SetPrintLevel (bHYPRE_Solver self, int32_t level, sidl_BaseInterface *_ex) (Optional) Set the print level, specifying the degree of informational data to be printed either to the screen or to a file	109
	SIDL_C_INLINE_DECL int32_t bHYPRE_Solver_GetNumIterations (bHYPRE_Solver self,	
	SIDL_C_INLINE_DECL int32_t bHYPRE_Solver_GetRelResidualNorm (bHYPRE_Solver self,	
	_ex Cast method for interface and class type conversions	
	void*	

	bHYPRE_Solvercast2 (void* obj, const char* type, sidl_BaseInterface *_ex) String cast method for interface and class type conversions	
	SIDL_C_INLINE_DECL void bHYPRE_Solverexec (bHYPRE_Solver self, const char* methodName,	
	SIDL_C_INLINE_DECL_char* bHYPRE_SolvergetURL (bHYPRE_Solver self, sidl_BaseInterface *_ex) Get the URL of the Implementation of this object (for RMI)	
	SIDL_C_INLINE_DECL void bHYPRE_SolverraddRef (bHYPRE_Solver self, sidl_BaseInterface *_ex) On a remote object, addrefs the remote instance	
	SIDL_C_INLINE_DECL sidl_bool bHYPRE_Solver_isRemote (bHYPRE_Solver self, sidl_BaseInterface *_ex) TRUE if this object is remote, false if local	
	sidl_bool bHYPRE_SolverisLocal (bHYPRE_Solver self, sidl_BaseInterface *_ex) TRUE if this object is remote, false if local	
	**_ex Cast method for interface and class type conversions	
5.8	**_ex RMI connector function for the class	109
5.9	Identity Solver (does nothing)	
5.10	Hybrid Solver	109
		116

5.1

 $struct \ \mathbf{bHYPRE_Solver_object}$

Symbol "bHYPRE.Solver" (version 1.0.0)

5.2

extern C bHYPRE_Solver ${\bf bHYPRE_Solver_connect}$ (const char *, sidl_BaseInterface *_ex)

RMI connector function for the class.(addrefs)

_ 5.3 _

SIDL_C_INLINE_DECL int32_t bHYPRE_Solver_SetOperator (bHYPRE_Solver self, bHYPRE_Operator A, sidl_BaseInterface *_ex)

Set the operator for the linear system being solved. DEPRECATED. use Create

_ 5.4 _____

SIDL_C_INLINE_DECL int32_t bHYPRE_Solver_SetTolerance (bHYPRE_Solver self, double tolerance, sidl_BaseInterface *_ex)

(Optional) Set the convergence tolerance. DEPRECATED. use SetDoubleParameter

5.5

SIDL_C_INLINE_DECL int32_t bHYPRE_Solver_SetMaxIterations (bHYPRE_Solver self, int32_t max_iterations, sidl_BaseInterface *_ex)

(Optional) Set maximum number of iterations. DEPRECATED use SetIntParameter

5.6

SIDL_C_INLINE_DECL int32_t bHYPRE_Solver_SetLogging (bHYPRE_Solver self, int32_t level, sidl_BaseInterface *_ex)

(Optional) Set the *logging level*, specifying the degree of additional informational data to be accumulated. Does nothing by default (level = 0). Other levels (if any) are implementation-specific. Must be called before Setup and Apply. DEPRECATED use SetIntParameter

_ 5.7 _

SIDL_C_INLINE_DECL int32_t bHYPRE_Solver_SetPrintLevel (bHYPRE_Solver self, int32_t level, sidl_BaseInterface *_ex)

(Optional) Set the *print level*, specifying the degree of informational data to be printed either to the screen or to a file. Does nothing by default (level=0). Other levels (if any) are implementation-specific. Must be called before Setup and Apply. DEPRECATED use SetIntParameter

_ 5.8 _

struct bHYPRE_Solver__object* bHYPRE_Solver__connectI const char * url sidl_bool ar struct sidl_BaseInterface__object **_ex

RMI connector function for the class. (no addref)

_ 5.9 _

Identity Solver (does nothing)

Names

5.9.1

_ex

 $Constructor\ function\ for\ the\ class$

bHYPRE_IdentitySolver

bHYPRE_IdentitySolver__createRemote (const char * url,

sidl_BaseInterface *_ex)

RMI constructor function for the class

bHYPRE_IdentitySolver

	bHYPRE_IdentitySolverwrapObj (void * data, sidl_BaseInterface *_ex) Wraps up the private data struct pointer (struct bHYPRE_IdentitySolverdata) passed in rather than running the con- structor	
5.9.2	bHYPRE_IdentitySolver_connect (const char *, sidl_BaseInterface *_ex) RMI connector function for the class	114
	bHYPRE_IdentitySolver_Create (bHYPRE_MPICommunicator mpi_comm, sidl_BaseInterface *_ex)	
	Method: Create[]	
5.9.3	SIDL_C_INLINE_DECL int32_t bHYPRE_IdentitySolver_SetOperator (bHYPRE_IdentitySolver self, bHYPRE_Operator A, sidl_BaseInterface *_ex)	
	Set the operator for the linear system being solved	114
5.9.4	SIDL_C_INLINE_DECL int32_t bHYPRE_IdentitySolver_SetTolerance (bHYPRE_IdentitySolver self, double tolerance, sidl_BaseInterface *_ex)	
	(Optional) Set the convergence tolerance	114
5.9.5	SIDL_C_INLINE_DECL int32_t bHYPRE_IdentitySolver_SetMaxIterations (bHYPRE_IdentitySolver self,	
	(Optional) Set maximum number of iterations	114
5.9.6	SIDL_C_INLINE_DECL int32_t bHYPRE_IdentitySolver_SetLogging (bHYPRE_IdentitySolver self,	
	tional data to be accumulated	115
5.9.7	SIDL_C_INLINE_DECL int32_t bHYPRE_IdentitySolver_SetPrintLevel (bHYPRE_IdentitySolver self, int32_t level, sidl_BaseInterface *_ex)	
	(Optional) Set the print level, specifying the degree of informational data to be printed either to the screen or to a file	115
	SIDL_C_INLINE_DECL int32_t bHYPRE_IdentitySolver_GetNumIterations (bHYPRE_IdentitySolver self, int32_t* num_iterations, sidl_BaseInterface *_ex)	
	(Optional) Return the number of iterations taken	
	SIDL_C_INLINE_DECL int32_t bHYPRE_IdentitySolver_GetRelResidualNorm (bHYPRE_IdentitySolver self, double* norm, sidl_BaseInterface *_ex)	
	(Optional) Return the norm of the relative residual	
5.9.8	SIDL_C_INLINE_DECL int32_t	

	bHYPRE_IdentitySolver_SetCommunicator (bHYPRE_IdentitySolver self, bHYPRE_MPICommunicator	
	$\mathrm{mpi_comm},$	
	sidl_BaseInterface *_ex)	
	Set the MPI Communicator	115
5.9.9	SIDL_C_INLINE_DECL_void bHYPRE_IdentitySolver_Destroy (bHYPRE_IdentitySolver_self,	
	sidl_BaseInterface *_ex) The Destroy function doesn't necessarily destroy anything	115
	SIDL_C_INLINE_DECL int32_t	110
	bHYPRE_IdentitySolver_SetIntParameter (bHYPRE_IdentitySolver self, const char* name, int32_t value,	
	sidl_BaseInterface *_ex) Set the int parameter associated with name	
	SIDL_C_INLINE_DECL int32_t bHYPRE_IdentitySolver_SetDoubleParameter (bHYPRE_IdentitySolver self, const char* name, double value, sidl_BaseInterface *_ex)	
	Set the double parameter associated with name	
	SIDL_C_INLINE_DECL int32_t bHYPRE_IdentitySolver_SetStringParameter (bHYPRE_IdentitySolver self, const char* name, const char* value, sidl_BaseInterface *_ex)	
	Set the string parameter associated with name	
	SIDL_C_INLINE_DECL int32_t bHYPRE_IdentitySolver_SetIntArray1Parameter (bHYPRE_IdentitySolver self, const char* name, int32_t* value, int32_t nvalues,	
	sidl_BaseInterface *_ex) Set the int 1-D array parameter associated with name	
	SIDL_C_INLINE_DECL int32_t bHYPRE_IdentitySolver_SetIntArray2Parameter (
	bHYPRE_IdentitySolver self, const char* name, struct sidl_intarray* value, sidl_BaseInterface *_ex)	
	Set the int 2-D array parameter associated with name	
	SIDL_C_INLINE_DECL int32_t	

```
bHYPRE_IdentitySolver_SetDoubleArray1Parameter (
                                                         bHYPRE_IdentitySolver
                                                         const char* name,
                                                         double* value,
                                                         int32_t nvalues,
                                                         sidl_BaseInterface
                                                         *_ex)
      Set the double 1-D array parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_IdentitySolver_SetDoubleArray2Parameter (
                                                         {\it bHYPRE\_IdentitySolver}
                                                         self,
                                                         const char* name,
                                                         struct
                                                         sidl_double_array*
                                                         value,
                                                         sidl_BaseInterface
                                                         *_ex)
      Set the double 2-D array parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_IdentitySolver_GetIntValue ( bHYPRE_IdentitySolver self,
                                         const char* name, int32_t* value,
                                         sidl_BaseInterface *_ex)
      Set the int parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_IdentitySolver_GetDoubleValue ( bHYPRE_IdentitySolver self,
                                             const char* name.
                                             double* value,
                                             sidl_BaseInterface *_ex)
      Get the double parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_IdentitySolver_Setup ( bHYPRE_IdentitySolver self,
                                  bHYPRE_Vector b, bHYPRE_Vector x,
                                 sidl_BaseInterface *_ex)
       (Optional) Do any preprocessing that may be necessary in order to execute
      Apply
SIDL_C_INLINE_DECL int32_t
bHYPRE_IdentitySolver_Apply ( bHYPRE_IdentitySolver self,
                                  bHYPRE_Vector b, bHYPRE_Vector* x,
                                  sidl_BaseInterface *_ex)
       Apply the operator to b, returning x
SIDL_C_INLINE_DECL int32_t
bHYPRE_IdentitySolver_ApplyAdjoint ( bHYPRE_IdentitySolver self,
                                          bHYPRE_Vector b,
                                          bHYPRE_Vector* x,
                                          sidl_BaseInterface *_ex)
      Apply the adjoint of the operator to b, returning x
```

_ex

Cast method for interface and class type conversions

void*

bHYPRE_IdentitySolver__cast2 (void* obj, const char* type, sidl_BaseInterface *_ex)

String cast method for interface and class type conversions

SIDL_C_INLINE_DECL void

bHYPRE_IdentitySolver__exec (bHYPRE_IdentitySolver self, const char* methodName, sidl_rmi_Call inArgs, sidl_rmi_Return outArgs, sidl_BaseInterface *_ex)

Select and execute a method by name

SIDL_C_INLINE_DECL char*

bHYPRE_IdentitySolver__getURL (bHYPRE_IdentitySolver self, sidl_BaseInterface *_ex)

Get the URL of the Implementation of this object (for RMI)

SIDL_C_INLINE_DECL void

bHYPRE_IdentitySolver__raddRef (bHYPRE_IdentitySolver self, sidl_BaseInterface *_ex)

On a remote object, addrefs the remote instance

SIDL_C_INLINE_DECL sidl_bool

bHYPRE_IdentitySolver__isRemote (bHYPRE_IdentitySolver self, sidl_BaseInterface *_ex)

TRUE if this object is remote, false if local

sidl_bool

bHYPRE_IdentitySolver__isLocal (bHYPRE_IdentitySolver self, sidl_BaseInterface *_ex)

TRUE if this object is remote, false if local

**_ex

Cast method for interface and class type conversions

5.9.10 ******_**ex**

5.9.1 .

struct bHYPRE_IdentitySolver__object

Symbol "bHYPRE.IdentitySolver" (version 1.0.0)

Identity solver, just solves an identity matrix, for when you don't really want a preconditioner

Objects of this type can be cast to Solver objects using the __cast methods.

5.9.2

bHYPRE_IdentitySolver

bHYPRE_IdentitySolver_connect (const char *, sidl_BaseInterface *_ex)

RMI connector function for the class.(addrefs)

_ 5.9.3 __

SIDL_C_INLINE_DECL int32_t bHYPRE_IdentitySolver_SetOperator (bHYPRE_IdentitySolver self, bHYPRE_Operator A, sidl_BaseInterface *_ex)

Set the operator for the linear system being solved. DEPRECATED. use Create

__ 5.9.4 _____

SIDL_C_INLINE_DECL int32_t

bHYPRE_IdentitySolver_SetTolerance (bHYPRE_IdentitySolver self, double tolerance, sidl_BaseInterface *_ex)

(Optional) Set the convergence tolerance. DEPRECATED. use SetDoubleParameter

5.9.5

SIDL_C_INLINE_DECL int32_t **bHYPRE_IdentitySolver_SetMaxIterations** (bHYPRE_IdentitySolver self, int32_t max_iterations, sidl_BaseInterface *_ex)

(Optional) Set maximum number of iterations. DEPRECATED use SetIntParameter

5.9.6

SIDL_C_INLINE_DECL int32_t bHYPRE_IdentitySolver_SetLogging (bHYPRE_IdentitySolver self, int32_t level, sidl_BaseInterface *_ex)

(Optional) Set the *logging level*, specifying the degree of additional informational data to be accumulated. Does nothing by default (level = 0). Other levels (if any) are implementation-specific. Must be called before Setup and Apply. DEPRECATED use SetIntParameter

 $_$ 5.9.7 $_$

SIDL_C_INLINE_DECL int32_t bHYPRE_IdentitySolver_SetPrintLevel (bHYPRE_IdentitySolver self, int32_t level, sidl_BaseInterface *_ex)

(Optional) Set the *print level*, specifying the degree of informational data to be printed either to the screen or to a file. Does nothing by default (level=0). Other levels (if any) are implementation-specific. Must be called before Setup and Apply. DEPRECATED use SetIntParameter

 $_{-}$ 5.9.8 $_{-}$

SIDL_C_INLINE_DECL int32_t bHYPRE_IdentitySolver_SetCommunicator (bHYPRE_IdentitySolver self, bHYPRE_MPICommunicator mpi_comm, sidl_BaseInterface *_ex)

Set the MPI Communicator. DEPRECATED, use Create:

 $_$ 5.9.9 $_$

SIDL_C_INLINE_DECL void bHYPRE_IdentitySolver_Destroy (bHYPRE_IdentitySolver self, sidl_BaseInterface *_ex)

The Destroy function doesn't necessarily destroy anything. It is just another name for deleteRef. Thus it decrements the object's reference count. The Babel memory management system will destroy the object if the reference count goes to zero.

5 9 10

RMI connector function for the class. (no addref)

_ 5.10 _

Hybrid Solver

Names		
5.10.1	struct bHYPRE_Hybridobject Symbol "bHYPRE	119
	_ex Constructor function for the class	
	bHYPRE_Hybrid bHYPRE_HybridcreateRemote (const char * url, sidl_BaseInterface *_ex) RMI constructor function for the class	
	bHYPRE_Hybrid bHYPRE_HybridwrapObj (void * data, sidl_BaseInterface *_ex) Wraps up the private data struct pointer (struct bHYPRE_Hybrid_data) passed in rather than running the constructor	
5.10.2	bHYPRE_Hybrid bHYPRE_Hybrid_connect (const char *, sidl_BaseInterface *_ex) RMI connector function for the class	120
	bHYPRE_Hybrid bHYPRE_Hybrid_Create (bHYPRE_MPICommunicator mpi_comm, bHYPRE_PreconditionedSolver SecondSolver, bHYPRE_Operator A, sidl_BaseInterface *_ex) Method: Create[]	
	SIDL_C_INLINE_DECL int32_t bHYPRE_Hybrid_GetFirstSolver (bHYPRE_Hybrid self, bHYPRE_PreconditionedSolver* FirstSolver, sidl_BaseInterface *_ex) Method: GetFirstSolver[]	
	SIDL_C_INLINE_DECL int32_t bHYPRE_Hybrid_GetSecondSolver (bHYPRE_Hybrid self, bHYPRE_PreconditionedSolver* SecondSolver, sidl_BaseInterface *_ex)	
	Method: GetSecondSolver[]	
5.10.3	SIDL_C_INLINE_DECL int32_t	

	bHYPRE_Hybrid_SetOperator (bHYPRE_Hybrid self, bHYPRE_Operator A, sidl_BaseInterface *_ex)	
	Set the operator for the linear system being solved	120
5.10.4	SIDL_C_INLINE_DECL int32_t bHYPRE_Hybrid_SetTolerance (bHYPRE_Hybrid self, double tolerance, sidl_BaseInterface *_ex)	
	(Optional) Set the convergence tolerance	120
5.10.5	SIDL_C_INLINE_DECL int32_t bHYPRE_Hybrid_SetMaxIterations (bHYPRE_Hybrid self,	
	(Optional) Set maximum number of iterations	121
5.10.6	SIDL_C_INLINE_DECL int32_t bHYPRE_Hybrid_SetLogging (bHYPRE_Hybrid self, int32_t level,	
	(Optional) Set the logging level, specifying the degree of additional informational data to be accumulated	121
5.10.7	SIDL_C_INLINE_DECL int32_t bHYPRE_Hybrid_SetPrintLevel (bHYPRE_Hybrid self, int32_t level,	
	(Optional) Set the print level, specifying the degree of informational data to be printed either to the screen or to a file	121
	SIDL_C_INLINE_DECL int32_t bHYPRE_Hybrid_GetNumIterations (bHYPRE_Hybrid self,	
	SIDL_C_INLINE_DECL int32_t bHYPRE_Hybrid_GetRelResidualNorm (bHYPRE_Hybrid self,	
5.10.8	(Optional) Return the norm of the relative residual SIDL_C_INLINE_DECL int32_t bHYPRE_Hybrid_SetCommunicator (bHYPRE_Hybrid self,	
	Set the MPI Communicator	121
5.10.9	SIDL_C_INLINE_DECL void bHYPRE_Hybrid_Destroy (bHYPRE_Hybrid self, sidl_BaseInterface *_ex) The Destroy function doesn't necessarily destroy anything	122
	SIDL_C_INLINE_DECL int32_t bHYPRE_Hybrid_SetIntParameter (bHYPRE_Hybrid self,	
	Set the int parameter associated with name	
	SIDL_C_INLINE_DECL int32_t	

```
bHYPRE_Hybrid_SetDoubleParameter ( bHYPRE_Hybrid self.
                                         const char* name, double value,
                                         sidl_BaseInterface *_ex)
      Set the double parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_Hybrid_SetStringParameter ( bHYPRE_Hybrid self,
                                        const char* name,
                                        const char* value,
                                        sidl_BaseInterface *_ex)
      Set the string parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_Hybrid_SetIntArray1Parameter ( bHYPRE_Hybrid self,
                                             const char* name,
                                            int32_t* value, int32_t nvalues,
                                            sidl_BaseInterface *_ex)
      Set the int 1-D array parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_Hybrid_SetIntArray2Parameter ( bHYPRE_Hybrid self,
                                            const char* name,
                                            struct sidl_int_array* value,
                                            sidl_BaseInterface *_ex)
      Set the int 2-D array parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_Hybrid_SetDoubleArray1Parameter ( bHYPRE_Hybrid self,
                                                 const char* name.
                                                 double* value,
                                                 int32_t nvalues,
                                                 sidl_BaseInterface *_ex)
      Set the double 1-D array parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_Hybrid_SetDoubleArray2Parameter ( bHYPRE_Hybrid self,
                                                 const char* name, struct
                                                 sidl_double_array* value,
                                                 sidl_BaseInterface *_ex)
      Set the double 2-D array parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_Hybrid_GetIntValue ( bHYPRE_Hybrid self, const char* name,
                                int32_t* value, sidl_BaseInterface *_ex)
      Set the int parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_Hybrid_GetDoubleValue ( bHYPRE_Hybrid self,
                                     const char* name, double* value,
                                     sidl_BaseInterface *_ex)
      Get the double parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_Hybrid_Setup ( bHYPRE_Hybrid self, bHYPRE_Vector b,
                          bHYPRE_Vector x, sidl_BaseInterface *_ex)
      (Optional) Do any preprocessing that may be necessary in order to execute
      Apply
```

SIDL_C_INLINE_DECL int32_t

```
bHYPRE_Hybrid_Apply ( bHYPRE_Hybrid self, bHYPRE_Vector b,
                                       bHYPRE_Vector* x, sidl_BaseInterface *_ex)
                   Apply the operator to b, returning x
            SIDL_C_INLINE_DECL int32_t
            bHYPRE_Hybrid_ApplyAdjoint ( bHYPRE_Hybrid self,
                                              bHYPRE_Vector b, bHYPRE_Vector* x,
                                              sidl_BaseInterface *_ex)
                   Apply the adjoint of the operator to b, returning x
            _ex
                   Cast method for interface and class type conversions
            void*
            bHYPRE_Hybrid__cast2 (void* obj, const char* type,
                                      sidl_BaseInterface *_ex)
                   String cast method for interface and class type conversions
            SIDL_C_INLINE_DECL void
            bHYPRE_Hybrid_exec ( bHYPRE_Hybrid self, const char* methodName,
                                      sidl_rmi_Call inArgs, sidl_rmi_Return outArgs,
                                      sidl_BaseInterface *_ex)
                   Select and execute a method by name
            SIDL_C_INLINE_DECL char*
            bHYPRE_Hybrid__getURL ( bHYPRE_Hybrid self, sidl_BaseInterface *_ex)
                   Get the URL of the Implementation of this object (for RMI)
            SIDL_C_INLINE_DECL void
            bHYPRE_Hybrid __raddRef ( bHYPRE_Hybrid self, sidl_BaseInterface *_ex)
                   On a remote object, addrefs the remote instance
            SIDL_C_INLINE_DECL sidl_bool
            bHYPRE_Hybrid_isRemote ( bHYPRE_Hybrid self,
                                          sidl_BaseInterface *_ex)
                   TRUE if this object is remote, false if local
            sidl\_bool
            bHYPRE_Hybrid__isLocal ( bHYPRE_Hybrid self, sidl_BaseInterface *_ex)
                   TRUE if this object is remote, false if local
            **_ex
                   Cast method for interface and class type conversions
            **_ex
5.10.10
                   RMI connector function for the class ......
                                                                                            122
```

5.10.1

 $struct\ bHYPRE_Hybrid__object$

Symbol "bHYPRE.Hybrid" (version 1.0.0)

Hybrid solver first tries to solve with the specified Krylov solver, preconditioned by diagonal scaling (this combination is the "first solver") If that fails to converge, it will try again with the user-specified preconditioner (this combination is the "second solver").

Specify the preconditioner by calling SecondSolver's SetPreconditioner method. If no preconditioner is specified (equivalently, if the preconditioner for SecondSolver is IdentitySolver), the preconditioner for the second try will be one of the following defaults. StructMatrix: SMG. other matrix types: not implemented

The Hybrid solver's Setup method will call Setup on KrylovSolver, so the user should not call Setup on KrylovSolver.

 $_{-}$ 5.10.2 $_{-}$

bHYPRE_Hybrid bHYPRE_Hybrid__connect (const char *, sidl_BaseInterface *_ex)

RMI connector function for the class.(addrefs)

5.10.3

SIDL_C_INLINE_DECL int32_t bHYPRE_Hybrid_SetOperator (bHYPRE_Hybrid self, bHYPRE_Operator A, sidl_BaseInterface *_ex)

Set the operator for the linear system being solved. DEPRECATED. use Create

 $_{-}$ 5.10.4 $_{-}$

SIDL_C_INLINE_DECL int32_t bHYPRE_Hybrid_SetTolerance (bHYPRE_Hybrid self, double tolerance, sidl_BaseInterface *_ex)

(Optional) Set the convergence tolerance. DEPRECATED. use SetDoubleParameter

5.10.5

SIDL_C_INLINE_DECL int32_t bHYPRE_Hybrid_SetMaxIterations (bHYPRE_Hybrid self, int32_t max_iterations, sidl_BaseInterface *_ex)

(Optional) Set maximum number of iterations. DEPRECATED use SetIntParameter

__ 5.10.6 ____

SIDL_C_INLINE_DECL int32_t bHYPRE_Hybrid_SetLogging (bHYPRE_Hybrid self, int32_t level, sidl_BaseInterface *_ex)

(Optional) Set the $logging\ level$, specifying the degree of additional informational data to be accumulated. Does nothing by default (level = 0). Other levels (if any) are implementation-specific. Must be called before Setup and Apply. DEPRECATED use SetIntParameter

5.10.7

(Optional) Set the *print level*, specifying the degree of informational data to be printed either to the screen or to a file. Does nothing by default (level=0). Other levels (if any) are implementation-specific. Must be called before Setup and Apply. DEPRECATED use SetIntParameter

__ 5.10.8 ____

SIDL_C_INLINE_DECL int32_t bHYPRE_Hybrid_SetCommunicator (bHYPRE_Hybrid self, bHYPRE_MPICommunicator mpi_comm, sidl_BaseInterface *_ex)

Set the MPI Communicator. DEPRECATED, use Create:

5.10.9

SIDL_C_INLINE_DECL void bHYPRE_Hybrid_Destroy (bHYPRE_Hybrid self, sidl_BaseInterface *_ex)

The Destroy function doesn't necessarily destroy anything. It is just another name for deleteRef. Thus it decrements the object's reference count. The Babel memory management system will destroy the object if the reference count goes to zero.

5.10.10

RMI connector function for the class. (no addref)

6

ParCSR Matrix Solvers

Names		
6.1	ParCSRDiagScale Solver	
		123
6.2	ParCSR BoomerAMG Solver	
		130
6.3	ParCSR Euclid Solver	
		138
6.4	ParCSR Schwarz Solver	
		144
6.5	ParCSR ParaSails Solver	150
		150
6.6	ParCSR Pilut Solver	157
		194

These solvers use matrix/vector storage schemes that are tailored for general sparse matrix systems.

6.1

ParCSRDiagScale Solver

Names 6.1.1 $struct \ \ \mathbf{bHYPRE_ParCSRDiagScale__object}$ Symbol "bHYPRE 128 _ex Constructor function for the class $bHYPRE_ParCSRDiagScale$ bHYPRE_ParCSRDiagScale__createRemote (const char * url, sidl_BaseInterface *_ex) RMI constructor function for the class $bHYPRE_ParCSRDiagScale$ bHYPRE_ParCSRDiagScale__wrapObj (void * data, sidl_BaseInterface *_ex) Wrapstheprivatedatastructpointer(struct $bHYPRE_ParCSRDiagScale__data) \hspace{0.2cm} passed \hspace{0.2cm} in \hspace{0.2cm} rather \hspace{0.2cm} than \hspace{0.2cm} running \hspace{0.2cm} the \hspace{0.2cm} and \hspace{$ constructor6.1.2 $bHYPRE_ParCSRDiagScale$

	bHYPRE_ParCSRDiagScaleconnect (const char *, sidl_BaseInterface *_ex) RMI connector function for the class	128
	bHYPRE_ParCSRDiagScale bHYPRE_ParCSRDiagScale_Create (bHYPRE_MPICommunicator	
	mpi_comm, bHYPRE_IJParCSRMatrix A, sidl_BaseInterface *_ex)	
	Method: Create[]	
6.1.3	SIDL_C_INLINE_DECL_int32_t bHYPRE_ParCSRDiagScale_SetOperator (bHYPRE_ParCSRDiagScale self, bHYPRE_Operator A, sidl_BaseInterface *_ex)	
	Set the operator for the linear system being solved	128
6.1.4	SIDL_C_INLINE_DECL_int32_t bHYPRE_ParCSRDiagScale_SetTolerance (bHYPRE_ParCSRDiagScale self, double tolerance, sidl_BaseInterface *_ex)	
	(Optional) Set the convergence tolerance	128
6.1.5	SIDL_C_INLINE_DECL_int32_t bHYPRE_ParCSRDiagScale_SetMaxIterations (
	bHYPRE_ParCSRDiagScale self, int32_t max_iterations, sidl_BaseInterface *_ex)	
	(Optional) Set maximum number of iterations	129
6.1.6	SIDL_C_INLINE_DECL int32_t bHYPRE_ParCSRDiagScale_SetLogging (bHYPRE_ParCSRDiagScale self, int32_t level, sidl_BaseInterface *_ex)	
	(Optional) Set the logging level, specifying the degree of additional informational data to be accumulated	129
6.1.7	SIDL_C_INLINE_DECL_int32_t bHYPRE_ParCSRDiagScale_SetPrintLevel (bHYPRE_ParCSRDiagScale self, int32_t level, sidl_BaseInterface *_ex)	
	(Optional) Set the print level, specifying the degree of informational data to be printed either to the screen or to a file	129
	SIDL_C_INLINE_DECL_int32_t bHYPRE_ParCSRDiagScale_GetNumIterations (
	bHYPRE_ParCSRDiagScale self, int32_t* num_iterations, sidl_BaseInterface *_ex)	
	(Optional) Return the number of iterations taken	
	SIDL_C_INLINE_DECL int32_t	

```
bHYPRE\_ParCSRDiagScale\_GetRelResidualNorm (
                                                               bHYPRE_ParCSRDiagScale
                                                               self, double* norm,
                                                               sidl_BaseInterface
                                                               *_ex)
                  (Optional) Return the norm of the relative residual
6.1.8
            SIDL_C_INLINE_DECL int32_t
            bHYPRE_ParCSRDiagScale_SetCommunicator (
                                                            bHYPRE_ParCSRDiagScale
                                                            self,
                                                            b HYPRE\_MPIC ommunicator
                                                            mpi_comm,
                                                            sidl_BaseInterface *_ex)
                  Set the MPI Communicator .....
                                                                                         129
6.1.9
            SIDL_C_INLINE_DECL void
            bHYPRE_ParCSRDiagScale_Destroy ( bHYPRE_ParCSRDiagScale self,
                                                  sidl_BaseInterface *_ex)
                  The Destroy function doesn't necessarily destroy anything .....
                                                                                         130
            SIDL_C_INLINE_DECL int32_t
            bHYPRE_ParCSRDiagScale_SetIntParameter (
                                                          bHYPRE\_ParCSRDiagScale
                                                          self, const char* name,
                                                          int32_t value,
                                                          sidl_BaseInterface *_ex)
                  Set the int parameter associated with name
            SIDL_C_INLINE_DECL int32_t
            bHYPRE_ParCSRDiagScale_SetDoubleParameter (
                                                              bHYPRE_ParCSRDiagScale
                                                              self,
                                                              const char* name,
                                                              double value,
                                                              sidl_BaseInterface *_ex)
                  Set the double parameter associated with name
            SIDL_C_INLINE_DECL int32_t
            bHYPRE_ParCSRDiagScale_SetStringParameter (
                                                              bHYPRE_ParCSRDiagScale
                                                              self, const char* name,
                                                              const char* value,
                                                              sidl_BaseInterface *_ex)
                  Set the string parameter associated with name
            SIDL_C_INLINE_DECL int32_t
            bHYPRE_ParCSRDiagScale_SetIntArray1Parameter (
                                                                 bHYPRE\_ParCSRDiagScale
                                                                 const char* name,
                                                                 int32_t* value,
                                                                 int32_t nvalues,
                                                                 sidl_BaseInterface
                                                                 *_ex)
                  Set the int 1-D array parameter associated with name
            SIDL_C_INLINE_DECL int32_t
```

```
bHYPRE_ParCSRDiagScale_SetIntArray2Parameter (
                                                       bHYPRE_ParCSRDiagScale
                                                       const char* name,
                                                       struct
                                                       sidl_int_array*
                                                       value.
                                                       sidl_BaseInterface
                                                        *_ex)
      Set the int 2-D array parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE\_ParCSRDiagScale\_SetDoubleArray1Parameter\ (
                                                            bHYPRE\_ParCSRDiagScale
                                                            self, const
                                                            char* name,
                                                            double* value,
                                                            int32_t nvalues,
                                                            sidl\_BaseInterface
                                                            *_ex)
      Set the double 1-D array parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_ParCSRDiagScale_SetDoubleArray2Parameter (
                                                            bHYPRE_ParCSRDiagScale
                                                            self, const
                                                            char* name,
                                                            struct
                                                            sidl_double_array*
                                                            value.
                                                            sidl_BaseInterface
                                                            *_ex)
      Set the double 2-D array parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_ParCSRDiagScale_GetIntValue ( bHYPRE_ParCSRDiagScale
                                            self, const char* name,
                                            int32_t* value,
                                            sidl_BaseInterface *_ex)
      Set the int parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_ParCSRDiagScale_GetDoubleValue (
                                                bHYPRE_ParCSRDiagScale
                                                self, const char* name,
                                                double* value,
                                                sidl_BaseInterface *_ex)
      Get the double parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_ParCSRDiagScale_Setup ( bHYPRE_ParCSRDiagScale self,
                                     bHYPRE_Vector b, bHYPRE_Vector x,
                                     sidl_BaseInterface *_ex)
      (Optional) Do any preprocessing that may be necessary in order to execute
      Apply
SIDL_C_INLINE_DECL int32_t
```

bHYPRE_ParCSRDiagScale_Apply (bHYPRE_ParCSRDiagScale self,

```
bHYPRE_Vector b.
                                     bHYPRE_Vector* x,
                                     sidl_BaseInterface *_ex)
      Apply the operator to b, returning x
SIDL_C_INLINE_DECL int32_t
bHYPRE_ParCSRDiagScale_ApplyAdjoint ( bHYPRE_ParCSRDiagScale
                                             self, bHYPRE_Vector b,
                                             bHYPRE_Vector* x,
                                             sidl_BaseInterface *_ex)
      Apply the adjoint of the operator to b, returning x
_ex
      Cast method for interface and class type conversions
void*
bHYPRE_ParCSRDiagScale__cast2 (void* obj, const char* type,
                                     sidl_BaseInterface *_ex)
      String cast method for interface and class type conversions
SIDL_C_INLINE_DECL void
bHYPRE_ParCSRDiagScale__exec ( bHYPRE_ParCSRDiagScale self,
                                    const char* methodName,
                                    sidl_rmi_Call inArgs,
                                    sidl_rmi_Return outArgs,
                                    sidl_BaseInterface *_ex)
      Select and execute a method by name
SIDL_C_INLINE_DECL char*
bHYPRE_ParCSRDiagScale__getURL ( bHYPRE_ParCSRDiagScale self,
                                        sidl_BaseInterface *_ex)
      Get the URL of the Implementation of this object (for RMI)
SIDL_C_INLINE_DECL_void
bHYPRE_ParCSRDiagScale__raddRef ( bHYPRE_ParCSRDiagScale self,
                                        sidl_BaseInterface *_ex)
      On a remote object, addrefs the remote instance
SIDL_C_INLINE_DECL sidl_bool
bHYPRE_ParCSRDiagScale__isRemote ( bHYPRE_ParCSRDiagScale self,
                                         sidl_BaseInterface *_ex)
      TRUE if this object is remote, false if local
sidl_bool
bHYPRE_ParCSRDiagScale__isLocal ( bHYPRE_ParCSRDiagScale self,
                                       sidl_BaseInterface *_ex)
      TRUE if this object is remote, false if local
**_ex
      Cast method for interface and class type conversions
**_ex
      RMI connector function for the class ......
                                                                                130
```

6.1.10

 $_{-}$ 6.1.1 $_{-}$

struct bHYPRE_ParCSRDiagScale_object

Symbol "bHYPRE.ParCSRDiagScale" (version 1.0.0)

Diagonal scaling preconditioner for ParCSR matrix class.

Objects of this type can be cast to Solver objects using the __cast methods.

 $_{-}$ 6.1.2 $_{-}$

bHYPRE_ParCSRDiagScale

bHYPRE_ParCSRDiagScale__connect (const char *, sidl_BaseInterface *_ex)

RMI connector function for the class.(addrefs)

6.1.3

SIDL_C_INLINE_DECL int32_t bHYPRE_ParCSRDiagScale_SetOperator (bHYPRE_ParCSRDiagScale self, bHYPRE_Operator A, sidl_BaseInterface *_ex)

Set the operator for the linear system being solved. DEPRECATED. use Create

6.1.4

SIDL_C_INLINE_DECL int32_t

bHYPRE_ParCSRDiagScale_SetTolerance (bHYPRE_ParCSRDiagScale self, double tolerance, sidl_BaseInterface *_ex)

(Optional) Set the convergence tolerance. DEPRECATED. use SetDoubleParameter $\,$

6.1.5

SIDL_C_INLINE_DECL int32_t bHYPRE_ParCSRDiagScale_SetMaxIterations (

bHYPRE_ParCSRDiagScale self, int32_t max_iterations, sidl_BaseInterface *_ex)

(Optional) Set maximum number of iterations. DEPRECATED use SetIntParameter

6.1.6

SIDL_C_INLINE_DECL int32_t bHYPRE_ParCSRDiagScale_SetLogging (bHYPRE_ParCSRDiagScale self, int32_t level, sidl_BaseInterface *_ex)

(Optional) Set the $logging\ level$, specifying the degree of additional informational data to be accumulated. Does nothing by default (level = 0). Other levels (if any) are implementation-specific. Must be called before Setup and Apply. DEPRECATED use SetIntParameter

__ 6.1.7 _____

SIDL_C_INLINE_DECL int32_t bHYPRE_ParCSRDiagScale_SetPrintLevel (bHYPRE_ParCSRDiagScale self, int32_t level, sidl_BaseInterface *_ex)

(Optional) Set the *print level*, specifying the degree of informational data to be printed either to the screen or to a file. Does nothing by default (level=0). Other levels (if any) are implementation-specific. Must be called before Setup and Apply. DEPRECATED use SetIntParameter

__ 6.1.8 _____

SIDL_C_INLINE_DECL int32_t bHYPRE_ParCSRDiagScale_SetCommunicator (bHYPRE_ParCSRDiagScale self, bHYPRE_MPICommunicator mpi_comm, sidl_BaseInterface *_ex)

Set the MPI Communicator. DEPRECATED, use Create:

6.1.9

SIDL_C_INLINE_DECL_void bHYPRE_ParCSRDiagScale_Destroy (bHYPRE_ParCSRDiagScale self, sidl_BaseInterface *_ex)

The Destroy function doesn't necessarily destroy anything. It is just another name for deleteRef. Thus it decrements the object's reference count. The Babel memory management system will destroy the object if the reference count goes to zero.

6.1.10

struct bHYPRE_ParCSRDiagScale__object* bHYPRE_ParCSRDiagScale__connectI const char * url sidl_bool ar struct sidl_BaseInterface__object **_ex

RMI connector function for the class. (no addref)

6.2

ParCSR BoomerAMG Solver

Names 6.2.1 struct bHYPRE_BoomerAMG__object Symbol "bHYPRE 134 _ex Constructor function for the class bHYPRE_BoomerAMG bHYPRE_BoomerAMG__createRemote (const char * url, sidl_BaseInterface *_ex) RMI constructor function for the class bHYPRE_BoomerAMG bHYPRE_BoomerAMG__wrapObj (void * data, sidl_BaseInterface *_ex) theprivatedatastructbHYPRE_BoomerAMG_data) passed in rather than running the constructor6.2.2bHYPRE_BoomerAMG bHYPRE_BoomerAMG__connect (const char *, sidl_BaseInterface *_ex) RMI connector function for the class 136 bHYPRE_BoomerAMG

	bHYPRE_BoomerAMG_Create (bHYPRE_MPICommunicator mpi_comm, bHYPRE_IJParCSRMatrix A, sidl_BaseInterface *_ex)	
	Method: Create[]	
	SIDL_C_INLINE_DECL int32_t bHYPRE_BoomerAMG_SetLevelRelaxWt (bHYPRE_BoomerAMG self,	
	double relax_wt, int32_t level, sidl_BaseInterface *_ex)	
	$Method:\ SetLevelRelaxWt[]$	
	SIDL_C_INLINE_DECL int32_t bHYPRE_BoomerAMG_InitGridRelaxation (bHYPRE_BoomerAMG self,	
	num_grid_sweeps, struct sidl_intarray**	
	grid_relax_type, struct sidl_intarray**	
	grid_relax_points,	
	int32_t coarsen_type, struct sidl_doublearray**	
	relax_weights, int32_t max_levels, sidl_BaseInterface *_ex)	
	$Method:\ InitGridRelaxation[]$	
6.2.3	SIDL_C_INLINE_DECL int32_t bHYPRE_BoomerAMG_SetOperator (bHYPRE_BoomerAMG self,	
	bHYPRE_Operator A, sidl_BaseInterface *_ex)	
	Set the operator for the linear system being solved	136
6.2.4	SIDL_C_INLINE_DECL_int32_t bHYPRE_BoomerAMG_SetTolerance (bHYPRE_BoomerAMG self, double tolerance,	
	sidl_BaseInterface *_ex) (Optional) Set the convergence tolerance	137
005		191
6.2.5	SIDL_C_INLINE_DECL int32_t bHYPRE_BoomerAMG_SetMaxIterations (bHYPRE_BoomerAMG self,	
	(Optional) Set maximum number of iterations	137
6.2.6	SIDL_C_INLINE_DECL int32_t bHYPRE_BoomerAMG_SetLogging (bHYPRE_BoomerAMG self,	
	(Optional) Set the logging level, specifying the degree of additional informational data to be accumulated	137
6.2.7	SIDL_C_INLINE_DECL int32_t bHYPRE_BoomerAMG_SetPrintLevel (bHYPRE_BoomerAMG self, int32_t level,	
	sidl_BaseInterface *_ex) (Optional) Set the print level, specifying the degree of informational data to be printed either to the screen or to a file	137
	SIDL_C_INLINE_DECL_int32_t	

	bHYPRE_BoomerAMG_GetNumIterations (bHYPRE_BoomerAMG self, int32_t* num_iterations, sidl_BaseInterface *_ex)	
	(Optional) Return the number of iterations taken	
	SIDL_C_INLINE_DECL_int32_t	
	bHYPRE_BoomerAMG_GetRelResidualNorm (bHYPRE_BoomerAMG self, double* norm, sidl_BaseInterface *_ex)	
	(Optional) Return the norm of the relative residual	
6.2.8	SIDL_C_INLINE_DECL_int32_t	
	bHYPRE_BoomerAMG_SetCommunicator (bHYPRE_BoomerAMG self, bHYPRE_MPICommunicator mpi_comm, sidl_BaseInterface *_ex)	
	Set the MPI Communicator	138
6.2.9	SIDL_C_INLINE_DECL_void bHYPRE_BoomerAMG_Destroy (bHYPRE_BoomerAMG self,	
	The Destroy function doesn't necessarily destroy anything	138
	SIDL_C_INLINE_DECL_int32_t bHYPRE_BoomerAMG_SetIntParameter (bHYPRE_BoomerAMG self,	
	Set the int parameter associated with name	
	SIDL_C_INLINE_DECL_int32_t bHYPRE_BoomerAMG_SetDoubleParameter (bHYPRE_BoomerAMG self, const char* name, double value, sidl_BaseInterface *_ex)	
	Set the double parameter associated with name	
	SIDL_C_INLINE_DECL int32_t bHYPRE_BoomerAMG_SetStringParameter (bHYPRE_BoomerAMG self, const char* name, const char* value, sidl_BaseInterface *_ex)	
	Set the string parameter associated with name	
	SIDL_C_INLINE_DECL_int32_t bHYPRE_BoomerAMG_SetIntArray1Parameter (bHYPRE_BoomerAMG self, const char* name, int32_t* value, int32_t nvalues, sidl_BaseInterface *_ex)	
	Set the int 1-D array parameter associated with name	
	SIDL_C_INLINE_DECL_int32_t	

```
bHYPRE_BoomerAMG_SetIntArray2Parameter ( bHYPRE_BoomerAMG
                                                  self, const char* name,
                                                  struct sidl_int_array*
                                                  value.
                                                  sidl_BaseInterface *_ex)
      Set the int 2-D array parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_BoomerAMG_SetDoubleArray1Parameter (
                                                      bHYPRE_BoomerAMG
                                                      self.
                                                      const char* name,
                                                      double* value,
                                                      int32_t nvalues,
                                                      sidl_BaseInterface
                                                      *_ex)
      Set the double 1-D array parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_BoomerAMG_SetDoubleArray2Parameter (
                                                      bHYPRE_BoomerAMG
                                                      self.
                                                      const char* name,
                                                      struct
                                                      sidl_double_array*
                                                      value.
                                                      sidl\_BaseInterface
                                                      *_ex)
      Set the double 2-D array parameter associated with name
SIDL_C_INLINE_DECL_int32_t
bHYPRE_BoomerAMG_GetIntValue ( bHYPRE_BoomerAMG self,
                                      const char* name, int32_t* value,
                                      sidl\_BaseInterface *\_ex)
      Set the int parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_BoomerAMG_GetDoubleValue ( bHYPRE_BoomerAMG self,
                                          const char* name.
                                           double* value.
                                          sidl_BaseInterface *_ex)
      Get the double parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_BoomerAMG_Setup ( bHYPRE_BoomerAMG self,
                                bHYPRE_Vector b, bHYPRE_Vector x,
                                sidl_BaseInterface *_ex)
      (Optional) Do any preprocessing that may be necessary in order to execute
      Apply
SIDL_C_INLINE_DECL int32_t
bHYPRE_BoomerAMG_Apply ( bHYPRE_BoomerAMG self,
                                bHYPRE_Vector b, bHYPRE_Vector* x,
                                sidl_BaseInterface *_ex)
      Apply the operator to b, returning x
SIDL_C_INLINE_DECL int32_t
```

```
bHYPRE_BoomerAMG_ApplyAdjoint ( bHYPRE_BoomerAMG self,
                                        bHYPRE_Vector b,
                                        bHYPRE_Vector* x,
                                       sidl_BaseInterface *_ex)
      Apply the adjoint of the operator to b, returning x
_ex
      Cast method for interface and class type conversions
void*
bHYPRE_BoomerAMG__cast2 (void* obj, const char* type,
                                sidl_BaseInterface *_ex)
      String cast method for interface and class type conversions
SIDL_C_INLINE_DECL void
bHYPRE_BoomerAMG__exec ( bHYPRE_BoomerAMG self,
                               const char* methodName,
                               sidl_rmi_Call inArgs,
                               sidl_rmi_Return outArgs,
                               sidl_BaseInterface *_ex)
      Select and execute a method by name
SIDL_C_INLINE_DECL char*
bHYPRE_BoomerAMG__getURL ( bHYPRE_BoomerAMG self,
                                   sidl_BaseInterface *_ex)
      Get the URL of the Implementation of this object (for RMI)
SIDL_C_INLINE_DECL void
bHYPRE_BoomerAMG_raddRef ( bHYPRE_BoomerAMG self,
                                   sidl_BaseInterface *_ex)
      On a remote object, addrefs the remote instance
SIDL_C_INLINE_DECL_sidl_bool
bHYPRE_BoomerAMG__isRemote ( bHYPRE_BoomerAMG self,
                                    sidl_BaseInterface *_ex)
      TRUE if this object is remote, false if local
sidl\_bool
bHYPRE_BoomerAMG__isLocal ( bHYPRE_BoomerAMG self,
                                 sidl_BaseInterface *_ex)
      TRUE if this object is remote, false if local
**_ex
      Cast method for interface and class type conversions
**_ex
      RMI connector function for the class .....
                                                                              138
```

6.2.1

6.2.10

struct bHYPRE_BoomerAMG__object

Symbol "bHYPRE.BoomerAMG" (version 1.0.0)

Algebraic multigrid solver, based on classical Ruge-Stueben.

BoomerAMG requires an IJParCSR matrix

The following optional parameters are available and may be set using the appropriate Parameter function (as indicated in parentheses):

MaxLevels (Int) - maximum number of multigrid levels.

StrongThreshold (Double) - AMG strength threshold.

MaxRowSum (Double) -

CoarsenType (Int) - type of parallel coarsening algorithm used.

MeasureType (Int) - type of measure used; local or global.

CycleType (Int) - type of cycle used; a V-cycle (default) or a W-cycle.

NumGridSweeps (IntArray 1D) - number of sweeps for fine and coarse grid, up and down cycle. DEP-RECATED: Use NumSweeps or Cycle?NumSweeps instead.

NumSweeps (Int) - number of sweeps for fine grid, up and down cycle.

Cycle0NumSweeps (Int) - number of sweeps for fine grid

Cycle1NumSweeps (Int) - number of sweeps for down cycle

Cycle2NumSweeps (Int) - number of sweeps for up cycle

 $\mathbf{Cycle3NumSweeps}$ (Int) - number of sweeps for coarse grid

GridRelaxType (IntArray 1D) - type of smoother used on fine and coarse grid, up and down cycle. DEPRECATED: Use RelaxType or Cycle?RelaxType instead.

RelaxType (Int) - type of smoother for fine grid, up and down cycle.

CycleORelaxType (Int) - type of smoother for fine grid

Cycle1RelaxType (Int) - type of smoother for down cycle

Cycle2RelaxType (Int) - type of smoother for up cycle

Cycle3RelaxType (Int) - type of smoother for coarse grid

GridRelaxPoints (IntArray 2D) - point ordering used in relaxation. DEPRECATED.

RelaxWeight (DoubleArray 1D) - relaxation weight for smoothed Jacobi and hybrid SOR. DEPRE-CATED: Instead, use the RelaxWt parameter and the SetLevelRelaxWt function.

RelaxWt (Int) - relaxation weight for all levels for smoothed Jacobi and hybrid SOR.

TruncFactor (Double) - truncation factor for interpolation.

JacobiTruncThreshold (Double) - threshold for truncation of Jacobi interpolation.

SmoothType (Int) - more complex smoothers.

SmoothNumLevels (Int) - number of levels for more complex smoothers.

SmoothNumSweeps (Int) - number of sweeps for more complex smoothers.

PrintFileName (String) - name of file printed to in association with SetPrintLevel. (not yet implemented).

NumFunctions (Int) - size of the system of PDEs (when using the systems version).

DOFFunc (IntArray 1D) - mapping that assigns the function to each variable (when using the systems version).

Variant (Int) - variant of Schwarz used.

Overlap (Int) - overlap for Schwarz.

DomainType (Int) - type of domain used for Schwarz.

SchwarzRlxWeight (Double) - the smoothing parameter for additive Schwarz.

DebugFlag (Int) -

The following function is specific to this class:

SetLevelRelxWeight (Double , Int) - relaxation weight for one specified level of smoothed Jacobi and hybrid SOR.

Objects of this type can be cast to Solver objects using the **__cast** methods.

6.2.2 $_{-}$

bHYPRE_BoomerAMG bHYPRE_BoomerAMG__connect (const char *, sidl_BaseInterface *_ex)

RMI connector function for the class.(addrefs)

6.2.3

SIDL_C_INLINE_DECL int32_t **bHYPRE_BoomerAMG_SetOperator** (bHYPRE_BoomerAMG self, bHYPRE_Operator A, sidl_BaseInterface *_ex)

Set the operator for the linear system being solved. DEPRECATED. use Create

6.2.4

SIDL_C_INLINE_DECL int32_t bHYPRE_BoomerAMG_SetTolerance (bHYPRE_BoomerAMG self, double tolerance, sidl_BaseInterface *_ex)

(Optional) Set the convergence tolerance. DEPRECATED. use SetDoubleParameter

__ 6.2.5 _____

SIDL_C_INLINE_DECL int32_t bHYPRE_BoomerAMG_SetMaxIterations (bHYPRE_BoomerAMG self, int32_t max_iterations, sidl_BaseInterface *_ex)

(Optional) Set maximum number of iterations. DEPRECATED use SetIntParameter

_ 6.2.6 __

SIDL_C_INLINE_DECL int32_t bHYPRE_BoomerAMG_SetLogging (bHYPRE_BoomerAMG self, int32_t level, sidl_BaseInterface *_ex)

(Optional) Set the $logging\ level$, specifying the degree of additional informational data to be accumulated. Does nothing by default (level = 0). Other levels (if any) are implementation-specific. Must be called before Setup and Apply. DEPRECATED use SetIntParameter

6.2.7

SIDL_C_INLINE_DECL int32_t bHYPRE_BoomerAMG_SetPrintLevel (bHYPRE_BoomerAMG self, int32_t level, sidl_BaseInterface *_ex)

(Optional) Set the *print level*, specifying the degree of informational data to be printed either to the screen or to a file. Does nothing by default (level=0). Other levels (if any) are implementation-specific. Must be called before Setup and Apply. DEPRECATED use SetIntParameter

6.2.8

SIDL_C_INLINE_DECL int32_t bHYPRE_BoomerAMG_SetCommunicator (bHYPRE_BoomerAMG self, bHYPRE_MPICommunicator mpi_comm, sidl_BaseInterface *_ex)

Set the MPI Communicator. DEPRECATED, use Create:

_ 6.2.9 _

SIDL_C_INLINE_DECL_void **bHYPRE_BoomerAMG_Destroy** (bHYPRE_BoomerAMG self, sidl_BaseInterface *_ex)

The Destroy function doesn't necessarily destroy anything. It is just another name for deleteRef. Thus it decrements the object's reference count. The Babel memory management system will destroy the object if the reference count goes to zero.

_ 6.2.10 __

RMI connector function for the class. (no addref)

6.3

ParCSR Euclid Solver

Names

6.3.1 struct bHYPRE_Euclid_object

Symbol "bHYPRE

142

_ex

Constructor function for the class

bHYPRE_Euclid

	bHYPRE_EuclidcreateRemote (const char * url, sidl_BaseInterface *_ex) RMI constructor function for the class	
	bHYPRE_Euclid bHYPRE_EuclidwrapObj (void * data, sidl_BaseInterface *_ex) Wraps up the private data struct pointer (struct bHYPRE_Eucliddata) passed in rather than running the constructor	
6.3.2	bHYPRE_Euclid bHYPRE_Euclidconnect (const char *, sidl_BaseInterface *_ex) RMI connector function for the class	142
	bHYPRE_Euclid bHYPRE_MPICommunicator mpi_comm, bHYPRE_IJParCSRMatrix A, sidl_BaseInterface *_ex)	
	Method: Create[]	
	SIDL_C_INLINE_DECL int32_t bHYPRE_Euclid_SetParameters (bHYPRE_Euclid self, int32_t argc,	
	$Method:\ SetParameters[]$	
6.3.3	SIDL_C_INLINE_DECL int32_t bHYPRE_Euclid_SetOperator (bHYPRE_Euclid self, bHYPRE_Operator A, sidl_BaseInterface *_ex)	
	Set the operator for the linear system being solved	142
6.3.4	SIDL_C_INLINE_DECL int32_t bHYPRE_Euclid_SetTolerance (bHYPRE_Euclid self, double tolerance, sidl_BaseInterface *_ex)	
	(Optional) Set the convergence tolerance	143
6.3.5	SIDL_C_INLINE_DECL int32_t bHYPRE_Euclid_SetMaxIterations (bHYPRE_Euclid self,	
	(Optional) Set maximum number of iterations	143
6.3.6	SIDL_C_INLINE_DECL int32_t bHYPRE_Euclid_SetLogging (bHYPRE_Euclid self, int32_t level,	
	(Optional) Set the logging level, specifying the degree of additional informational data to be accumulated	143
6.3.7	SIDL_C_INLINE_DECL int32_t bHYPRE_Euclid_SetPrintLevel (bHYPRE_Euclid self, int32_t level, sidl_BaseInterface *_ex)	
	(Optional) Set the print level, specifying the degree of informational data to be printed either to the screen or to a file	143
	SIDL_C_INLINE_DECL int32_t bHYPRE_Euclid_GetNumIterations (bHYPRE_Euclid self,	
	(Optional) Return the number of iterations taken	
	SIDL_C_INLINE_DECL int32_t	

	bHYPRE_Euclid_GetRelResidualNorm (bHYPRE_Euclid self,
	double* norm,
	$sidl_BaseInterface *_ex)$
	(Optional) Return the norm of the relative residual
5.3.8	SIDL_C_INLINE_DECL int32_t
	bHYPRE_Euclid_SetCommunicator (bHYPRE_Euclid self,
	bHYPRE_MPICommunicator
	mpi_comm, sidl_BaseInterface *_ex)
	Set the MPI Communicator
5.3.9	SIDL_C_INLINE_DECL void
	bHYPRE_Euclid_Destroy (bHYPRE_Euclid self, sidl_BaseInterface *_ex) The Destroy function doesn't necessarily destroy anything
	SIDL_C_INLINE_DECL int32_t
	bHYPRE_Euclid_SetIntParameter (bHYPRE_Euclid self,
	const char* name, int32_t value,
	sidl_BaseInterface *_ex)
	Set the int parameter associated with name
	SIDL_C_INLINE_DECL int32_t
	bHYPRE_Euclid_SetDoubleParameter (bHYPRE_Euclid self,
	const char* name, double value,
	$sidl_BaseInterface *_ex)$
	Set the double parameter associated with name
	SIDL_C_INLINE_DECL int32_t
	bHYPRE_Euclid_SetStringParameter (bHYPRE_Euclid self,
	const char* name, const char* value,
	sidl_BaseInterface *_ex)
	Set the string parameter associated with name
	SIDL_C_INLINE_DECL int32_t
	bHYPRE_Euclid_SetIntArray1Parameter (bHYPRE_Euclid self,
	const char* name,
	$int32_{-}t^*$ value, $int32_{-}t$ nvalues,
	sidl_BaseInterface *_ex)
	Set the int 1-D array parameter associated with name
	SIDL_C_INLINE_DECL int32_t
	${\bf bHYPRE_Euclid_SetIntArray2Parameter} \ (\ \ {\bf bHYPRE_Euclid\ self},$
	const char* name,
	struct sidl_int_array* value,
	sidl_BaseInterface *_ex)
	Set the int 2-D array parameter associated with name
	SIDL_C_INLINE_DECL int32_t
	$\mathbf{bHYPRE_Euclid_SetDoubleArray1Parameter} \ (\ \mathbf{bHYPRE_Euclid\ self},$
	const char* name,
	double* value,
	int32_t nvalues,
	sidl_BaseInterface *_ex)
	Set the double 1-D array parameter associated with name
	SIDL_C_INLINE_DECL int32_t

 $\begin{tabular}{ll} bHYPRE_Euclid_SetDoubleArray2Parameter (bHYPRE_Euclid self, const char* name, struct sidl_double_array* value, sidl_BaseInterface *_ex) \end{tabular}$

Set the double 2-D array parameter associated with name

SIDL_C_INLINE_DECL int32_t

bHYPRE_Euclid_GetIntValue (bHYPRE_Euclid self, const char* name, int32_t* value, sidl_BaseInterface *_ex)

Set the int parameter associated with name

SIDL_C_INLINE_DECL int32_t

bHYPRE_Euclid_GetDoubleValue (bHYPRE_Euclid self,

const char* name, double* value, sidl_BaseInterface *_ex)

Get the double parameter associated with name

SIDL_C_INLINE_DECL int32_t

bHYPRE_Euclid_Setup (bHYPRE_Euclid self, bHYPRE_Vector b, bHYPRE_Vector x, sidl_BaseInterface *_ex)

(Optional) Do any preprocessing that may be necessary in order to execute Apply

SIDL_C_INLINE_DECL int32_t

bHYPRE_Euclid_Apply (bHYPRE_Euclid self, bHYPRE_Vector b, bHYPRE_Vector* x, sidl_BaseInterface *_ex)

Apply the operator to b, returning x

SIDL_C_INLINE_DECL int32_t

 $\begin{tabular}{ll} \begin{tabular}{ll} \beg$

sidl_BaseInterface *_ex)

Apply the adjoint of the operator to b, returning x

 $_{\mathbf{ex}}$

Cast method for interface and class type conversions

void*

bHYPRE_Euclid__cast2 (void* obj, const char* type, sidl_BaseInterface *_ex)

String cast method for interface and class type conversions

SIDL_C_INLINE_DECL void

bHYPRE_Euclid__exec (bHYPRE_Euclid self, const char* methodName, sidl_rmi_Call inArgs, sidl_rmi_Return outArgs, sidl_BaseInterface *_ex)

Select and execute a method by name

SIDL_C_INLINE_DECL char*

bHYPRE_Euclid__getURL (bHYPRE_Euclid self, sidl_BaseInterface *_ex)

Get the URL of the Implementation of this object (for RMI)

SIDL_C_INLINE_DECL void

 $\label{eq:bhypre_euclid_raddref} \begin{picture}(100,00) \put(0,0){\line(0,0){100}} \put(0,0){\lin$

SIDL_C_INLINE_DECL_sidl_bool

sidl_bool

 $\begin{tabular}{ll} \bf bHYPRE_Euclid_isLocal~(~bHYPRE_Euclid~self,~sidl_BaseInterface~*_ex)\\ TRUE~if~this~object~is~remote,~false~if~local \end{tabular}$

 $**_ex$

Cast method for interface and class type conversions

6.3.10 **_ex

6.3.1 $_{-}$

struct bHYPRE_Euclid_object

Symbol "bHYPRE.Euclid" (version 1.0.0)

Objects of this type can be cast to Solver objects using the __cast methods.

RDF: Documentation goes here. Although the usual Solver SetParameter functions are available, a Euclid-stype parameter-setting function is also available, SetParameters.

 $_{-}$ 6.3.2 $_{-}$

bHYPRE_Euclid bHYPRE_Euclid__connect (const char *, sidl_BaseInterface *_ex)

RMI connector function for the class.(addrefs)

6.3.3

SIDL_C_INLINE_DECL int32_t **bHYPRE_Euclid_SetOperator** (bHYPRE_Euclid self, bHYPRE_Operator A, sidl_BaseInterface *_ex)

Set the operator for the linear system being solved. DEPRECATED. use Create

6.3.4

SIDL_C_INLINE_DECL int32_t bHYPRE_Euclid_SetTolerance (bHYPRE_Euclid self, double tolerance, sidl_BaseInterface *_ex)

(Optional) Set the convergence tolerance. DEPRECATED. use SetDoubleParameter

__ 6.3.5 ____

SIDL_C_INLINE_DECL int32_t bHYPRE_Euclid_SetMaxIterations (bHYPRE_Euclid self, int32_t max_iterations, sidl_BaseInterface *_ex)

(Optional) Set maximum number of iterations. DEPRECATED use SetIntParameter

_ 6.3.6 ___

SIDL_C_INLINE_DECL int32_t bHYPRE_Euclid_SetLogging (bHYPRE_Euclid self, int32_t level, sidl_BaseInterface *_ex)

(Optional) Set the $logging\ level$, specifying the degree of additional informational data to be accumulated. Does nothing by default (level = 0). Other levels (if any) are implementation-specific. Must be called before Setup and Apply. DEPRECATED use SetIntParameter

_ 6.3.7 __

SIDL_C_INLINE_DECL int32_t bHYPRE_Euclid_SetPrintLevel (bHYPRE_Euclid self, int32_t level, sidl_BaseInterface *_ex)

(Optional) Set the *print level*, specifying the degree of informational data to be printed either to the screen or to a file. Does nothing by default (level=0). Other levels (if any) are implementation-specific. Must be called before Setup and Apply. DEPRECATED use SetIntParameter

6.3.8

SIDL_C_INLINE_DECL int32_t bHYPRE_Euclid_SetCommunicator (bHYPRE_Euclid self, bHYPRE_MPICommunicator mpi_comm, sidl_BaseInterface *_ex)

Set the MPI Communicator. DEPRECATED, use Create:

6.3.9

SIDL_C_INLINE_DECL void **bHYPRE_Euclid_Destroy** (bHYPRE_Euclid self, sidl_BaseInterface *_ex)

The Destroy function doesn't necessarily destroy anything. It is just another name for deleteRef. Thus it decrements the object's reference count. The Babel memory management system will destroy the object if the reference count goes to zero.

6.3.10

RMI connector function for the class. (no addref)

6.4

ParCSR Schwarz Solver

Names

6.4.1 struct bHYPRE_Schwarz__object

 $_{\mathbf{ex}}$

Constructor function for the class

bHYPRE_Schwarz

	bHYPRE_SchwarzcreateRemote (const char * url, sidl_BaseInterface *_ex) RMI constructor function for the class	
	bHYPRE_Schwarz bHYPRE_SchwarzwrapObj (void * data, sidl_BaseInterface *_ex) Wraps up the private data struct pointer (struct bHYPRE_Schwarzdata) passed in rather than running the constructor	
6.4.2	bHYPRE_Schwarz bHYPRE_Schwarzconnect (const char *, sidl_BaseInterface *_ex) RMI connector function for the class	148
	bHYPRE_Schwarz bHYPRE_Schwarz_Create (bHYPRE_IJParCSRMatrix A, sidl_BaseInterface *_ex) Method: Create[]	
6.4.3	SIDL_C_INLINE_DECL int32_t bHYPRE_Schwarz_SetOperator (bHYPRE_Schwarz self, bHYPRE_Operator A, sidl_BaseInterface *_ex)	
	Set the operator for the linear system being solved	148
6.4.4	SIDL_C_INLINE_DECL int32_t bHYPRE_Schwarz_SetTolerance (bHYPRE_Schwarz self, double tolerance, sidl_BaseInterface *_ex)	
	(Optional) Set the convergence tolerance	149
6.4.5	SIDL_C_INLINE_DECL int32_t bHYPRE_Schwarz_SetMaxIterations (bHYPRE_Schwarz self, int32_t max_iterations, sidl_BaseInterface *_ex)	
	(Optional) Set maximum number of iterations	149
6.4.6	SIDL_C_INLINE_DECL int32_t bHYPRE_Schwarz_SetLogging (bHYPRE_Schwarz self, int32_t level,	
	(Optional) Set the logging level, specifying the degree of additional informational data to be accumulated	149
6.4.7	SIDL_C_INLINE_DECL int32_t bHYPRE_Schwarz_SetPrintLevel (bHYPRE_Schwarz self, int32_t level, sidl_BaseInterface *_ex)	
	(Optional) Set the print level, specifying the degree of informational data to be printed either to the screen or to a file	149
	SIDL_C_INLINE_DECL int32_t bHYPRE_Schwarz_GetNumIterations (bHYPRE_Schwarz self, int32_t* num_iterations, sidl_BaseInterface *_ex)	
	(Optional) Return the number of iterations taken	
	SIDL_C_INLINE_DECL_int32_t bHYPRE_Schwarz_GetRelResidualNorm (bHYPRE_Schwarz self, double* norm, sidl_BaseInterface *_ex)	
	(Optional) Return the norm of the relative residual	
6.4.8	SIDL_C_INLINE_DECL int32_t	

```
bHYPRE_Schwarz_SetCommunicator ( bHYPRE_Schwarz self.
                                                   bHYPRE_MPICommunicator
                                                   mpi_comm, sidl_BaseInterface *_ex)
                  Set the MPI Communicator .....
                                                                                           150
6.4.9
            SIDL_C_INLINE_DECL void
            bHYPRE_Schwarz_Destroy ( bHYPRE_Schwarz self,
                                         sidl_BaseInterface *_ex)
                   The Destroy function doesn't necessarily destroy anything .....
                                                                                           150
            SIDL_C_INLINE_DECL int32_t
            bHYPRE_Schwarz_SetIntParameter ( bHYPRE_Schwarz self,
                                                 const char* name, int32_t value,
                                                 sidl_BaseInterface *_ex)
                  Set the int parameter associated with name
            SIDL_C_INLINE_DECL int32_t
            bHYPRE_Schwarz_SetDoubleParameter ( bHYPRE_Schwarz self,
                                                      const char* name, double value,
                                                      sidl_BaseInterface *_ex)
                   Set the double parameter associated with name
            SIDL_C_INLINE_DECL int32_t
            bHYPRE_Schwarz_SetStringParameter ( bHYPRE_Schwarz self,
                                                     const char* name,
                                                     const char* value,
                                                     sidl_BaseInterface *_ex)
                  Set the string parameter associated with name
            SIDL_C_INLINE_DECL_int32_t
            b HYPRE\_Schwarz\_SetIntArray1Parameter \ ( \ bHYPRE\_Schwarz \ self,
                                                        const char* name,
                                                         int32_t* value,
                                                        int32_t nvalues,
                                                        sidl_BaseInterface *_ex)
                  Set the int 1-D array parameter associated with name
            SIDL_C_INLINE_DECL int32_t
            bHYPRE_Schwarz_SetIntArray2Parameter ( bHYPRE_Schwarz self,
                                                        const char* name,
                                                         struct sidl_int_array* value,
                                                         sidl_BaseInterface *_ex)
                  Set the int 2-D array parameter associated with name
            SIDL_C_INLINE_DECL_int32_t
            bHYPRE_Schwarz_SetDoubleArray1Parameter ( bHYPRE_Schwarz self,
                                                             const char* name.
                                                             double* value,
                                                             int32_t nvalues,
                                                             sidl_BaseInterface *_ex)
                  Set the double 1-D array parameter associated with name
            SIDL_C_INLINE_DECL int32_t
```

```
bHYPRE_Schwarz_SetDoubleArray2Parameter ( bHYPRE_Schwarz self,
                                                  const char* name, struct
                                                  sidl_double_array* value,
                                                  sidl_BaseInterface *_ex)
      Set the double 2-D array parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_Schwarz_GetIntValue ( bHYPRE_Schwarz self, const char* name,
                                  int32_t* value, sidl_BaseInterface *_ex)
      Set the int parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_Schwarz_GetDoubleValue ( bHYPRE_Schwarz self,
                                      const char* name, double* value,
                                      sidl_BaseInterface *_ex)
      Get the double parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_Schwarz_Setup ( bHYPRE_Schwarz self, bHYPRE_Vector b,
                           bHYPRE_Vector x, sidl_BaseInterface *_ex)
      (Optional) Do any preprocessing that may be necessary in order to execute
      Apply
SIDL_C_INLINE_DECL int32_t
bHYPRE_Schwarz_Apply ( bHYPRE_Schwarz self, bHYPRE_Vector b,
                           bHYPRE_Vector* x, sidl_BaseInterface *_ex)
      Apply the operator to b, returning x
SIDL_C_INLINE_DECL int32_t
bHYPRE_Schwarz_ApplyAdjoint ( bHYPRE_Schwarz self,
                                   bHYPRE_Vector b, bHYPRE_Vector* x,
                                   sidl_BaseInterface *_ex)
      Apply the adjoint of the operator to b, returning x
_{\mathbf{ex}}
      Cast method for interface and class type conversions
void*
bHYPRE_Schwarz__cast2 (void* obj, const char* type,
                           sidl_BaseInterface *_ex)
      String cast method for interface and class type conversions
SIDL_C_INLINE_DECL_void
bHYPRE_Schwarz_exec ( bHYPRE_Schwarz self, const char* methodName,
                          sidl_rmi_Call inArgs, sidl_rmi_Return outArgs,
                          sidl_BaseInterface *_ex)
      Select and execute a method by name
SIDL_C_INLINE_DECL char*
bHYPRE_Schwarz_getURL ( bHYPRE_Schwarz self,
                              sidl_BaseInterface *_ex)
      Get the URL of the Implementation of this object (for RMI)
SIDL_C_INLINE_DECL_void
bHYPRE_Schwarz_raddRef ( bHYPRE_Schwarz self,
                              sidl_BaseInterface *_ex)
      On a remote object, addrefs the remote instance
```

SIDL_C_INLINE_DECL sidl_bool

 $\mathbf{bHYPRE_Schwarz__isRemote} \ (\ \ \mathbf{bHYPRE_Schwarz} \ \mathbf{self},$

sidl_BaseInterface *_ex)

TRUE if this object is remote, false if local

sidl_bool

bHYPRE_Schwarz_isLocal (bHYPRE_Schwarz self, sidl_BaseInterface *_ex)

TRUE if this object is remote, false if local

 $**_ex$

Cast method for interface and class type conversions

6.4.10 ******_**ex**

6.4.1

struct bHYPRE_Schwarz__object

Symbol "bHYPRE.Schwarz" (version 1.0.0)

Objects of this type can be cast to Solver objects using the __cast methods.

RDF: Documentation goes here.

Schwarz requires an IJParCSR matrix

6.4.2

bHYPRE_Schwarz

bHYPRE_Schwarz__connect (const char *, sidl_BaseInterface *_ex)

RMI connector function for the class.(addrefs)

6.4.3

SIDL_C_INLINE_DECL int32_t

bHYPRE_Schwarz_SetOperator (bHYPRE_Schwarz self, bHYPRE_Operator A, sidl_BaseInterface *_ex)

Set the operator for the linear system being solved. DEPRECATED. use Create

6.4.4

SIDL_C_INLINE_DECL int32_t bHYPRE_Schwarz_SetTolerance (bHYPRE_Schwarz self, double tolerance, sidl_BaseInterface *_ex)

(Optional) Set the convergence tolerance. DEPRECATED. use SetDoubleParameter

$_$ 6.4.5 $_$

SIDL_C_INLINE_DECL int32_t bHYPRE_Schwarz_SetMaxIterations (bHYPRE_Schwarz self, int32_t max_iterations, sidl_BaseInterface *_ex)

(Optional) Set maximum number of iterations. DEPRECATED use SetIntParameter

_ 6.4.6 _____

SIDL_C_INLINE_DECL int32_t bHYPRE_Schwarz_SetLogging (bHYPRE_Schwarz self, int32_t level, sidl_BaseInterface *_ex)

(Optional) Set the *logging level*, specifying the degree of additional informational data to be accumulated. Does nothing by default (level = 0). Other levels (if any) are implementation-specific. Must be called before Setup and Apply. DEPRECATED use SetIntParameter

_ 6.4.7 ____

SIDL_C_INLINE_DECL int32_t bHYPRE_Schwarz_SetPrintLevel (bHYPRE_Schwarz self, int32_t level, sidl_BaseInterface *_ex)

(Optional) Set the *print level*, specifying the degree of informational data to be printed either to the screen or to a file. Does nothing by default (level=0). Other levels (if any) are implementation-specific. Must be called before Setup and Apply. DEPRECATED use SetIntParameter

_ 6.4.8 _

SIDL_C_INLINE_DECL int32_t bHYPRE_Schwarz_SetCommunicator (bHYPRE_Schwarz self, bHYPRE_MPICommunicator mpi_comm, sidl_BaseInterface *_ex)

Set the MPI Communicator. DEPRECATED, use Create:

 $_$ 6.4.9 $_$

SIDL_C_INLINE_DECL_void bHYPRE_Schwarz_Destroy (bHYPRE_Schwarz_self, sidl_BaseInterface *_ex)

The Destroy function doesn't necessarily destroy anything. It is just another name for deleteRef. Thus it decrements the object's reference count. The Babel memory management system will destroy the object if the reference count goes to zero.

_ 6.4.10 __

RMI connector function for the class. (no addref)

__ 6.5 _____

ParCSR ParaSails Solver

Names

6.5.1 struct bHYPRE_ParaSails_object

	Symbol "bHYPRE	154
	_ex Constructor function for the class	
	bHYPRE_ParaSails bHYPRE_ParaSailscreateRemote (const char * url, sidl_BaseInterface *_ex)	
	RMI constructor function for the class	
	bHYPRE_ParaSails bHYPRE_ParaSailswrapObj (void * data, sidl_BaseInterface *_ex) Wraps up the private data struct pointer (struct bHYPRE_ParaSailsdata) passed in rather than running the constructor	
6.5.2	bHYPRE_ParaSails bHYPRE_ParaSailsconnect (const char *, sidl_BaseInterface *_ex) RMI connector function for the class	154
	bHYPRE_ParaSails bHYPRE_ParaSails_Create (bHYPRE_MPICommunicator mpi_comm, bHYPRE_IJParCSRMatrix A, sidl_BaseInterface *_ex)	
	Method: Create[]	
6.5.3	SIDL_C_INLINE_DECL int32_t bHYPRE_ParaSails_SetOperator (bHYPRE_ParaSails self, bHYPRE_Operator A, sidl_BaseInterface *_ex)	
	Set the operator for the linear system being solved	155
6.5.4	SIDL_C_INLINE_DECL int32_t bHYPRE_ParaSails_SetTolerance (bHYPRE_ParaSails self,	155
6.5.5	SIDL_C_INLINE_DECL int32_t bHYPRE_ParaSails_SetMaxIterations (bHYPRE_ParaSails self,	
	(Optional) Set maximum number of iterations	155
6.5.6	SIDL_C_INLINE_DECL int32_t bHYPRE_ParaSails_SetLogging (bHYPRE_ParaSails self, int32_t level,	
	(Optional) Set the logging level, specifying the degree of additional informational data to be accumulated	155
6.5.7	SIDL_C_INLINE_DECL int32_t bHYPRE_ParaSails_SetPrintLevel (bHYPRE_ParaSails self, int32_t level, sidl_BaseInterface *_ex)	
	(Optional) Set the print level, specifying the degree of informational data to be printed either to the screen or to a file	156
	SIDL_C_INLINE_DECL int32_t	

```
bHYPRE_ParaSails_GetNumIterations ( bHYPRE_ParaSails self.
                                                     int32_t* num_iterations,
                                                     sidl_BaseInterface *_ex)
                   (Optional) Return the number of iterations taken
            SIDL_C_INLINE_DECL int32_t
            bHYPRE_ParaSails_GetRelResidualNorm ( bHYPRE_ParaSails self,
                                                        double* norm,
                                                        sidl_BaseInterface *_ex)
                   (Optional) Return the norm of the relative residual
            SIDL_C_INLINE_DECL int32_t
6.5.8
            bHYPRE_ParaSails_SetCommunicator ( bHYPRE_ParaSails self,
                                                    bHYPRE_MPICommunicator
                                                    mpi_comm, sidl_BaseInterface *_ex)
                  Set the MPI Communicator .....
                                                                                           156
            SIDL_C_INLINE_DECL void
6.5.9
            bHYPRE_ParaSails_Destroy ( bHYPRE_ParaSails self,
                                          sidl_BaseInterface *_ex)
                   The Destroy function doesn't necessarily destroy anything .....
                                                                                           156
            SIDL_C_INLINE_DECL_int32_t
            bHYPRE_ParaSails_SetIntParameter ( bHYPRE_ParaSails self,
                                                   const char* name, int32_t value,
                                                   sidl_BaseInterface *_ex)
                  Set the int parameter associated with name
            SIDL_C_INLINE_DECL int32_t
            bHYPRE_ParaSails_SetDoubleParameter ( bHYPRE_ParaSails self,
                                                       const char* name, double value,
                                                       sidl_BaseInterface *_ex)
                  Set the double parameter associated with name
            SIDL_C_INLINE_DECL int32_t
            bHYPRE_ParaSails_SetStringParameter ( bHYPRE_ParaSails self,
                                                      const char* name.
                                                      const char* value,
                                                      sidl_BaseInterface *_ex)
                  Set the string parameter associated with name
            SIDL_C_INLINE_DECL int32_t
            bHYPRE_ParaSails_SetIntArray1Parameter ( bHYPRE_ParaSails self,
                                                          const char* name,
                                                          int32_t* value,
                                                          int32_t nvalues.
                                                          sidl_BaseInterface *_ex)
                  Set the int 1-D array parameter associated with name
            SIDL_C_INLINE_DECL int32_t
            bHYPRE_ParaSails_SetIntArray2Parameter ( bHYPRE_ParaSails self,
                                                          const char* name,
                                                          struct sidl_int_array* value,
                                                          sidl_BaseInterface *_ex)
                  Set the int 2-D array parameter associated with name
            SIDL_C_INLINE_DECL int32_t
```

```
bHYPRE_ParaSails_SetDoubleArray1Parameter ( bHYPRE_ParaSails self,
                                                   const char* name.
                                                   double* value,
                                                   int32_t nvalues,
                                                   sidl_BaseInterface *_ex)
      Set the double 1-D array parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_ParaSails_SetDoubleArray2Parameter ( bHYPRE_ParaSails self,
                                                   const char* name,
                                                   struct sidl_double_array*
                                                   sidl_BaseInterface *_ex)
      Set the double 2-D array parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_ParaSails_GetIntValue ( bHYPRE_ParaSails self,
                                   const char* name, int32_t* value,
                                   sidl_BaseInterface *_ex)
      Set the int parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_ParaSails_GetDoubleValue ( bHYPRE_ParaSails self,
                                        const char* name, double* value,
                                        sidl_BaseInterface *_ex)
       Get the double parameter associated with name
SIDL_C_INLINE_DECL_int32_t
bHYPRE_ParaSails_Setup ( bHYPRE_ParaSails self, bHYPRE_Vector b,
                            bHYPRE_Vector x, sidl_BaseInterface *_ex)
       (Optional) Do any preprocessing that may be necessary in order to execute
      Apply
SIDL_C_INLINE_DECL int32_t
bHYPRE_ParaSails_Apply ( bHYPRE_ParaSails self, bHYPRE_Vector b,
                             bHYPRE_Vector* x, sidl_BaseInterface *_ex)
      Apply the operator to b, returning x
SIDL_C_INLINE_DECL int32_t
bHYPRE_ParaSails_ApplyAdjoint ( bHYPRE_ParaSails self,
                                     bHYPRE_Vector b,
                                     bHYPRE_Vector* x,
                                     sidl_BaseInterface *_ex)
      Apply the adjoint of the operator to b, returning x
_ex
       Cast method for interface and class type conversions
void*
bHYPRE_ParaSails__cast2 (void* obj, const char* type,
                             sidl_BaseInterface *_ex)
       String cast method for interface and class type conversions
SIDL_C_INLINE_DECL void
```

```
bHYPRE_ParaSails_exec ( bHYPRE_ParaSails self,
                                        const char* methodName, sidl_rmi_Call inArgs,
                                        sidl_rmi_Return outArgs, sidl_BaseInterface *_ex)
                   Select and execute a method by name
            SIDL_C_INLINE_DECL char*
            bHYPRE_ParaSails__getURL ( bHYPRE_ParaSails self,
                                            sidl_BaseInterface *_ex)
                   Get the URL of the Implementation of this object (for RMI)
            SIDL_C_INLINE_DECL_void
            bHYPRE_ParaSails__raddRef ( bHYPRE_ParaSails self,
                                            sidl_BaseInterface *_ex)
                   On a remote object, addrefs the remote instance
            SIDL_C_INLINE_DECL sidl_bool
            bHYPRE_ParaSails__isRemote ( bHYPRE_ParaSails self,
                                             sidl_BaseInterface *_ex)
                   TRUE if this object is remote, false if local
            sidl_bool
            bHYPRE_ParaSails__isLocal ( bHYPRE_ParaSails self,
                                          sidl_BaseInterface *_ex)
                   TRUE if this object is remote, false if local
            **_ex
                   Cast method for interface and class type conversions
            **_ex
6.5.10
                   RMI connector function for the class ......
                                                                                             156
```

6.5.1

struct bHYPRE_ParaSails_object

Symbol "bHYPRE.ParaSails" (version 1.0.0)

Objects of this type can be cast to Solver objects using the __cast methods.

RDF: Documentation goes here.

ParaSails requires an IJParCSR matrix

6.5.2

bHYPRE_ParaSails
bHYPRE_ParaSails__connect (const char *, sidl_BaseInterface *_ex)

RMI connector function for the class.(addrefs)

__ 6.5.3 _____

SIDL_C_INLINE_DECL int32_t bHYPRE_ParaSails_SetOperator (bHYPRE_ParaSails self, bHYPRE_Operator A, sidl_BaseInterface *_ex)

Set the operator for the linear system being solved. DEPRECATED. use Create

6.5.4

(Optional) Set the convergence tolerance. DEPRECATED. use SetDoubleParameter

6.5.5

SIDL_C_INLINE_DECL int32_t bHYPRE_ParaSails_SetMaxIterations (bHYPRE_ParaSails self, int32_t max_iterations, sidl_BaseInterface *_ex)

(Optional) Set maximum number of iterations. DEPRECATED use SetIntParameter

_ 6.5.6 ____

SIDL_C_INLINE_DECL int32_t bHYPRE_ParaSails_SetLogging (bHYPRE_ParaSails self, int32_t level, sidl_BaseInterface *_ex)

(Optional) Set the $logging\ level$, specifying the degree of additional informational data to be accumulated. Does nothing by default (level = 0). Other levels (if any) are implementation-specific. Must be called before Setup and Apply. DEPRECATED use SetIntParameter

6.5.7

SIDL_C_INLINE_DECL int32_t bHYPRE_ParaSails_SetPrintLevel (bHYPRE_ParaSails self, int32_t level, sidl_BaseInterface *_ex)

(Optional) Set the *print level*, specifying the degree of informational data to be printed either to the screen or to a file. Does nothing by default (level=0). Other levels (if any) are implementation-specific. Must be called before Setup and Apply. DEPRECATED use SetIntParameter

___ 6.5.8 _____

SIDL_C_INLINE_DECL int32_t bHYPRE_ParaSails_SetCommunicator (bHYPRE_ParaSails self, bHYPRE_MPICommunicator mpi_comm, sidl_BaseInterface *_ex)

Set the MPI Communicator. DEPRECATED, use Create:

_ 6.5.9 _

SIDL_C_INLINE_DECL_void bHYPRE_ParaSails_Destroy (bHYPRE_ParaSails self, sidl_BaseInterface *_ex)

The Destroy function doesn't necessarily destroy anything. It is just another name for deleteRef. Thus it decrements the object's reference count. The Babel memory management system will destroy the object if the reference count goes to zero.

6.5.10

struct bHYPRE_ParaSails__object* bHYPRE_ParaSails__connectI const char * url sidl_bool ar struct sidl_BaseInterface__object **_ex

RMI connector function for the class. (no addref)

6 6

ParCSR Pilut Solver

Names		
6.6.1	struct bHYPRE_Pilutobject Symbol "bHYPRE	160
	_ex Constructor function for the class	
	bHYPRE_Pilut bHYPRE_PilutcreateRemote (const char * url, sidl_BaseInterface *_ex) RMI constructor function for the class	
	bHYPRE_Pilut bHYPRE_PilutwrapObj (void * data, sidl_BaseInterface *_ex) Wraps up the private data struct pointer (struct bHYPRE_Pilut_data) passed in rather than running the constructor	
6.6.2	bHYPRE_Pilut bHYPRE_Pilutconnect (const char *, sidl_BaseInterface *_ex) RMI connector function for the class	160
	bHYPRE_Pilut bHYPRE_MPICommunicator mpi_comm, bHYPRE_Operator A, sidl_BaseInterface *_ex)	
	Method: Create[]	
6.6.3	SIDL_C_INLINE_DECL int32_t bHYPRE_Pilut_SetOperator (bHYPRE_Pilut self, bHYPRE_Operator A, sidl_BaseInterface *_ex)	
	Set the operator for the linear system being solved	160
6.6.4	SIDL_C_INLINE_DECL int32_t bHYPRE_Pilut_SetTolerance (bHYPRE_Pilut self, double tolerance, sidl_BaseInterface *_ex)	
	(Optional) Set the convergence tolerance	161
6.6.5	SIDL_C_INLINE_DECL int32_t bHYPRE_Pilut_SetMaxIterations (bHYPRE_Pilut self,	
	(Optional) Set maximum number of iterations	161
6.6.6	SIDL_C_INLINE_DECL int32_t bHYPRE_Pilut_SetLogging (bHYPRE_Pilut self, int32_t level,	
	(Optional) Set the logging level, specifying the degree of additional informational data to be accumulated	161
6.6.7	SIDL_C_INLINE_DECL int32_t bHYPRE_Pilut_SetPrintLevel (bHYPRE_Pilut self, int32_t level, sidl_BaseInterface *_ex)	
	(Optional) Set the print level, specifying the degree of informational data to be printed either to the screen or to a file	161
	SIDL_C_INLINE_DECL int32_t	

	bHYPRE_Pilut_GetNumIterations (bHYPRE_Pilut self, int32_t* num_iterations, sidl_BaseInterface *_ex)	
	(Optional) Return the number of iterations taken	
	SIDL_C_INLINE_DECL int32_t bHYPRE_Pilut_GetRelResidualNorm (bHYPRE_Pilut self, double* norm,	
6.6.8	SIDL_C_INLINE_DECL int32_t bHYPRE_Pilut_SetCommunicator (bHYPRE_Pilut self, bHYPRE_MPICommunicator mpi_comm, sidl_BaseInterface *_ex)	
	Set the MPI Communicator	162
6.6.9	SIDL_C_INLINE_DECL void bHYPRE_Pilut_Destroy (bHYPRE_Pilut self, sidl_BaseInterface *_ex) The Destroy function doesn't necessarily destroy anything	162
	SIDL_C_INLINE_DECL int32_t bHYPRE_Pilut_SetIntParameter (bHYPRE_Pilut self, const char* name,	
	SIDL_C_INLINE_DECL int32_t bHYPRE_Pilut_SetDoubleParameter (bHYPRE_Pilut self,	
	SIDL_C_INLINE_DECL int32_t bHYPRE_Pilut_SetStringParameter (bHYPRE_Pilut self,	
	Set the string parameter associated with name	
	SIDL_C_INLINE_DECL int32_t bHYPRE_Pilut_SetIntArray1Parameter (bHYPRE_Pilut self,	
	Set the int 1-D array parameter associated with name	
	SIDL_C_INLINE_DECL_int32_t bHYPRE_Pilut_SetIntArray2Parameter (bHYPRE_Pilut self,	
	sidl_BaseInterface *_ex) Set the int 2-D array parameter associated with name	

 $SIDL_C_INLINE_DECL \ int 32_t$

```
bHYPRE_Pilut_SetDoubleArray1Parameter ( bHYPRE_Pilut self.
                                               const char* name,
                                               double* value,
                                               int32_t nvalues,
                                               sidl_BaseInterface *_ex)
      Set the double 1-D array parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_Pilut_SetDoubleArray2Parameter ( bHYPRE_Pilut self,
                                               const char* name, struct
                                               sidl_double_array* value,
                                               sidl_BaseInterface *_ex)
      Set the double 2-D array parameter associated with name
SIDL_C_INLINE_DECL_int32_t
bHYPRE_Pilut_GetIntValue ( bHYPRE_Pilut self, const char* name,
                               int32_t* value, sidl_BaseInterface *_ex)
      Set the int parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_Pilut_GetDoubleValue ( bHYPRE_Pilut self, const char* name,
                                   double* value, sidl_BaseInterface *_ex)
       Get the double parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_Pilut_Setup ( bHYPRE_Pilut self, bHYPRE_Vector b,
                        bHYPRE_Vector x, sidl_BaseInterface *_ex)
       (Optional) Do any preprocessing that may be necessary in order to execute
      Apply
SIDL_C_INLINE_DECL int32_t
bHYPRE_Pilut_Apply ( bHYPRE_Pilut self, bHYPRE_Vector b,
                        bHYPRE_Vector* x, sidl_BaseInterface *_ex)
      Apply the operator to b, returning x
SIDL_C_INLINE_DECL int32_t
bHYPRE_Pilut_ApplyAdjoint ( bHYPRE_Pilut self, bHYPRE_Vector b,
                                bHYPRE_Vector* x, sidl_BaseInterface *_ex)
       Apply the adjoint of the operator to b, returning x
_{-}ex
       Cast method for interface and class type conversions
void*
bHYPRE_Pilut__cast2 (void* obj, const char* type, sidl_BaseInterface *_ex)
       String cast method for interface and class type conversions
SIDL_C_INLINE_DECL void
bHYPRE_Pilut__exec ( bHYPRE_Pilut self, const char* methodName,
                       sidl_rmi_Call inArgs, sidl_rmi_Return outArgs,
                       sidl_BaseInterface *_ex)
      Select and execute a method by name
SIDL_C_INLINE_DECL_char*
bHYPRE_Pilut__getURL ( bHYPRE_Pilut self, sidl_BaseInterface *_ex)
       Get the URL of the Implementation of this object (for RMI)
SIDL_C_INLINE_DECL_void
```

6.6.1 _

struct bHYPRE_Pilut_object

Symbol "bHYPRE.Pilut" (version 1.0.0)

Objects of this type can be cast to Solver objects using the __cast methods.

RDF: Documentation goes here.

Pilut has not been implemented yet.

_ 6.6.2 _

bHYPRE_Pilut bHYPRE_Pilut_connect (const char *, sidl_BaseInterface *_ex)

RMI connector function for the class.(addrefs)

_ 6.6.3 _

SIDL_C_INLINE_DECL int32_t bHYPRE_Pilut_SetOperator (bHYPRE_Pilut_self, bHYPRE_Operator A, sidl_BaseInterface *_ex)

Set the operator for the linear system being solved. DEPRECATED. use Create

6.6.4

(Optional) Set the convergence tolerance. DEPRECATED. use SetDoubleParameter

 $_$ 6.6.5 $_$

SIDL_C_INLINE_DECL int32_t bHYPRE_Pilut_SetMaxIterations (bHYPRE_Pilut self, int32_t max_iterations, sidl_BaseInterface *_ex)

(Optional) Set maximum number of iterations. DEPRECATED use SetIntParameter

_ 6.6.6 _____

SIDL_C_INLINE_DECL int32_t bHYPRE_Pilut_SetLogging (bHYPRE_Pilut self, int32_t level, sidl_BaseInterface *_ex)

(Optional) Set the *logging level*, specifying the degree of additional informational data to be accumulated. Does nothing by default (level = 0). Other levels (if any) are implementation-specific. Must be called before Setup and Apply. DEPRECATED use SetIntParameter

_ 6.6.7 _____

SIDL_C_INLINE_DECL int32_t bHYPRE_Pilut_SetPrintLevel (bHYPRE_Pilut self, int32_t level, sidl_BaseInterface *_ex)

(Optional) Set the *print level*, specifying the degree of informational data to be printed either to the screen or to a file. Does nothing by default (level=0). Other levels (if any) are implementation-specific. Must be called before Setup and Apply. DEPRECATED use SetIntParameter

6.6.8

SIDL_C_INLINE_DECL int32_t bHYPRE_Pilut_SetCommunicator (bHYPRE_Pilut_self, bHYPRE_MPICommunicator mpi_comm, sidl_BaseInterface *_ex)

Set the MPI Communicator. DEPRECATED, use Create:

6.6.9

SIDL_C_INLINE_DECL void bHYPRE_Pilut_Destroy (bHYPRE_Pilut self, sidl_BaseInterface *_ex)

The Destroy function doesn't necessarily destroy anything. It is just another name for deleteRef. Thus it decrements the object's reference count. The Babel memory management system will destroy the object if the reference count goes to zero.

 $6.6.10_{-}$

struct bHYPRE_Pilut_object* bHYPRE_Pilut_connectI const char * url sidl_bool ar struct sidl_BaseInterface_object
**_ex

RMI connector function for the class. (no addref)

. 7

Structured Matrix Solvers

Names		
7.1	StructDiagScale Solver	
		163
7.2	Struct Jacobi Solver	
		170
7.3	Struct PFMG Solver	
		177
7.4	Struct SMG Solver	
		183

These solvers use structured matrix/vector storage schemes.

_ 7.1 _

${\bf Struct Diag Scale\ Solver}$

Names		
7.1.1	struct bHYPRE_StructDiagScaleobject Symbol "bHYPRE	167
	_ex Constructor function for the class	
	bHYPRE_StructDiagScale bHYPRE_StructDiagScalecreateRemote (const char * url,	
	bHYPRE_StructDiagScale bHYPRE_StructDiagScalewrapObj (void * data, sidl_BaseInterface *_ex) Wraps up the private data struct pointer (struct bHYPRE_StructDiagScaledata) passed in rather than running the constructor	
7.1.2	bHYPRE_StructDiagScale bHYPRE_StructDiagScaleconnect (const char *, sidl_BaseInterface *_ex) RMI connector function for the class	168
	bHYPRE_StructDiagScale	

	bHYPRE_StructDiagScale_Create (bHYPRE_MPICommunicator	
	mpi_comm, bHYPRE_StructMatrix A,	
	sidl_BaseInterface *_ex)	
	Method: Create[]	
7.1.3	SIDL_C_INLINE_DECL int32_t	
	bHYPRE_StructDiagScale_SetOperator (bHYPRE_StructDiagScale self,	
	bHYPRE_Operator A,	
	sidl_BaseInterface *_ex) Set the operator for the linear system being solved	168
714	· · · · · · · · · · · · · · · · · · ·	100
7.1.4	SIDL_C_INLINE_DECL int32_t bHYPRE_StructDiagScale_SetTolerance (bHYPRE_StructDiagScale self,	
	double tolerance,	
	sidl_BaseInterface *_ex)	
	(Optional) Set the convergence tolerance	168
7.1.5	SIDL_C_INLINE_DECL_int32_t	
1.1.0	bHYPRE_StructDiagScale_SetMaxIterations (bHYPRE_StructDiagScale	
	self, int32_t max_iterations,	
	sidl_BaseInterface *_ex)	
	(Optional) Set maximum number of iterations	168
7.1.6	SIDL_C_INLINE_DECL int32_t	
	bHYPRE_StructDiagScale_SetLogging (bHYPRE_StructDiagScale self,	
	int32_t level,	
	sidl_BaseInterface *_ex) (Optional) Set the logging level, specifying the degree of additional informa-	
	tional data to be accumulated	169
717		100
7.1.7	SIDL_C_INLINE_DECL int32_t bHYPRE_StructDiagScale_SetPrintLevel (bHYPRE_StructDiagScale self,	
	int32_t level,	
	sidl_BaseInterface *_ex)	
	(Optional) Set the print level, specifying the degree of informational data	
	to be printed either to the screen or to a file	169
	SIDL_C_INLINE_DECL int32_t	
	$b HYPRE_StructDiagScale_GetNumIterations~(~~bHYPRE_StructDiagScale$	
	$\operatorname{self},$	
	int32_t* num_iterations,	
	sidl_BaseInterface *_ex) (Optional) Return the number of iterations taken	
	, -	
	SIDL_C_INLINE_DECL int32_t bHYPRE_StructDiagScale_GetRelResidualNorm (
	bHYPRE_StructDiagScale	
	self, double* norm,	
	sidl_BaseInterface *_ex)	
	(Optional) Return the norm of the relative residual	
7.1.8	SIDL_C_INLINE_DECL int32_t	

```
bHYPRE_StructDiagScale_SetCommunicator ( bHYPRE_StructDiagScale
                                                           self.
                                                           bHYPRE_MPICommunicator
                                                           mpi_comm,
                                                           sidl_BaseInterface *_ex)
                   Set the MPI Communicator .....
                                                                                            169
7.1.9
            SIDL_C_INLINE_DECL void
            bHYPRE_StructDiagScale_Destroy ( bHYPRE_StructDiagScale self,
                                                 sidl_BaseInterface *_ex)
                   The Destroy function doesn't necessarily destroy anything .....
                                                                                            169
            SIDL_C_INLINE_DECL int32_t
            bHYPRE\_StructDiagScale\_SetIntParameter ( bHYPRE\_StructDiagScale
                                                          self, const char* name,
                                                          int32_t value,
                                                          sidl_BaseInterface *_ex)
                   Set the int parameter associated with name
            SIDL_C_INLINE_DECL int32_t
            bHYPRE_StructDiagScale_SetDoubleParameter (
                                                              bHYPRE\_StructDiagScale
                                                              self, const char* name,
                                                              double value,
                                                              sidl_BaseInterface *_ex)
                   Set the double parameter associated with name
            SIDL_C_INLINE_DECL int32_t
            {\bf bHYPRE\_StructDiagScale\_SetStringParameter}\ (
                                                             b HYPRE\_StructDiagScale
                                                             self, const char* name,
                                                             const char* value,
                                                             sidl_BaseInterface *_ex)
                   Set the string parameter associated with name
            SIDL_C_INLINE_DECL int32_t
            bHYPRE_StructDiagScale_SetIntArray1Parameter (
                                                                 bHYPRE_StructDiagScale
                                                                 self,
                                                                 const char* name,
                                                                 int32_t* value,
                                                                 int32_t nvalues.
                                                                 sidl\_BaseInterface
                                                                 *_ex)
                   Set the int 1-D array parameter associated with name
            SIDL_C_INLINE_DECL int32_t
            bHYPRE_StructDiagScale_SetIntArray2Parameter (
                                                                 bHYPRE_StructDiagScale
                                                                 self,
                                                                 const char* name,
                                                                 struct sidl_int_array*
                                                                 value.
                                                                 sidl_BaseInterface
                                                                 *_ex)
                   Set the int 2-D array parameter associated with name
            SIDL_C_INLINE_DECL int32_t
```

```
bHYPRE_StructDiagScale_SetDoubleArray1Parameter (
                                                           bHYPRE_StructDiagScale
                                                           self, const char*
                                                           name,
                                                           double* value,
                                                           int32_t nvalues,
                                                           sidl_BaseInterface
                                                           *_ex)
      Set the double 1-D array parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_StructDiagScale_SetDoubleArray2Parameter (
                                                           b HYPRE\_StructDiagScale
                                                           self, const char*
                                                           name, struct
                                                           sidl_double_array*
                                                           value,
                                                           sidl_BaseInterface
                                                           *_ex)
      Set the double 2-D array parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_StructDiagScale_GetIntValue ( bHYPRE_StructDiagScale self,
                                          const char* name, int32_t* value,
                                          sidl_BaseInterface *_ex)
      Set the int parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE\_StructDiagScale\_GetDoubleValue ( bHYPRE\_StructDiagScale
                                               self, const char* name,
                                               double* value.
                                               sidl_BaseInterface *_ex)
       Get the double parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_StructDiagScale_Setup ( bHYPRE_StructDiagScale self,
                                    bHYPRE_Vector b, bHYPRE_Vector x,
                                    sidl_BaseInterface *_ex)
       (Optional) Do any preprocessing that may be necessary in order to execute
      Apply
SIDL_C_INLINE_DECL_int32_t
bHYPRE_StructDiagScale_Apply ( bHYPRE_StructDiagScale self,
                                    bHYPRE_Vector b, bHYPRE_Vector* x,
                                    sidl_BaseInterface *_ex)
      Apply the operator to b, returning x
SIDL_C_INLINE_DECL int32_t
bHYPRE_StructDiagScale_ApplyAdjoint ( bHYPRE_StructDiagScale self,
                                            bHYPRE_Vector b,
                                            bHYPRE_Vector* x,
                                            sidl_BaseInterface *_ex)
      Apply the adjoint of the operator to b, returning x
_{\mathbf{ex}}
       Cast method for interface and class type conversions
void*
```

bHYPRE_StructDiagScale__cast2 (void* obj, const char* type, sidl_BaseInterface *_ex) String cast method for interface and class type conversions SIDL_C_INLINE_DECL_void bHYPRE_StructDiagScale_exec (bHYPRE_StructDiagScale self, const char* methodName, sidl_rmi_Call inArgs, sidl_rmi_Return outArgs, sidl_BaseInterface *_ex) Select and execute a method by name SIDL_C_INLINE_DECL char* bHYPRE_StructDiagScale__getURL (bHYPRE_StructDiagScale self, sidl_BaseInterface *_ex) Get the URL of the Implementation of this object (for RMI) SIDL_C_INLINE_DECL void bHYPRE_StructDiagScale_raddRef (bHYPRE_StructDiagScale self, sidl_BaseInterface *_ex) On a remote object, addrefs the remote instance SIDL_C_INLINE_DECL sidl_bool bHYPRE_StructDiagScale__isRemote (bHYPRE_StructDiagScale self, sidl_BaseInterface *_ex) TRUE if this object is remote, false if local $sidl_bool$ bHYPRE_StructDiagScale__isLocal (bHYPRE_StructDiagScale self, sidl_BaseInterface *_ex) TRUE if this object is remote, false if local **_ex

 $Cast\ method\ for\ interface\ and\ class\ type\ conversions$

7.1.10 ******_**ex**

7.1.1

struct bHYPRE_StructDiagScale__object

Symbol "bHYPRE.StructDiagScale" (version 1.0.0)

Diagonal scaling preconditioner for STruct matrix class.

Objects of this type can be cast to Solver objects using the __cast methods.

7.1.2

bHYPRE_StructDiagScale
bHYPRE_StructDiagScale__connect (const char *, sidl_BaseInterface *_ex)

RMI connector function for the class.(addrefs)

__ 7.1.3 _____

SIDL_C_INLINE_DECL int32_t bHYPRE_StructDiagScale_SetOperator (bHYPRE_StructDiagScale self, bHYPRE_Operator A, sidl_BaseInterface *_ex)

Set the operator for the linear system being solved. DEPRECATED. use Create

7.1.4

SIDL_C_INLINE_DECL int32_t bHYPRE_StructDiagScale_SetTolerance (bHYPRE_StructDiagScale self, double tolerance, sidl_BaseInterface *_ex)

(Optional) Set the convergence tolerance. DEPRECATED. use SetDoubleParameter

7.1.5

SIDL_C_INLINE_DECL int32_t bHYPRE_StructDiagScale_SetMaxIterations (bHYPRE_StructDiagScale self, int32_t max_iterations, sidl_BaseInterface *_ex)

(Optional) Set maximum number of iterations. DEPRECATED use SetIntParameter

7.1.6

SIDL_C_INLINE_DECL int32_t bHYPRE_StructDiagScale_SetLogging (bHYPRE_StructDiagScale self, int32_t level, sidl_BaseInterface *_ex)

(Optional) Set the *logging level*, specifying the degree of additional informational data to be accumulated. Does nothing by default (level = 0). Other levels (if any) are implementation-specific. Must be called before Setup and Apply. DEPRECATED use SetIntParameter

_ 7.1.7 _

SIDL_C_INLINE_DECL int32_t bHYPRE_StructDiagScale_SetPrintLevel (bHYPRE_StructDiagScale self, int32_t level, sidl_BaseInterface *_ex)

(Optional) Set the *print level*, specifying the degree of informational data to be printed either to the screen or to a file. Does nothing by default (level=0). Other levels (if any) are implementation-specific. Must be called before Setup and Apply. DEPRECATED use SetIntParameter

_ 7.1.8 _

SIDL_C_INLINE_DECL int32_t bHYPRE_StructDiagScale_SetCommunicator (bHYPRE_StructDiagScale self, bHYPRE_MPICommunicator mpi_comm, sidl_BaseInterface *_ex)

Set the MPI Communicator. DEPRECATED, use Create:

$_{-}$ 7.1.9 $_{-}$

SIDL_C_INLINE_DECL_void bHYPRE_StructDiagScale_Destroy (bHYPRE_StructDiagScale self, sidl_BaseInterface *_ex)

The Destroy function doesn't necessarily destroy anything. It is just another name for deleteRef. Thus it decrements the object's reference count. The Babel memory management system will destroy the object if the reference count goes to zero.

7.1.10

struct bHYPRE_StructDiagScale__object* bHYPRE_StructDiagScale__connectI const char * url sidl_bool ar struct sidl_BaseInterface__object **_ex

RMI connector function for the class. (no addref)

_ 7.2 _

Struct Jacobi Solver

Names		
7.2.1	struct bHYPRE_StructJacobiobject Symbol "bHYPRE	174
	_ex Constructor function for the class	
	bHYPRE_StructJacobi bHYPRE_StructJacobicreateRemote (const char * url,	
	bHYPRE_StructJacobi bHYPRE_StructJacobiwrapObj (void * data, sidl_BaseInterface *_ex) Wraps up the private data struct pointer (struct bHYPRE_StructJacobi_data) passed in rather than running the con- structor	
7.2.2	bHYPRE_StructJacobi_connect (const char *, sidl_BaseInterface *_ex) RMI connector function for the class	174
	bHYPRE_StructJacobi bHYPRE_MPICommunicator mpi_comm, bHYPRE_StructMatrix A, sidl_BaseInterface *_ex)	
	Method: Create[]	
7.2.3	SIDL_C_INLINE_DECL int32_t bHYPRE_StructJacobi_SetOperator (bHYPRE_StructJacobi self,	
	Set the operator for the linear system being solved	174
7.2.4	SIDL_C_INLINE_DECL_int32_t	

	bHYPRE_StructJacobi_SetTolerance (bHYPRE_StructJacobi self, double tolerance,	
	sidl_BaseInterface *_ex) (Optional) Set the convergence tolerance	175
705	· -	110
7.2.5	SIDL_C_INLINE_DECL int32_t bHYPRE_StructJacobi_SetMaxIterations (bHYPRE_StructJacobi self,	
	(Optional) Set maximum number of iterations	175
7.2.6	SIDL_C_INLINE_DECL int32_t bHYPRE_StructJacobi_SetLogging (bHYPRE_StructJacobi self,	
	(Optional) Set the logging level, specifying the degree of additional informational data to be accumulated	175
7.2.7	SIDL_C_INLINE_DECL int32_t bHYPRE_StructJacobi_SetPrintLevel (bHYPRE_StructJacobi self,	175
	SIDL_C_INLINE_DECL int32_t bHYPRE_StructJacobi_GetNumIterations (bHYPRE_StructJacobi self, int32_t* num_iterations, sidl_BaseInterface *_ex)	
	(Optional) Return the number of iterations taken	
	SIDL_C_INLINE_DECL int32_t bHYPRE_StructJacobi_GetRelResidualNorm (bHYPRE_StructJacobi self, double* norm, sidl_BaseInterface *_ex) (Optional) Return the norm of the relative residual	
7.2.8	SIDL_C_INLINE_DECL int32_t	
1.2.0	bHYPRE_StructJacobi_SetCommunicator (bHYPRE_StructJacobi self, bHYPRE_MPICommunicator mpi_comm, sidl_BaseInterface *_ex)	
	Set the MPI Communicator	176
7.2.9	SIDL_C_INLINE_DECL void bHYPRE_StructJacobi self, sidl_BaseInterface *_ex)	
	The Destroy function doesn't necessarily destroy anything	176
	SIDL_C_INLINE_DECL int32_t bHYPRE_StructJacobi_SetIntParameter (bHYPRE_StructJacobi self,	
	Set the int parameter associated with name	
	SIDL_C_INLINE_DECL int32_t	

```
bHYPRE_StructJacobi_SetDoubleParameter ( bHYPRE_StructJacobi self,
                                                const char* name,
                                                double value,
                                                sidl_BaseInterface *_ex)
      Set the double parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_StructJacobi_SetStringParameter ( bHYPRE_StructJacobi self,
                                               const char* name,
                                               const char* value,
                                               sidl_BaseInterface *_ex)
      Set the string parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_StructJacobi_SetIntArray1Parameter ( bHYPRE_StructJacobi
                                                   self, const char* name,
                                                   int32_t* value,
                                                   int32_t nvalues.
                                                   sidl_BaseInterface *_ex)
      Set the int 1-D array parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_StructJacobi_SetIntArray2Parameter ( bHYPRE_StructJacobi
                                                   self, const char* name,
                                                   struct sidl_int_array*
                                                   value.
                                                   sidl_BaseInterface *_ex)
      Set the int 2-D array parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_StructJacobi_SetDoubleArray1Parameter (
                                                       bHYPRE_StructJacobi
                                                       self.
                                                       const char* name,
                                                       double* value,
                                                       int32_t nvalues,
                                                       sidl\_BaseInterface
                                                       *_ex)
      Set the double 1-D array parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_StructJacobi_SetDoubleArray2Parameter (
                                                       bHYPRE\_StructJacobi
                                                       const char* name,
                                                       struct
                                                       sidl_double_array*
                                                       value.
                                                       sidl_BaseInterface
                                                       *_ex)
      Set the double 2-D array parameter associated with name
SIDL_C_INLINE_DECL int32_t
```

```
bHYPRE_StructJacobi_GetIntValue ( bHYPRE_StructJacobi self.
                                       const char* name, int32_t* value,
                                       sidl_BaseInterface *_ex)
      Set the int parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_StructJacobi_GetDoubleValue ( bHYPRE_StructJacobi self,
                                           const char* name, double* value,
                                           sidl_BaseInterface *_ex)
       Get the double parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_StructJacobi_Setup ( bHYPRE_StructJacobi self,
                                bHYPRE_Vector b, bHYPRE_Vector x,
                                sidl_BaseInterface *_ex)
       (Optional) Do any preprocessing that may be necessary in order to execute
      Apply
SIDL_C_INLINE_DECL int32_t
bHYPRE_StructJacobi_Apply ( bHYPRE_StructJacobi self,
                                bHYPRE_Vector b, bHYPRE_Vector* x,
                                sidl_BaseInterface *_ex)
      Apply the operator to b, returning x
SIDL_C_INLINE_DECL_int32_t
bHYPRE_StructJacobi_ApplyAdjoint ( bHYPRE_StructJacobi self,
                                        bHYPRE_Vector b.
                                        bHYPRE_Vector* x,
                                        sidl_BaseInterface *_ex)
      Apply the adjoint of the operator to b, returning x
_ex
       Cast method for interface and class type conversions
void*
bHYPRE_StructJacobi_cast2 (void* obj, const char* type,
                                sidl_BaseInterface *_ex)
      String cast method for interface and class type conversions
SIDL_C_INLINE_DECL void
bHYPRE_StructJacobi_exec ( bHYPRE_StructJacobi self,
                               const char* methodName,
                               sidl_rmi_Call inArgs, sidl_rmi_Return outArgs,
                               sidl_BaseInterface *_ex)
      Select and execute a method by name
SIDL_C_INLINE_DECL char*
bHYPRE_StructJacobi__getURL ( bHYPRE_StructJacobi self,
                                   sidl_BaseInterface *_ex)
       Get the URL of the Implementation of this object (for RMI)
SIDL_C_INLINE_DECL void
bHYPRE_StructJacobi_raddRef ( bHYPRE_StructJacobi self,
                                   sidl_BaseInterface *_ex)
       On a remote object, addrefs the remote instance
```

SIDL_C_INLINE_DECL_sidl_bool

bHYPRE_StructJacobi_isRemote (bHYPRE_StructJacobi self, sidl_BaseInterface *_ex)

TRUE if this object is remote, false if local

sidl_bool

bHYPRE_StructJacobi__isLocal (bHYPRE_StructJacobi self, sidl_BaseInterface *_ex)

TRUE if this object is remote, false if local

 $**_ex$

Cast method for interface and class type conversions

7.2.10 **_ex

7.2.1 $_{-}$

struct bHYPRE_StructJacobi_object

Symbol "bHYPRE.StructJacobi" (version 1.0.0)

Objects of this type can be cast to Solver objects using the __cast methods.

RDF: Documentation goes here.

The StructJacobi solver requires a Struct matrix.

_ 7.2.2 _

bHYPRE_StructJacobi bHYPRE_StructJacobi__connect (const char *, sidl_BaseInterface *_ex)

RMI connector function for the class.(addrefs)

_ 7.2.3 __

SIDL_C_INLINE_DECL int32_t bHYPRE_StructJacobi_SetOperator (bHYPRE_StructJacobi self, bHYPRE_Operator A, sidl_BaseInterface *_ex)

Set the operator for the linear system being solved. DEPRECATED. use Create

7.2.4

SIDL_C_INLINE_DECL int32_t bHYPRE_StructJacobi_SetTolerance (bHYPRE_StructJacobi self, double tolerance, sidl_BaseInterface *_ex)

(Optional) Set the convergence tolerance. DEPRECATED. use SetDoubleParameter

$_$ 7.2.5 $_$

SIDL_C_INLINE_DECL int32_t bHYPRE_StructJacobi_SetMaxIterations (bHYPRE_StructJacobi self, int32_t max_iterations, sidl_BaseInterface *_ex)

(Optional) Set maximum number of iterations. DEPRECATED use SetIntParameter

_ 7.2.6 __

SIDL_C_INLINE_DECL int32_t bHYPRE_StructJacobi_SetLogging (bHYPRE_StructJacobi self, int32_t level, sidl_BaseInterface *_ex)

(Optional) Set the *logging level*, specifying the degree of additional informational data to be accumulated. Does nothing by default (level = 0). Other levels (if any) are implementation-specific. Must be called before Setup and Apply. DEPRECATED use SetIntParameter

$_$ 7.2.7 $_$

SIDL_C_INLINE_DECL int32_t bHYPRE_StructJacobi_SetPrintLevel (bHYPRE_StructJacobi self, int32_t level, sidl_BaseInterface *_ex)

(Optional) Set the *print level*, specifying the degree of informational data to be printed either to the screen or to a file. Does nothing by default (level=0). Other levels (if any) are implementation-specific. Must be called before Setup and Apply. DEPRECATED use SetIntParameter

7.2.8

SIDL_C_INLINE_DECL int32_t bHYPRE_StructJacobi_SetCommunicator (bHYPRE_StructJacobi self, bHYPRE_MPICommunicator mpi_comm, sidl_BaseInterface *_ex)

Set the MPI Communicator. DEPRECATED, use Create:

7.2.9

SIDL_C_INLINE_DECL void **bHYPRE_StructJacobi_Destroy** (bHYPRE_StructJacobi self, sidl_BaseInterface *_ex)

The Destroy function doesn't necessarily destroy anything. It is just another name for deleteRef. Thus it decrements the object's reference count. The Babel memory management system will destroy the object if the reference count goes to zero.

7.2.10

RMI connector function for the class. (no addref)

7.3

Struct PFMG Solver

Names		
7.3.1	struct bHYPRE_StructPFMGobject Symbol "bHYPRE	180
	_ex Constructor function for the class	
	bHYPRE_StructPFMG bHYPRE_StructPFMGcreateRemote (const char * url, sidl_BaseInterface *_ex)	
	RMI constructor function for the class	
	bHYPRE_StructPFMG bHYPRE_StructPFMGwrapObj (void * data, sidl_BaseInterface *_ex) Wraps up the private data struct pointer (struct bHYPRE_StructPFMGdata) passed in rather than running the con- structor	
7.3.2	bHYPRE_StructPFMG bHYPRE_StructPFMGconnect (const char *, sidl_BaseInterface *_ex) RMI connector function for the class	181
	bHYPRE_StructPFMG bHYPRE_StructPFMG_Create (bHYPRE_MPICommunicator mpi_comm, bHYPRE_StructMatrix A, sidl_BaseInterface *_ex) Method: Create[]	
7.3.3	SIDL_C_INLINE_DECL int32_t bHYPRE_StructPFMG_SetOperator (bHYPRE_StructPFMG self,	
	Set the operator for the linear system being solved	181
7.3.4	SIDL_C_INLINE_DECL_int32_t bHYPRE_StructPFMG_SetTolerance (bHYPRE_StructPFMG self, double tolerance, sidl_BaseInterface *_ex)	
	(Optional) Set the convergence tolerance	181
7.3.5	SIDL_C_INLINE_DECL int32_t bHYPRE_StructPFMG_SetMaxIterations (bHYPRE_StructPFMG self, int32_t max_iterations, sidl_BaseInterface *_ex)	
	(Optional) Set maximum number of iterations	181
7.3.6	SIDL_C_INLINE_DECL int32_t bHYPRE_StructPFMG_SetLogging (bHYPRE_StructPFMG self,	
	tional data to be accumulated	182
7.3.7	SIDL_C_INLINE_DECL_int32_t	

	bHYPRE_StructPFMG_SetPrintLevel (bHYPRE_StructPFMG self,	
	int32_t level, sidl_BaseInterface *_ex)	
	(Optional) Set the print level, specifying the degree of informational data	
	to be printed either to the screen or to a file	182
	SIDL_C_INLINE_DECL int32_t bHYPRE_StructPFMG_GetNumIterations (bHYPRE_StructPFMG self, int32_t* num_iterations,	
	sidl_BaseInterface *_ex) (Optional) Return the number of iterations taken	
	· · ·	
	SIDL_C_INLINE_DECL int32_t bHYPRE_StructPFMG_GetRelResidualNorm (bHYPRE_StructPFMG self, double* norm, sidl_BaseInterface *_ex)	
	(Optional) Return the norm of the relative residual	
7.3.8	SIDL_C_INLINE_DECL int32_t	
	bHYPRE_StructPFMG_SetCommunicator (bHYPRE_StructPFMG self, bHYPRE_MPICommunicator	
	mpi_comm, sidl_BaseInterface *_ex)	
	Set the MPI Communicator	182
7.3.9	SIDL_C_INLINE_DECL void bHYPRE_StructPFMG_Destroy (bHYPRE_StructPFMG self,	
	The Destroy function doesn't necessarily destroy anything	183
	SIDL_C_INLINE_DECL int32_t bHYPRE_StructPFMG_SetIntParameter (bHYPRE_StructPFMG self,	
	const char* name, int32_t value, sidl_BaseInterface *_ex)	
	Set the int parameter associated with name	
	SIDL_C_INLINE_DECL int32_t	
	bHYPRE_StructPFMG_SetDoubleParameter (bHYPRE_StructPFMG self, const char* name, double value, sidl_BaseInterface *_ex)	
	Set the double parameter associated with name	
	SIDL_C_INLINE_DECL int32_t bHYPRE_StructPFMG_SetStringParameter (bHYPRE_StructPFMG self,	
	sidl_BaseInterface *_ex)	
	Set the string parameter associated with name	
	SIDL_C_INLINE_DECL int32_t bHYPRE_StructPFMG_SetIntArray1Parameter (bHYPRE_StructPFMG self, const char* name, int32_t* value, int32_t* values, sidl_BaseInterface *_ex)	
	Set the int 1-D array parameter associated with name	
	SIDL_C_INLINE_DECL int32_t	

```
bHYPRE_StructPFMG_SetIntArray2Parameter ( bHYPRE_StructPFMG
                                                  self, const char* name,
                                                  struct sidl_int_array*
                                                  value.
                                                  sidl_BaseInterface *_ex)
      Set the int 2-D array parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_StructPFMG_SetDoubleArray1Parameter (
                                                      bHYPRE_StructPFMG
                                                      self.
                                                      const char* name,
                                                      double* value,
                                                      int32_t nvalues,
                                                      sidl_BaseInterface
                                                      *_ex)
      Set the double 1-D array parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_StructPFMG_SetDoubleArray2Parameter (
                                                      bHYPRE_StructPFMG
                                                      self.
                                                      const char* name,
                                                      struct
                                                      sidl_double_array*
                                                      value,
                                                      sidl_BaseInterface
                                                      *_ex)
      Set the double 2-D array parameter associated with name
SIDL_C_INLINE_DECL_int32_t
bHYPRE_StructPFMG_GetIntValue ( bHYPRE_StructPFMG self,
                                      const char* name, int32_t* value,
                                      sidl_BaseInterface *_ex)
      Set the int parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_StructPFMG_GetDoubleValue ( bHYPRE_StructPFMG self,
                                           const char* name.
                                           double* value.
                                           sidl_BaseInterface *_ex)
      Get the double parameter associated with name
SIDL_C_INLINE_DECL_int32_t
bHYPRE_StructPFMG_Setup ( bHYPRE_StructPFMG self,
                                bHYPRE_Vector b, bHYPRE_Vector x,
                                sidl_BaseInterface *_ex)
      (Optional) Do any preprocessing that may be necessary in order to execute
      Apply
SIDL_C_INLINE_DECL int32_t
bHYPRE_StructPFMG_Apply ( bHYPRE_StructPFMG self,
                                bHYPRE_Vector b, bHYPRE_Vector* x,
                                sidl_BaseInterface *_ex)
      Apply the operator to b, returning x
SIDL_C_INLINE_DECL int32_t
```

```
bHYPRE_StructPFMG_ApplyAdjoint ( bHYPRE_StructPFMG self,
                                        bHYPRE_Vector b,
                                        bHYPRE_Vector* x,
                                        sidl_BaseInterface *_ex)
      Apply the adjoint of the operator to b, returning x
_ex
      Cast method for interface and class type conversions
void*
bHYPRE_StructPFMG__cast2 (void* obj, const char* type,
                                sidl_BaseInterface *_ex)
      String cast method for interface and class type conversions
SIDL_C_INLINE_DECL void
bHYPRE_StructPFMG__exec ( bHYPRE_StructPFMG self,
                               const char* methodName,
                               sidl_rmi_Call inArgs,
                               sidl_rmi_Return outArgs,
                               sidl_BaseInterface *_ex)
      Select and execute a method by name
SIDL_C_INLINE_DECL char*
bHYPRE_StructPFMG__getURL ( bHYPRE_StructPFMG self,
                                   sidl_BaseInterface *_ex)
      Get the URL of the Implementation of this object (for RMI)
SIDL_C_INLINE_DECL_void
bHYPRE_StructPFMG__raddRef ( bHYPRE_StructPFMG self,
                                   sidl_BaseInterface *_ex)
      On a remote object, addrefs the remote instance
SIDL_C_INLINE_DECL_sidl_bool
bHYPRE_StructPFMG__isRemote ( bHYPRE_StructPFMG self,
                                    sidl_BaseInterface *_ex)
       TRUE if this object is remote, false if local
sidl\_bool
bHYPRE_StructPFMG__isLocal ( bHYPRE_StructPFMG self,
                                  sidl_BaseInterface *_ex)
      TRUE if this object is remote, false if local
**_ex
       Cast method for interface and class type conversions
**_ex
      RMI connector function for the class .....
                                                                               183
```

_ 7.3.1 _

7.3.10

struct bHYPRE_StructPFMG__object

Symbol "bHYPRE.StructPFMG" (version 1.0.0)

Objects of this type can be cast to Solver objects using the __cast methods.

RDF: Documentation goes here.

The StructPFMG solver requires a Struct matrix.

 $_{-}$ 7.3.2 $_{-}$

bHYPRE_StructPFMG bHYPRE_StructPFMG__connect (const char *, sidl_BaseInterface *_ex)

RMI connector function for the class.(addrefs)

7.3.3

SIDL_C_INLINE_DECL int32_t bHYPRE_StructPFMG_SetOperator (bHYPRE_StructPFMG self, bHYPRE_Operator A, sidl_BaseInterface *_ex)

Set the operator for the linear system being solved. DEPRECATED. use Create

 $_$ 7.3.4 $_$

SIDL_C_INLINE_DECL int32_t bHYPRE_StructPFMG_SetTolerance (bHYPRE_StructPFMG self, double tolerance, sidl_BaseInterface *_ex)

(Optional) Set the convergence tolerance. DEPRECATED. use SetDoubleParameter

 $_$ 7.3.5 $_$

SIDL_C_INLINE_DECL int32_t bHYPRE_StructPFMG_SetMaxIterations (bHYPRE_StructPFMG self, int32_t max_iterations, sidl_BaseInterface *_ex)

(Optional) Set maximum number of iterations. DEPRECATED use SetIntParameter

7.3.6

SIDL_C_INLINE_DECL int32_t bHYPRE_StructPFMG_SetLogging (bHYPRE_StructPFMG self, int32_t level, sidl_BaseInterface *_ex)

(Optional) Set the *logging level*, specifying the degree of additional informational data to be accumulated. Does nothing by default (level = 0). Other levels (if any) are implementation-specific. Must be called before Setup and Apply. DEPRECATED use SetIntParameter

7.3.7

SIDL_C_INLINE_DECL int32_t bHYPRE_StructPFMG_SetPrintLevel (bHYPRE_StructPFMG self, int32_t level, sidl_BaseInterface *_ex)

(Optional) Set the *print level*, specifying the degree of informational data to be printed either to the screen or to a file. Does nothing by default (level=0). Other levels (if any) are implementation-specific. Must be called before Setup and Apply. DEPRECATED use SetIntParameter

7.3.8

SIDL_C_INLINE_DECL int32_t bHYPRE_StructPFMG_SetCommunicator (bHYPRE_StructPFMG self, bHYPRE_MPICommunicator mpi_comm, sidl_BaseInterface *_ex)

Set the MPI Communicator. DEPRECATED, use Create:

7.3.9

SIDL_C_INLINE_DECL void bHYPRE_StructPFMG_Destroy (bHYPRE_StructPFMG self, sidl_BaseInterface *_ex)

The Destroy function doesn't necessarily destroy anything. It is just another name for deleteRef. Thus it decrements the object's reference count. The Babel memory management system will destroy the object if the reference count goes to zero.

7.3.10

RMI connector function for the class. (no addref)

_ 7.4 _

Struct SMG Solver

Names

7.4.1struct bHYPRE_StructSMG__object Symbol "bHYPRE 187 _ex Constructor function for the class $bHYPRE_StructSMG$ bHYPRE_StructSMG__createRemote (const char * url, sidl_BaseInterface *_ex) RMI constructor function for the class bHYPRE_StructSMG bHYPRE_StructSMG_wrapObj (void * data, sidl_BaseInterface *_ex) Wraps up the private data struct pointer (struct bHYPRE_StructSMG__data) passed in rather than running the constructor $bHYPRE_StructSMG$ 7.4.2 bHYPRE_StructSMG__connect (const char *, sidl_BaseInterface *_ex) RMI connector function for the class 187 bHYPRE_StructSMG

	bHYPRE_StructSMG_Create (bHYPRE_MPICommunicator mpi_comm, bHYPRE_StructMatrix A,	
	sidl_BaseInterface *_ex)	
	Method: Create[]	
7.4.3	SIDL_C_INLINE_DECL int32_t	
	$\mathbf{bHYPRE_StructSMG_SetOperator} \ (\ \ \mathbf{bHYPRE_StructSMG} \ \mathbf{self},$	
	bHYPRE_Operator A,	
	sidl_BaseInterface *_ex) Set the operator for the linear system being solved	188
7 4 4	SIDL_C_INLINE_DECL int32_t	100
7.4.4	bHYPRE_StructSMG_SetTolerance (bHYPRE_StructSMG self,	
	double tolerance,	
	sidl_BaseInterface *_ex)	
	(Optional) Set the convergence tolerance	188
7.4.5	SIDL_C_INLINE_DECL int32_t	
	bHYPRE_StructSMG_SetMaxIterations (bHYPRE_StructSMG self,	
	int32_t max_iterations,	
	sidl_BaseInterface *_ex) (Optional) Set maximum number of iterations	188
7.4.6	SIDL_C_INLINE_DECL int32_t	100
7.4.0	bHYPRE_StructSMG_SetLogging (bHYPRE_StructSMG self,	
	int32_t level, sidl_BaseInterface *_ex)	
	(Optional) Set the logging level, specifying the degree of additional informational data to be accumulated	188
7.4.7	SIDL_C_INLINE_DECL int32_t	
	$\mathbf{bHYPRE_StructSMG_SetPrintLevel} \ (\ \ \mathbf{bHYPRE_StructSMG} \ \mathbf{self},$	
	int32_t level, sidl_BaseInterface *_ex)	
	(Optional) Set the print level, specifying the degree of informational data to be printed either to the screen or to a file	189
	SIDL_C_INLINE_DECL int32_t	
	bHYPRE_StructSMG_GetNumIterations (bHYPRE_StructSMG self,	
	int32_t* num_iterations, sidl_BaseInterface *_ex)	
	(Optional) Return the number of iterations taken	
	SIDL_C_INLINE_DECL int32_t	
	$\mathbf{bHYPRE_StructSMG_GetRelResidualNorm} \ (\ \ \mathbf{bHYPRE_StructSMG} \ \mathbf{self},$	
	double* norm,	
	sidl_BaseInterface *_ex)	
	(Optional) Return the norm of the relative residual	
7.4.8	SIDL_C_INLINE_DECL int32_t bHYPRE_StructSMG_SetCommunicator (bHYPRE_StructSMG self,	
	bHYPRE_MPICommunicator	
	mpi_comm,	
	sidl_BaseInterface *_ex)	
	Set the MPI Communicator	189
7.4.9	SIDL_C_INLINE_DECL void	

```
bHYPRE_StructSMG_Destroy ( bHYPRE_StructSMG self,
                                sidl_BaseInterface *_ex)
       The Destroy function doesn't necessarily destroy anything .....
                                                                                189
SIDL_C_INLINE_DECL int32_t
bHYPRE_StructSMG_SetIntParameter ( bHYPRE_StructSMG self,
                                         const char* name, int32_t value,
                                         sidl_BaseInterface *_ex)
      Set the int parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_StructSMG_SetDoubleParameter ( bHYPRE_StructSMG self,
                                              const char* name,
                                              double value.
                                              sidl_BaseInterface *_ex)
      Set the double parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_StructSMG_SetStringParameter ( bHYPRE_StructSMG self,
                                             const char* name,
                                             const char* value,
                                             sidl_BaseInterface *_ex)
      Set the string parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_StructSMG_SetIntArray1Parameter ( bHYPRE_StructSMG self,
                                                 const char* name,
                                                 int32_t* value,
                                                int32_t nvalues.
                                                sidl_BaseInterface *_ex)
      Set the int 1-D array parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_StructSMG_SetIntArray2Parameter ( bHYPRE_StructSMG self,
                                                const char* name, struct
                                                sidl_int_array* value,
                                                sidl_BaseInterface *_ex)
      Set the int 2-D array parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_StructSMG_SetDoubleArray1Parameter (
                                                     bHYPRE_StructSMG
                                                     const char* name.
                                                     double* value,
                                                     int32_t nvalues,
                                                     sidl\_BaseInterface
                                                     *_ex)
      Set the double 1-D array parameter associated with name
```

SIDL_C_INLINE_DECL int32_t

```
bHYPRE_StructSMG_SetDoubleArray2Parameter (
                                                     bHYPRE_StructSMG
                                                     const char* name,
                                                     struct
                                                     sidl_double_array*
                                                     value.
                                                     sidl_BaseInterface
                                                     *_ex)
      Set the double 2-D array parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_StructSMG_GetIntValue ( bHYPRE_StructSMG self,
                                     const char* name, int32_t* value,
                                     sidl_BaseInterface *_ex)
      Set the int parameter associated with name
SIDL_C_INLINE_DECL_int32_t
bHYPRE_StructSMG_GetDoubleValue ( bHYPRE_StructSMG self,
                                         const char* name, double* value,
                                         sidl_BaseInterface *_ex)
      Get the double parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_StructSMG_Setup ( bHYPRE_StructSMG self,
                              bHYPRE_Vector b, bHYPRE_Vector x,
                              sidl_BaseInterface *_ex)
      (Optional) Do any preprocessing that may be necessary in order to execute
      Apply
SIDL_C_INLINE_DECL int32_t
bHYPRE_StructSMG_Apply ( bHYPRE_StructSMG self,
                               bHYPRE_Vector b, bHYPRE_Vector* x,
                               sidl_BaseInterface *_ex)
      Apply the operator to b, returning x
SIDL_C_INLINE_DECL int32_t
bHYPRE_StructSMG_ApplyAdjoint ( bHYPRE_StructSMG self,
                                       bHYPRE_Vector b,
                                      bHYPRE_Vector* x,
                                       sidl_BaseInterface *_ex)
      Apply the adjoint of the operator to b, returning x
_{-}ex
      Cast method for interface and class type conversions
void*
bHYPRE_StructSMG__cast2 (void* obj, const char* type,
                              sidl_BaseInterface *_ex)
      String cast method for interface and class type conversions
SIDL_C_INLINE_DECL void
bHYPRE_StructSMG__exec ( bHYPRE_StructSMG self,
                              const char* methodName, sidl_rmi_Call inArgs,
                              sidl_rmi_Return outArgs,
                              sidl_BaseInterface *_ex)
      Select and execute a method by name
```

SIDL_C_INLINE_DECL_char*

bHYPRE_StructSMG__getURL (bHYPRE_StructSMG self, sidl_BaseInterface *_ex) Get the URL of the Implementation of this object (for RMI) SIDL_C_INLINE_DECL void bHYPRE_StructSMG__raddRef (bHYPRE_StructSMG self, sidl_BaseInterface *_ex) On a remote object, addrefs the remote instance SIDL_C_INLINE_DECL_sidl_bool bHYPRE_StructSMG__isRemote (bHYPRE_StructSMG self, sidl_BaseInterface *_ex) TRUE if this object is remote, false if local sidl_bool bHYPRE_StructSMG__isLocal (bHYPRE_StructSMG self, sidl_BaseInterface *_ex) TRUE if this object is remote, false if local **_ex Cast method for interface and class type conversions $\mathbf{**}_{-}\mathbf{ex}$

RMI connector function for the class

 $_{ extsf{-}}$ 7.4.1 $_{ extsf{-}}$

7.4.10

struct bHYPRE_StructSMG__object

 $Symbol "bHYPRE.StructSMG" \ (version \ 1.0.0)$

Objects of this type can be cast to Solver objects using the **__cast** methods.

RDF: Documentation goes here.

The StructSMG solver requires a Struct matrix.

7.4.2

bHYPRE_StructSMG bHYPRE_StructSMG__connect (const char *, sidl_BaseInterface *_ex)

RMI connector function for the class.(addrefs)

189

7.4.3

SIDL_C_INLINE_DECL int32_t bHYPRE_StructSMG_SetOperator (bHYPRE_StructSMG self, bHYPRE_Operator A, sidl_BaseInterface *_ex)

Set the operator for the linear system being solved. DEPRECATED. use Create

7.4.4

SIDL_C_INLINE_DECL int32_t bHYPRE_StructSMG_SetTolerance (bHYPRE_StructSMG self, double tolerance, sidl_BaseInterface *_ex)

(Optional) Set the convergence tolerance. DEPRECATED. use SetDoubleParameter

__ 7.4.5 _____

SIDL_C_INLINE_DECL int32_t bHYPRE_StructSMG_SetMaxIterations (bHYPRE_StructSMG self, int32_t max_iterations, sidl_BaseInterface *_ex)

(Optional) Set maximum number of iterations. DEPRECATED use SetIntParameter

__ 7.4.6 _____

(Optional) Set the $logging\ level$, specifying the degree of additional informational data to be accumulated. Does nothing by default (level = 0). Other levels (if any) are implementation-specific. Must be called before Setup and Apply. DEPRECATED use SetIntParameter

7.4.7

SIDL_C_INLINE_DECL int32_t bHYPRE_StructSMG_SetPrintLevel (bHYPRE_StructSMG self, int32_t level, sidl_BaseInterface *_ex)

(Optional) Set the *print level*, specifying the degree of informational data to be printed either to the screen or to a file. Does nothing by default (level=0). Other levels (if any) are implementation-specific. Must be called before Setup and Apply. DEPRECATED use SetIntParameter

7.4.8

SIDL_C_INLINE_DECL int32_t bHYPRE_StructSMG_SetCommunicator (bHYPRE_StructSMG self, bHYPRE_MPICommunicator mpi_comm, sidl_BaseInterface *_ex)

Set the MPI Communicator. DEPRECATED, use Create:

7.4.9

The Destroy function doesn't necessarily destroy anything. It is just another name for deleteRef. Thus it decrements the object's reference count. The Babel memory management system will destroy the object if the reference count goes to zero.

7.4.10

struct bHYPRE_StructSMG__object* bHYPRE_StructSMG__connectI const char * url sidl_bool ar struct sidl_BaseInterface__object **_ex

RMI connector function for the class. (no addref)

- 8

SemiStructured Matrix Solvers

Names 8.1	SemiStruct DiagScale Solver	
		190
8.2	Struct Split Solver	19'
These solve	ers use semi-structured matrix/vector storage schemes.	
8.1		
Sem	iStruct DiagScale Solver	
Names		
8.1.1	struct bHYPRE_SStructDiagScaleobject Symbol "bHYPRE	194
	_ex Constructor function for the class	
	bhypre_sstructDiagScale bhypre_sstructDiagScalecreateRemote (const char * url,	
	RMI constructor function for the class	
	bHYPRE_SStructDiagScale bHYPRE_SStructDiagScale_wrapObj (void * data, sidl_BaseInterface *_ex) Wraps up the private data struct pointer (struct bHYPRE_SStructDiagScale_data) passed in rather than running the constructor	
3.1.2	bHYPRE_SStructDiagScale bHYPRE_SStructDiagScaleconnect (const char *, sidl_BaseInterface *_ex) RMI connector function for the class	195
	bHYPRE_SStructDiagScale bHYPRE_SStructDiagScale_Create (bHYPRE_MPICommunicator mpi_comm, bHYPRE_Operator A, sidl_BaseInterface *_ex)	1 <i>3</i> t

8.1.3

Method: Create[]

 $SIDL_C_INLINE_DECL \ int 32_t$

	bHYPRE_SStructDiagScale_SetOperator (bHYPRE_SStructDiagScale self, bHYPRE_Operator A, sidl_BaseInterface *_ex)	
	,	195
8.1.4	SIDL_C_INLINE_DECL int32_t bHYPRE_SStructDiagScale_SetTolerance (bHYPRE_SStructDiagScale self,	
	double tolerance, sidl_BaseInterface *_ex)	
	,	195
8.1.5	SIDL_C_INLINE_DECL int32_t	
0.2.0	bHYPRE_SStructDiagScale_SetMaxIterations (
	bHYPRE_SStructDiagScale	
	$_{ m self}, \\ m int 32_t \ max_iterations, \\$	
	sidl_BaseInterface *_ex)	
	,	195
8.1.6	SIDL_C_INLINE_DECL int32_t	
	bHYPRE_SStructDiagScale_SetLogging (bHYPRE_SStructDiagScale self, int32_t level,	
	sidl_BaseInterface *_ex)	
	(Optional) Set the logging level, specifying the degree of additional informational data to be accumulated	196
8.1.7	SIDL_C_INLINE_DECL int32_t bHYPRE_SStructDiagScale_SetPrintLevel (bHYPRE_SStructDiagScale self, int32_t level,	
	$sidl_BaseInterface *_ex)$	
	(Optional) Set the print level, specifying the degree of informational data	
	·	196
	SIDL_C_INLINE_DECL int32_t	
	$b HYPRE_SStructDiagScale_GetNumIterations \ (\\bHYPRE_SStructDiagScale$	
	self,	
	int32_t* num_iterations,	
	sidl_BaseInterface *_ex)	
	(Optional) Return the number of iterations taken	
	SIDL_C_INLINE_DECL int32_t bHYPRE_SStructDiagScale_GetRelResidualNorm (
	bHYPRE_SStructDiagScale self, double* norm,	
	sidl_BaseInterface *_ex) (Optional) Return the norm of the relative residual	
8.1.8	SIDL_C_INLINE_DECL int32_t	
0.1.0	bHYPRE_SStructDiagScale_SetCommunicator (
	bHYPRE_SStructDiagScale	
	self,	
	bHYPRE_MPICommunicator	
	mpi_comm, sidl_BaseInterface *_ex)	
	,	196
8.1.9	SIDL_C_INLINE_DECL_void	

```
bHYPRE_SStructDiagScale_Destroy ( bHYPRE_SStructDiagScale self,
                                        sidl_BaseInterface *_ex)
       The Destroy function doesn't necessarily destroy anything ...
                                                                                   196
SIDL_C_INLINE_DECL int32_t
bHYPRE\_SStructDiagScale\_SetIntParameter ( bHYPRE\_SStructDiagScale
                                                 self, const char* name,
                                                 int32_t value,
                                                 sidl_BaseInterface *_ex)
       Set\ the\ int\ parameter\ associated\ with\ {\tt name}
SIDL_C_INLINE_DECL int32_t
bHYPRE_SStructDiagScale_SetDoubleParameter (
                                                      bHYPRE\_SStructDiagScale
                                                     self, const char* name,
                                                      double value,
                                                     sidl_BaseInterface *_ex)
       Set the double parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_SStructDiagScale_SetStringParameter (
                                                     bHYPRE\_SStructDiagScale
                                                    self, const char* name,
                                                    const char* value,
                                                    sidl_BaseInterface *_ex)
       Set the string parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_SStructDiagScale_SetIntArray1Parameter (
                                                         bHYPRE\_SStructDiagScale
                                                         self.
                                                         const char* name,
                                                         int32_t* value.
                                                         int32_t nvalues,
                                                         sidl\_BaseInterface
                                                         *_ex)
       Set the int 1-D array parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_SStructDiagScale_SetIntArray2Parameter (
                                                         bHYPRE\_SStructDiagScale
                                                         self.
                                                         const char* name,
                                                         struct
                                                        sidl_int_array*
                                                         value,
                                                         sidl_BaseInterface
                                                         *_ex)
       Set the int 2-D array parameter associated with name
```

SIDL_C_INLINE_DECL int32_t

```
bHYPRE_SStructDiagScale_SetDoubleArray1Parameter (
                                                            bHYPRE_SStructDiagScale
                                                           self, const
                                                           char* name,
                                                           double* value,
                                                           int32_t nvalues,
                                                           sidl_BaseInterface
                                                            *_ex)
      Set the double 1-D array parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_SStructDiagScale_SetDoubleArray2Parameter (
                                                            bHYPRE_SStructDiagScale
                                                           self, const
                                                           char* name,
                                                           struct
                                                           sidl_double_array*
                                                            value,
                                                           sidl_BaseInterface
                                                            *_ex)
      Set the double 2-D array parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_SStructDiagScale_GetIntValue ( bHYPRE_SStructDiagScale self,
                                           const char* name,
                                           int32_t* value,
                                           sidl_BaseInterface *_ex)
      Set the int parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_SStructDiagScale_GetDoubleValue ( bHYPRE_SStructDiagScale
                                                self, const char* name,
                                                double* value,
                                                sidl_BaseInterface *_ex)
      Get the double parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_SStructDiagScale_Setup ( bHYPRE_SStructDiagScale self,
                                    bHYPRE_Vector b, bHYPRE_Vector x,
                                    sidl_BaseInterface *_ex)
       (Optional) Do any preprocessing that may be necessary in order to execute
      Apply
SIDL_C_INLINE_DECL int32_t
bHYPRE_SStructDiagScale_Apply ( bHYPRE_SStructDiagScale self,
                                     bHYPRE_Vector b,
                                     bHYPRE_Vector* x,
                                     sidl_BaseInterface *_ex)
      Apply the operator to b, returning x
SIDL_C_INLINE_DECL int32_t
bHYPRE_SStructDiagScale_ApplyAdjoint ( bHYPRE_SStructDiagScale
                                             self, bHYPRE_Vector b,
                                             bHYPRE_Vector* x,
                                             sidl_BaseInterface *_ex)
      Apply the adjoint of the operator to b, returning x
_{-}ex
```

Cast method for interface and class type conversions

void*

bHYPRE_SStructDiagScale__cast2 (void* obj, const char* type, sidl_BaseInterface *_ex)

String cast method for interface and class type conversions

SIDL_C_INLINE_DECL void

bHYPRE_SStructDiagScale __exec (bHYPRE_SStructDiagScale self,

const char* methodName, sidl_rmi_Call inArgs, sidl_rmi_Return outArgs, sidl_BaseInterface *_ex)

Select and execute a method by name

SIDL_C_INLINE_DECL char*

 ${\bf bHYPRE_SStructDiagScale__getURL} \ (\ \ {\bf bHYPRE_SStructDiagScale} \ self,$

sidl_BaseInterface *_ex)

Get the URL of the Implementation of this object (for RMI)

SIDL_C_INLINE_DECL void

bHYPRE_SStructDiagScale__raddRef (bHYPRE_SStructDiagScale self, sidl_BaseInterface *_ex)

On a remote object, addrefs the remote instance

SIDL_C_INLINE_DECL sidl_bool

 $b HYPRE_SStructDiagScale__isRemote \ (\ bHYPRE_SStructDiagScale \ self, \ bHYPRE_STRUCTDIAGSCALE \ sel$

 $sidl_BaseInterface *_ex)$

TRUE if this object is remote, false if local

sidl_bool

bHYPRE_SStructDiagScale__isLocal (bHYPRE_SStructDiagScale self,

sidl_BaseInterface *_ex)

TRUE if this object is remote, false if local

**_ex

Cast method for interface and class type conversions

8.1.10 ******_**ex**

8.1.1

 $struct\ bHYPRE_SStructDiagScale_object$

Symbol "bHYPRE.SStructDiagScale" (version 1.0.0)

_ 8.1.2 ____

bHYPRE_SStructDiagScale
bHYPRE_SStructDiagScale_connect (const char *, sidl_BaseInterface *_ex)

RMI connector function for the class.(addrefs)

8.1.3

SIDL_C_INLINE_DECL int32_t bHYPRE_SStructDiagScale_SetOperator (bHYPRE_SStructDiagScale self, bHYPRE_Operator A, sidl_BaseInterface *_ex)

Set the operator for the linear system being solved. DEPRECATED. use Create

8.1.4

(Optional) Set the convergence tolerance. DEPRECATED. use SetDoubleParameter

 $_$ 8.1.5 $_$

SIDL_C_INLINE_DECL int32_t bHYPRE_SStructDiagScale_SetMaxIterations (bHYPRE_SStructDiagScale self, int32_t max_iterations, sidl_BaseInterface *_ex)

(Optional) Set maximum number of iterations. DEPRECATED use SetIntParameter

8.1.6

(Optional) Set the *logging level*, specifying the degree of additional informational data to be accumulated. Does nothing by default (level = 0). Other levels (if any) are implementation-specific. Must be called before Setup and Apply. DEPRECATED use SetIntParameter

$_$ 8.1.7 $_$

SIDL_C_INLINE_DECL int32_t bHYPRE_SStructDiagScale_SetPrintLevel (bHYPRE_SStructDiagScale self, int32_t level, sidl_BaseInterface *_ex)

(Optional) Set the *print level*, specifying the degree of informational data to be printed either to the screen or to a file. Does nothing by default (level=0). Other levels (if any) are implementation-specific. Must be called before Setup and Apply. DEPRECATED use SetIntParameter

_ 8.1.8 _

SIDL_C_INLINE_DECL int32_t bHYPRE_SStructDiagScale_SetCommunicator (bHYPRE_SStructDiagScale self, bHYPRE_MPICommunicator mpi_comm, sidl_BaseInterface *_ex)

Set the MPI Communicator. DEPRECATED, use Create:

_ 8.1.9 _

SIDL_C_INLINE_DECL void bHYPRE_SStructDiagScale self, sidl_BaseInterface *_ex)

The Destroy function doesn't necessarily destroy anything. It is just another name for deleteRef. Thus it decrements the object's reference count. The Babel memory management system will destroy the object if the reference count goes to zero.

8.1.10

struct bHYPRE_SStructDiagScale__object* bHYPRE_SStructDiagScale__connectI const char * url sidl_bool ar struct sidl_BaseInterface__object **_ex

RMI connector function for the class. (no addref)

_ 8.2

Struct Split Solver

Names		
8.2.1	struct bHYPRE_SStructSplitobject	
	Symbol "bHYPRE	201
	_ex	
	Constructor function for the class	
	bHYPRE_SStructSplit	
	bHYPRE_SStructSplitcreateRemote (const char * url, sidl_BaseInterface *_ex)	
	RMI constructor function for the class	
	bHYPRE_SStructSplit	
	bHYPRE_SStructSplitwrapObj (void * data, sidl_BaseInterface *_ex)	
	Wraps up the private data struct pointer (struct bHYPRE_SStructSplit_data) passed in rather than running the constructor	
0.0.0		
8.2.2	bHYPRE_SStructSplit bHYPRE_SStructSplitconnect (const char *, sidl_BaseInterface *_ex)	
	RMI connector function for the class	201
	bHYPRE_SStructSplit	
	bHYPRE_SStructSplit_Create (bHYPRE_MPICommunicator mpi_comm,	
	bHYPRE_Operator A, sidl_BaseInterface *_ex)	
	Method: Create[]	
8.2.3	SIDL_C_INLINE_DECL int32_t	
	$\mathbf{bHYPRE_SStructSplit_SetOperator} \ (\ \ \mathbf{bHYPRE_SStructSplit} \ \mathbf{self},$	
	bHYPRE_Operator A,	
	sidl_BaseInterface *_ex) Set the operator for the linear system being solved	201
8.2.4	SIDL_C_INLINE_DECL int32_t	
0.2.1	bHYPRE_SStructSplit_SetTolerance (bHYPRE_SStructSplit self,	
	double tolerance,	
	sidl_BaseInterface *_ex) (Optional) Set the convergence tolerance	201
0.0.5		201
8.2.5	SIDL_C_INLINE_DECL int32_t	

	bHYPRE_SStructSplit_SetMaxIterations (bHYPRE_SStructSplit self, int32_t max_iterations, sidl_BaseInterface *_ex)	
	(Optional) Set maximum number of iterations	202
8.2.6	SIDL_C_INLINE_DECL int32_t bHYPRE_SStructSplit_SetLogging (bHYPRE_SStructSplit self,	
	(Optional) Set the logging level, specifying the degree of additional informational data to be accumulated	202
8.2.7	SIDL_C_INLINE_DECL int32_t bHYPRE_SStructSplit_SetPrintLevel (bHYPRE_SStructSplit self,	
	(Optional) Set the print level, specifying the degree of informational data to be printed either to the screen or to a file	202
	SIDL_C_INLINE_DECL int32_t bHYPRE_SStructSplit_GetNumIterations (bHYPRE_SStructSplit self,	
	(Optional) Return the number of iterations taken	
	SIDL_C_INLINE_DECL int32_t bHYPRE_SStructSplit_GetRelResidualNorm (bHYPRE_SStructSplit self, double* norm, sidl_BaseInterface *_ex) (Optional) Return the norm of the relative residual	
8.2.8	SIDL_C_INLINE_DECL int32_t bHYPRE_SStructSplit_SetCommunicator (bHYPRE_SStructSplit self, bHYPRE_MPICommunicator mpi_comm, sidl_BaseInterface *_ex) Set the MPI Communicator	202
8.2.9	SIDL_C_INLINE_DECL void bHYPRE_SStructSplit_Destroy (bHYPRE_SStructSplit self,	
	sidl_BaseInterface *_ex) The Destroy function doesn't necessarily destroy anything	203
	SIDL_C_INLINE_DECL int32_t bHYPRE_SStructSplit_SetIntParameter (bHYPRE_SStructSplit self,	
	Set the int parameter associated with name	
	SIDL_C_INLINE_DECL int32_t bHYPRE_SStructSplit_SetDoubleParameter (bHYPRE_SStructSplit self, const char* name, double value, sidl_BaseInterface *_ex)	
	Set the double parameter associated with name	
	SIDL_C_INLINE_DECL int32_t	

```
bHYPRE_SStructSplit_SetStringParameter ( bHYPRE_SStructSplit self,
                                               const char* name,
                                               const char* value,
                                               sidl_BaseInterface *_ex)
      Set the string parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_SStructSplit_SetIntArray1Parameter ( bHYPRE_SStructSplit
                                                   self, const char* name,
                                                   int32_t* value,
                                                   int32_t nvalues.
                                                   sidl_BaseInterface *_ex)
      Set the int 1-D array parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_SStructSplit_SetIntArray2Parameter ( bHYPRE_SStructSplit
                                                   self, const char* name,
                                                   struct sidl_int_array*
                                                   value,
                                                   sidl_BaseInterface *_ex)
      Set the int 2-D array parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_SStructSplit_SetDoubleArray1Parameter (
                                                       bHYPRE_SStructSplit
                                                       self.
                                                       const char* name,
                                                       double* value,
                                                       int32_t nvalues,
                                                       sidl_BaseInterface
      Set the double 1-D array parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_SStructSplit_SetDoubleArray2Parameter (
                                                       bHYPRE_SStructSplit
                                                       self,
                                                       const char* name,
                                                       struct
                                                       sidl\_double\_array*
                                                       value,
                                                       sidl\_BaseInterface
                                                       *_ex)
      Set the double 2-D array parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_SStructSplit_GetIntValue ( bHYPRE_SStructSplit self,
                                       const char* name, int32_t* value,
                                       sidl_BaseInterface *_ex)
      Set the int parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_SStructSplit_GetDoubleValue ( bHYPRE_SStructSplit self,
                                           const char* name, double* value,
                                           sidl_BaseInterface *_ex)
       Get the double parameter associated with name
SIDL_C_INLINE_DECL int32_t
```

```
bHYPRE_SStructSplit_Setup ( bHYPRE_SStructSplit self,
                               bHYPRE_Vector b, bHYPRE_Vector x,
                               sidl_BaseInterface *_ex)
       (Optional) Do any preprocessing that may be necessary in order to execute
      Apply
SIDL_C_INLINE_DECL int32_t
bHYPRE_SStructSplit_Apply ( bHYPRE_SStructSplit self,
                                bHYPRE_Vector b, bHYPRE_Vector* x,
                                sidl_BaseInterface *_ex)
      Apply the operator to b, returning x
SIDL_C_INLINE_DECL int32_t
bHYPRE_SStructSplit_ApplyAdjoint ( bHYPRE_SStructSplit self,
                                        bHYPRE_Vector b,
                                        bHYPRE_Vector* x,
                                        sidl_BaseInterface *_ex)
      Apply the adjoint of the operator to b, returning x
_ex
       Cast method for interface and class type conversions
void*
bHYPRE_SStructSplit__cast2 (void* obj, const char* type,
                               sidl_BaseInterface *_ex)
      String cast method for interface and class type conversions
SIDL_C_INLINE_DECL void
bHYPRE_SStructSplit_exec ( bHYPRE_SStructSplit self,
                               const char* methodName,
                               sidl_rmi_Call inArgs, sidl_rmi_Return outArgs,
                               sidl_BaseInterface *_ex)
      Select and execute a method by name
SIDL_C_INLINE_DECL_char*
bHYPRE_SStructSplit__getURL ( bHYPRE_SStructSplit self,
                                   sidl_BaseInterface *_ex)
       Get the URL of the Implementation of this object (for RMI)
SIDL_C_INLINE_DECL void
bHYPRE_SStructSplit_raddRef ( bHYPRE_SStructSplit self,
                                   sidl_BaseInterface *_ex)
       On a remote object, addrefs the remote instance
SIDL_C_INLINE_DECL_sidl_bool
bHYPRE_SStructSplit__isRemote ( bHYPRE_SStructSplit self,
                                    sidl_BaseInterface *_ex)
       TRUE if this object is remote, false if local
sidl_bool
bHYPRE_SStructSplit__isLocal ( bHYPRE_SStructSplit self,
                                 sidl_BaseInterface *_ex)
       TRUE if this object is remote, false if local
**_ex
       Cast method for interface and class type conversions
**_{-ex}
       RMI connector function for the class .....
                                                                                 203
```

struct bHYPRE_SStructSplit_object

Symbol "bHYPRE.SStructSplit" (version 1.0.0)

Documentation goes here.

The SStructSplit solver requires a SStruct matrix.

_ 8.2.2 _

bHYPRE_SStructSplit

bHYPRE_SStructSplit_connect (const char *, sidl_BaseInterface *_ex)

RMI connector function for the class.(addrefs)

__ 8.2.3 _____

SIDL_C_INLINE_DECL int32_t bHYPRE_SStructSplit_SetOperator (bHYPRE_SStructSplit self, bHYPRE_Operator A, sidl_BaseInterface *_ex)

Set the operator for the linear system being solved. DEPRECATED. use Create

_ 8.2.4 _

SIDL_C_INLINE_DECL int32_t bHYPRE_SStructSplit_SetTolerance (bHYPRE_SStructSplit self, double tolerance, sidl_BaseInterface *_ex)

(Optional) Set the convergence tolerance. DEPRECATED. use SetDoubleParameter

SIDL_C_INLINE_DECL int32_t bHYPRE_SStructSplit_SetMaxIterations (bHYPRE_SStructSplit self, int32_t max_iterations, sidl_BaseInterface *_ex)

(Optional) Set maximum number of iterations. DEPRECATED use SetIntParameter

___ 8.2.6 _____

(Optional) Set the *logging level*, specifying the degree of additional informational data to be accumulated. Does nothing by default (level = 0). Other levels (if any) are implementation-specific. Must be called before Setup and Apply. DEPRECATED use SetIntParameter

8.2.7

SIDL_C_INLINE_DECL int32_t bHYPRE_SStructSplit_SetPrintLevel (bHYPRE_SStructSplit self, int32_t level, sidl_BaseInterface *_ex)

(Optional) Set the *print level*, specifying the degree of informational data to be printed either to the screen or to a file. Does nothing by default (level=0). Other levels (if any) are implementation-specific. Must be called before Setup and Apply. DEPRECATED use SetIntParameter

__ 8.2.8 _____

SIDL_C_INLINE_DECL int32_t bHYPRE_SStructSplit_SetCommunicator (bHYPRE_SStructSplit self, bHYPRE_MPICommunicator mpi_comm, sidl_BaseInterface *_ex)

Set the MPI Communicator. DEPRECATED, use Create:

SIDL_C_INLINE_DECL void **bHYPRE_SStructSplit_Destroy** (bHYPRE_SStructSplit self, sidl_BaseInterface *_ex)

The Destroy function doesn't necessarily destroy anything. It is just another name for deleteRef. Thus it decrements the object's reference count. The Babel memory management system will destroy the object if the reference count goes to zero.

8.2.10

RMI connector function for the class. (no addref)

9

PreconditionedSolver Interface

```
Names
9.1
            struct bHYPRE_PreconditionedSolver_object
                  Symbol "bHYPRE .....
                                                                                         205
            extern C bHYPRE_PreconditionedSolver
9.2
            bHYPRE_PreconditionedSolver_connect (const char *,
                                                      sidl_BaseInterface *_ex)
                  RMI connector function for the class .....
                                                                                         205
            SIDL_C_INLINE_DECL int32_t
            {\bf bHYPRE\_PreconditionedSolver\_SetPreconditioner} \ (
                                                               bHYPRE_PreconditionedSolver
                                                               self.
                                                               bHYPRE_Solver s,
                                                               sidl_BaseInterface
                                                                *_ex)
                  Set the preconditioner
            SIDL_C_INLINE_DECL int32_t
            bHYPRE_PreconditionedSolver_GetPreconditioner (
                                                                bHYPRE_PreconditionedSolver
                                                                self.
                                                                bHYPRE_Solver* s,
                                                                sidl_BaseInterface
                                                                *_ex)
                  Method: GetPreconditioner[]
            SIDL_C_INLINE_DECL int32_t
            bHYPRE_PreconditionedSolver_Clone ( bHYPRE_PreconditionedSolver
                                                   bHYPRE_PreconditionedSolver* x,
                                                   sidl_BaseInterface *_ex)
                  Method: Clone[]
            _{\mathbf{ex}}
                  Cast method for interface and class type conversions
            void*
            bHYPRE_PreconditionedSolver_cast2 (void* obj, const char* type,
                                                   sidl_BaseInterface *_ex)
                  String cast method for interface and class type conversions
            SIDL_C_INLINE_DECL_void
            bHYPRE_PreconditionedSolver_exec ( bHYPRE_PreconditionedSolver self,
                                                   const char* methodName,
                                                   sidl_rmi_Call inArgs,
                                                  sidl_rmi_Return outArgs,
                                                  sidl\_BaseInterface *\_ex)
                  Select and execute a method by name
            SIDL_C_INLINE_DECL char*
```

bHYPRE_PreconditionedSolver_getURL (bHYPRE_PreconditionedSolver self, sidl_BaseInterface *_ex) Get the URL of the Implementation of this object (for RMI) SIDL_C_INLINE_DECL void bHYPRE_PreconditionedSolver_raddRef (bHYPRE_PreconditionedSolver self, sidl_BaseInterface *_ex) On a remote object, addrefs the remote instance SIDL_C_INLINE_DECL sidl_bool bHYPRE_PreconditionedSolver__isRemote (bHYPRE_PreconditionedSolver self, sidl_BaseInterface *_ex) TRUE if this object is remote, false if local $sidl_bool$ $b HYPRE_Preconditioned Solver__isLocal \left(\begin{array}{c} b HYPRE_Preconditioned Solver \\ \end{array} \right)$ sidl_BaseInterface *_ex) TRUE if this object is remote, false if local $**_{-}ex$ Cast method for interface and class type conversions 9.3 $**_ex$ RMI connector function for the class 206

9.1

struct bHYPRE_PreconditionedSolver_object

Symbol "bHYPRE.PreconditionedSolver" (version 1.0.0)

9.2

extern C bHYPRE_PreconditionedSolver bHYPRE_PreconditionedSolver_connect (const char *, sidl_BaseInterface *_ex)

RMI connector function for the class.(addrefs)

93

RMI connector function for the class. (no addref)

__ 10

Preconditioned Solvers

10.1	PCG Preconditioned Solver	
		207
10.2	GMRES Preconditioned Solver	016
10.3	BiCGSTAB Preconditioned Solver	213
		220
10.4	CGNR Preconditioned Solver	
		227
10	.1	

PCG Preconditioned Solver

Names		
10.1.1	struct bHYPRE_PCGobject Symbol "bHYPRE	211
	_ex Constructor function for the class	
	bHYPRE_PCG bHYPRE_PCGcreateRemote (const char * url, sidl_BaseInterface *_ex) RMI constructor function for the class	
	bHYPRE_PCG bHYPRE_PCGwrapObj (void * data, sidl_BaseInterface *_ex) Wraps up the private data struct pointer (struct bHYPRE_PCGdata) passed in rather than running the constructor	
10.1.2	bHYPRE_PCG bHYPRE_PCGconnect (const char *, sidl_BaseInterface *_ex) RMI connector function for the class	211
	bHYPRE_PCG bHYPRE_MPICommunicator mpi_comm, bHYPRE_Operator A, sidl_BaseInterface *_ex) Method: Create[]	
	SIDL_C_INLINE_DECL int32_t	

	bHYPRE_PCG_SetPreconditioner (bHYPRE_PCG self,	
	bHYPRE_Solver s, sidl_BaseInterface *_ex)	
	Set the preconditioner	
	SIDL_C_INLINE_DECL_int32_t	
	bHYPRE_PCG_GetPreconditioner (bHYPRE_PCG self, bHYPRE_Solver* s, sidl_BaseInterface *_ex)	
	Method: GetPreconditioner[]	
	SIDL_C_INLINE_DECL_int32_t bHYPRE_PCG_Clone (bHYPRE_PCG self,	
	Method: Clone[]	
10.1.3	SIDL_C_INLINE_DECL int32_t bHYPRE_PCG_SetOperator (bHYPRE_PCG self, bHYPRE_Operator A, sidl_BaseInterface *_ex)	
	Set the operator for the linear system being solved	211
10.1.4	SIDL_C_INLINE_DECL int32_t bHYPRE_PCG_SetTolerance (bHYPRE_PCG self, double tolerance, sidl_BaseInterface *_ex)	
	(Optional) Set the convergence tolerance	211
10.1.5	SIDL_C_INLINE_DECL int32_t bHYPRE_PCG_SetMaxIterations (bHYPRE_PCG self,	
	(Optional) Set maximum number of iterations	212
10.1.6	SIDL_C_INLINE_DECL int32_t bHYPRE_PCG_SetLogging (bHYPRE_PCG self, int32_t level,	
	(Optional) Set the logging level, specifying the degree of additional informational data to be accumulated	212
10.1.7	SIDL_C_INLINE_DECL int32_t bHYPRE_PCG_SetPrintLevel (bHYPRE_PCG self, int32_t level,	
	(Optional) Set the print level, specifying the degree of informational data to be printed either to the screen or to a file	212
	SIDL_C_INLINE_DECL int32_t bHYPRE_PCG_GetNumIterations (bHYPRE_PCG self,	
	sidl_BaseInterface *_ex) (Optional) Return the number of iterations taken	
	SIDL_C_INLINE_DECL int32_t bHYPRE_PCG_GetRelResidualNorm (bHYPRE_PCG self, double* norm,	
40.4.2	(Optional) Return the norm of the relative residual	
10.1.8	SIDL_C_INLINE_DECL int32_t	

	bHYPRE_PCG_SetCommunicator (bHYPRE_PCG self,	
	bHYPRE_MPICommunicator mpi_comm,	
	sidl_BaseInterface *_ex)	
	Set the MPI Communicator	212
10.1.9	SIDL_C_INLINE_DECL void bHYPRE_PCG_Destroy (bHYPRE_PCG self, sidl_BaseInterface *_ex) The Destroy function doesn't necessarily destroy anything	213
	SIDL_C_INLINE_DECL int32_t bHYPRE_PCG_SetIntParameter (bHYPRE_PCG self, const char* name, int32_t value, sidl_BaseInterface *_ex)	
	Set the int parameter associated with name	
	SIDL_C_INLINE_DECL int32_t bHYPRE_PCG_SetDoubleParameter (bHYPRE_PCG self,	
	Set the double parameter associated with name	
	SIDL_C_INLINE_DECL int32_t bHYPRE_PCG_SetStringParameter (bHYPRE_PCG self,	
	Set the string parameter associated with name	
	SIDL_C_INLINE_DECL int32_t bHYPRE_PCG_SetIntArray1Parameter (bHYPRE_PCG self,	
	Set the int 1-D array parameter associated with name	
	SIDL_C_INLINE_DECL int32_t bHYPRE_PCG_SetIntArray2Parameter (bHYPRE_PCG self,	
	sidl_BaseInterface *_ex)	
	Set the int 2-D array parameter associated with name	
	SIDL_C_INLINE_DECL int32_t	
	bHYPRE_PCG_SetDoubleArray1Parameter (bHYPRE_PCG self, const char* name, double* value, int32_t nvalues, sidl_BaseInterface *_ex)	
	Set the double 1-D array parameter associated with name	
	SIDL_C_INLINE_DECL int32_t bHYPRE_PCG_SetDoubleArray2Parameter (bHYPRE_PCG self,	
	const char* name, struct sidl_doublearray* value, sidl_BaseInterface *_ex)	
	Set the double 2-D array parameter associated with name	
	SIDL C INLINE DECL. int32 t	

```
b HYPRE\_PCG\_GetIntValue \ ( \ bHYPRE\_PCG \ self, \ const \ char^* \ name,
                               int32_t* value, sidl_BaseInterface *_ex)
      Set the int parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_PCG_GetDoubleValue ( bHYPRE_PCG self, const char* name,
                                   double* value, sidl_BaseInterface *_ex)
      Get the double parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_PCG_Setup ( bHYPRE_PCG self, bHYPRE_Vector b,
                        bHYPRE_Vector x, sidl_BaseInterface *_ex)
       (Optional) Do any preprocessing that may be necessary in order to execute
      Apply
SIDL_C_INLINE_DECL int32_t
bHYPRE_PCG_Apply ( bHYPRE_PCG self, bHYPRE_Vector b,
                        bHYPRE_Vector* x, sidl_BaseInterface *_ex)
      Apply the operator to b, returning x
SIDL_C_INLINE_DECL_int32_t
bHYPRE_PCG_ApplyAdjoint ( bHYPRE_PCG self, bHYPRE_Vector b,
                                bHYPRE_Vector* x, sidl_BaseInterface *_ex)
      Apply the adjoint of the operator to b, returning x
_{\mathbf{ex}}
      Cast method for interface and class type conversions
void*
bHYPRE_PCG__cast2 (void* obj, const char* type, sidl_BaseInterface *_ex)
      String cast method for interface and class type conversions
SIDL_C_INLINE_DECL void
bHYPRE_PCG__exec ( bHYPRE_PCG self, const char* methodName,
                       sidl_rmi_Call inArgs, sidl_rmi_Return outArgs,
                       sidl_BaseInterface *_ex)
      Select and execute a method by name
SIDL_C_INLINE_DECL char*
bHYPRE_PCG__getURL ( bHYPRE_PCG self, sidl_BaseInterface *_ex)
      Get the URL of the Implementation of this object (for RMI)
SIDL_C_INLINE_DECL void
bHYPRE_PCG__raddRef ( bHYPRE_PCG self, sidl_BaseInterface *_ex)
      On a remote object, addrefs the remote instance
SIDL_C_INLINE_DECL sidl_bool
bHYPRE_PCG__isRemote ( bHYPRE_PCG self, sidl_BaseInterface *_ex)
      TRUE if this object is remote, false if local
sidl_bool
bHYPRE_PCG__isLocal ( bHYPRE_PCG self, sidl_BaseInterface *_ex)
       TRUE if this object is remote, false if local
**_ex
      Cast method for interface and class type conversions
**_ex
```

RMI connector function for the class

10.1.10

213

10.1.1

struct bHYPRE_PCG__object

Symbol "bHYPRE.PCG" (version 1.0.0)

 $_$ 10.1.2 $_$

bHYPRE_PCG bHYPRE_PCG__connect (const char *, sidl_BaseInterface *_ex)

RMI connector function for the class.(addrefs)

___ 10.1.3 _____

SIDL_C_INLINE_DECL int32_t **bHYPRE_PCG_SetOperator** (bHYPRE_PCG self, bHYPRE_Operator A, sidl_BaseInterface *_ex)

Set the operator for the linear system being solved. DEPRECATED. use Create

__ 10.1.4 _____

SIDL_C_INLINE_DECL int32_t **bHYPRE_PCG_SetTolerance** (bHYPRE_PCG self, double tolerance, sidl_BaseInterface *_ex)

(Optional) Set the convergence tolerance. DEPRECATED. use SetDoubleParameter

10.1.5

SIDL_C_INLINE_DECL int32_t bHYPRE_PCG_SetMaxIterations (bHYPRE_PCG self, int32_t max_iterations, sidl_BaseInterface *_ex)

(Optional) Set maximum number of iterations. DEPRECATED use SetIntParameter

__ 10.1.6 ____

(Optional) Set the *logging level*, specifying the degree of additional informational data to be accumulated. Does nothing by default (level = 0). Other levels (if any) are implementation-specific. Must be called before Setup and Apply. DEPRECATED use SetIntParameter

10.1.7

(Optional) Set the *print level*, specifying the degree of informational data to be printed either to the screen or to a file. Does nothing by default (level=0). Other levels (if any) are implementation-specific. Must be called before Setup and Apply. DEPRECATED use SetIntParameter

10.1.8

SIDL_C_INLINE_DECL int32_t bHYPRE_PCG_SetCommunicator (bHYPRE_PCG self, bHYPRE_MPICommunicator mpi_comm, sidl_BaseInterface *_ex)

Set the MPI Communicator. DEPRECATED, use Create:

___ 10.1.9 _____

SIDL_C_INLINE_DECL void bHYPRE_PCG_Destroy (bHYPRE_PCG self, sidl_BaseInterface *_ex)

The Destroy function doesn't necessarily destroy anything. It is just another name for deleteRef. Thus it decrements the object's reference count. The Babel memory management system will destroy the object if the reference count goes to zero.

___ 10.1.10 _____

RMI connector function for the class. (no addref)

__ 10.2 ___

GMRES Preconditioned Solver

Names		
10.2.1	struct bHYPRE_GMRESobject Symbol "bHYPRE	217
	_ex	
	Constructor function for the class	
	bHYPRE_GMRES bHYPRE_GMREScreateRemote (const char * url, sidl_BaseInterface *_ex) RMI constructor function for the class	
	bHYPRE_GMRES bHYPRE_GMRESwrapObj (void * data, sidl_BaseInterface *_ex) Wraps up the private data struct pointer (struct bHYPRE_GMRESdata) passed in rather than running the constructor	
10.2.2	bHYPRE_GMRES bHYPRE_GMRESconnect (const char *, sidl_BaseInterface *_ex) RMI connector function for the class	217
	bHYPRE_GMRES bHYPRE_MPICommunicator mpi_comm, bHYPRE_Operator A, sidl_BaseInterface *_ex)	
	Method: Create[]	
	SIDL_C_INLINE_DECL int32_t	

	bHYPRE_GMRES_SetPreconditioner (bHYPRE_GMRES self, bHYPRE_Solver s, sidl_BaseInterface *_ex)	
	Set the preconditioner	
	SIDL_C_INLINE_DECL int32_t bHYPRE_GMRES_GetPreconditioner (bHYPRE_GMRES self, bHYPRE_Solver* s,	
	sidl_BaseInterface *_ex) Method: GetPreconditioner[]	
	SIDL_C_INLINE_DECL_int32_t bHYPRE_GMRES_Clone (bHYPRE_GMRES self, bHYPRE_PreconditionedSolver* x,	
	sidl_BaseInterface *_ex) Method: Clone[]	
10.2.3	SIDL_C_INLINE_DECL_int32_t bHYPRE_GMRES_SetOperator (bHYPRE_GMRES self,	
	bHYPRE_Operator A, sidl_BaseInterface *_ex) Set the operator for the linear system being solved	218
10.2.4	SIDL_C_INLINE_DECL int32_t bHYPRE_GMRES_SetTolerance (bHYPRE_GMRES self,	
	(Optional) Set the convergence tolerance	218
10.2.5	SIDL_C_INLINE_DECL int32_t bHYPRE_GMRES_SetMaxIterations (bHYPRE_GMRES self, int32_t max_iterations,	
	sidl_BaseInterface *_ex) (Optional) Set maximum number of iterations	218
10.2.6	SIDL_C_INLINE_DECL int32_t bHYPRE_GMRES_SetLogging (bHYPRE_GMRES self, int32_t level, sidl_BaseInterface *_ex)	
	(Optional) Set the logging level, specifying the degree of additional informational data to be accumulated	218
10.2.7	SIDL_C_INLINE_DECL int32_t bHYPRE_GMRES_SetPrintLevel (bHYPRE_GMRES self, int32_t level, sidl_BaseInterface *_ex)	
	(Optional) Set the print level, specifying the degree of informational data to be printed either to the screen or to a file	219
	SIDL_C_INLINE_DECL int32_t bHYPRE_GMRES_GetNumIterations (bHYPRE_GMRES self, int32_t* num_iterations, sidl_BaseInterface *_ex)	
	(Optional) Return the number of iterations taken	
	SIDL_C_INLINE_DECL int32_t bHYPRE_GMRES_GetRelResidualNorm (bHYPRE_GMRES self,	
	sidl_BaseInterface *_ex) (Optional) Return the norm of the relative residual	
10.2.8	SIDL_C_INLINE_DECL_int32_t	

	bHYPRE_GMRES_SetCommunicator (bHYPRE_GMRES self,	
	bHYPRE_MPICommunicator	
	mpi_comm, sidl_BaseInterface *_ex) Set the MPI Communicator	21
10.00		2.1
0.2.9	SIDL_C_INLINE_DECL void bHYPRE_GMRES_Destroy (bHYPRE_GMRES self,	
	sidl_BaseInterface *_ex)	
	The Destroy function doesn't necessarily destroy anything	21
	SIDL_C_INLINE_DECL int32_t	
	bHYPRE_GMRES_SetIntParameter (bHYPRE_GMRES self,	
	const char* name, int32_t value,	
	sidl_BaseInterface *_ex)	
	Set the int parameter associated with name	
	SIDL_C_INLINE_DECL int32_t	
	bHYPRE_GMRES_SetDoubleParameter (bHYPRE_GMRES self, const char* name, double value,	
	sidl_BaseInterface *_ex)	
	Set the double parameter associated with name	
	SIDL_C_INLINE_DECL int32_t	
	bHYPRE_GMRES_SetStringParameter (bHYPRE_GMRES self,	
	const char* name,	
	const char* value,	
	sidl_BaseInterface *_ex)	
	Set the string parameter associated with name	
	SIDL_C_INLINE_DECL int32_t	
	bHYPRE_GMRES_SetIntArray1Parameter (bHYPRE_GMRES self, const char* name,	
	$int32t^*$ value,	
	int32_t nvalues,	
	sidl_BaseInterface *_ex)	
	Set the int 1-D array parameter associated with name	
	SIDL_C_INLINE_DECL int32_t	
	bHYPRE_GMRES_SetIntArray2Parameter (bHYPRE_GMRES self,	
	const char* name, struct sidl_int_array* value,	
	sidl_BaseInterface *_ex)	
	Set the int 2-D array parameter associated with name	
	SIDL_C_INLINE_DECL int32_t	
	bHYPRE_GMRES_SetDoubleArray1Parameter (bHYPRE_GMRES self,	
	const char* name,	
	double* value,	
	int32_t nvalues,	
	sidl_BaseInterface *_ex) Set the double 1-D array parameter associated with name	
	SIDL_C_INLINE_DECL int32_t	

```
bHYPRE_GMRES_SetDoubleArray2Parameter ( bHYPRE_GMRES self.
                                                 const char* name, struct
                                                 sidl_double_array* value,
                                                 sidl_BaseInterface *_ex)
      Set the double 2-D array parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_GMRES_GetIntValue ( bHYPRE_GMRES self,
                                 const char* name, int32_t* value,
                                 sidl_BaseInterface *_ex)
      Set the int parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_GMRES_GetDoubleValue ( bHYPRE_GMRES self,
                                      const char* name, double* value,
                                      sidl_BaseInterface *_ex)
      Get the double parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_GMRES_Setup ( bHYPRE_GMRES self, bHYPRE_Vector b,
                           bHYPRE_Vector x, sidl_BaseInterface *_ex)
      (Optional) Do any preprocessing that may be necessary in order to execute
      Apply
SIDL_C_INLINE_DECL int32_t
bHYPRE_GMRES_Apply ( bHYPRE_GMRES self, bHYPRE_Vector b,
                           bHYPRE_Vector* x, sidl_BaseInterface *_ex)
      Apply the operator to b, returning x
SIDL_C_INLINE_DECL int32_t
bHYPRE_GMRES_ApplyAdjoint ( bHYPRE_GMRES self,
                                   bHYPRE_Vector b, bHYPRE_Vector* x,
                                   sidl_BaseInterface *_ex)
      Apply the adjoint of the operator to b, returning x
_{\mathbf{ex}}
      Cast method for interface and class type conversions
void*
bHYPRE_GMRES__cast2 (void* obj, const char* type,
                           sidl_BaseInterface *_ex)
      String cast method for interface and class type conversions
SIDL_C_INLINE_DECL_void
bHYPRE_GMRES__exec ( bHYPRE_GMRES self,
                          const char* methodName, sidl_rmi_Call inArgs,
                          sidl_rmi_Return outArgs, sidl_BaseInterface *_ex)
      Select and execute a method by name
SIDL_C_INLINE_DECL char*
bHYPRE_GMRES__getURL ( bHYPRE_GMRES self.
                              sidl_BaseInterface *_ex)
      Get the URL of the Implementation of this object (for RMI)
SIDL_C_INLINE_DECL_void
bHYPRE_GMRES__raddRef ( bHYPRE_GMRES self,
                              sidl_BaseInterface *_ex)
      On a remote object, address the remote instance
SIDL_C_INLINE_DECL_sidl_bool
```

bHYPRE_GMRES__isRemote (bHYPRE_GMRES self. sidl_BaseInterface *_ex) TRUE if this object is remote, false if local

sidl_bool

${\bf bHYPRE_GMRES__isLocal} \ (\ \ {\bf bHYPRE_GMRES} \ {\bf self},$ sidl_BaseInterface *_ex)

TRUE if this object is remote, false if local

 $**_ex$

Cast method for interface and class type conversions

 $\mathbf{**}_{-}\mathbf{ex}$ 10.2.10

> RMI connector function for the class 219

10.2.1

struct bHYPRE_GMRES__object

Symbol "bHYPRE.GMRES" (version 1.0.0)

Objects of this type can be cast to PreconditionedSolver objects using the __cast methods.

RDF: Documentation goes here.

The regular GMRES solver calls Babel-interface matrix and vector functions. The HGMRES solver calls HYPRE interface functions. The regular solver will work with any consistent matrix, vector, and preconditioner classes. The HGMRES solver will work with the more common combinations.

The HGMRES solver checks whether the matrix, vectors, and preconditioner are of known types, and will not work with any other types. Presently, the recognized data types are: matrix, vector: IJParCSRMatrix, IJParCSRVector preconditioner: BoomerAMG, ParCSRDiagScale

bHYPRE_GMRES bHYPRE_GMRES__connect (const char *, sidl_BaseInterface *_ex)

10.2.3

SIDL_C_INLINE_DECL int32_t bHYPRE_GMRES_SetOperator (bHYPRE_GMRES self, bHYPRE_Operator A, sidl_BaseInterface *_ex)

Set the operator for the linear system being solved. DEPRECATED. use Create

10.2.4

SIDL_C_INLINE_DECL int32_t bHYPRE_GMRES_SetTolerance (bHYPRE_GMRES self, double tolerance, sidl_BaseInterface *_ex)

(Optional) Set the convergence tolerance. DEPRECATED. use SetDoubleParameter

_ 10.2.5 _____

SIDL_C_INLINE_DECL int32_t bHYPRE_GMRES_SetMaxIterations (bHYPRE_GMRES self, int32_t max_iterations, sidl_BaseInterface *_ex)

(Optional) Set maximum number of iterations. DEPRECATED use SetIntParameter

__ 10.2.6 _____

SIDL_C_INLINE_DECL int32_t bHYPRE_GMRES_SetLogging (bHYPRE_GMRES self, int32_t level, sidl_BaseInterface *_ex)

(Optional) Set the $logging\ level$, specifying the degree of additional informational data to be accumulated. Does nothing by default (level = 0). Other levels (if any) are implementation-specific. Must be called before Setup and Apply. DEPRECATED use SetIntParameter

10.2.7

SIDL_C_INLINE_DECL int32_t bHYPRE_GMRES_SetPrintLevel (bHYPRE_GMRES self, int32_t level, sidl_BaseInterface *_ex)

(Optional) Set the *print level*, specifying the degree of informational data to be printed either to the screen or to a file. Does nothing by default (level=0). Other levels (if any) are implementation-specific. Must be called before Setup and Apply. DEPRECATED use SetIntParameter

___ 10.2.8 _____

SIDL_C_INLINE_DECL int32_t bHYPRE_GMRES_SetCommunicator (bHYPRE_GMRES self, bHYPRE_MPICommunicator mpi_comm, sidl_BaseInterface *_ex)

Set the MPI Communicator. DEPRECATED, use Create:

_ 10.2.9 __

SIDL_C_INLINE_DECL void **bHYPRE_GMRES_Destroy** (bHYPRE_GMRES self, sidl_BaseInterface *_ex)

The Destroy function doesn't necessarily destroy anything. It is just another name for deleteRef. Thus it decrements the object's reference count. The Babel memory management system will destroy the object if the reference count goes to zero.

10.2.10

struct bHYPRE_GMRES__object* bHYPRE_GMRES__connectI const char * url sidl_bool ar struct sidl_BaseInterface__object **_ex

_ 10.3 _

BiCGSTAB Preconditioned Solver

Names		
10.3.1	struct bHYPRE_BiCGSTAB_object Symbol "bHYPRE	224
	_ex	
	Constructor function for the class	
	bHYPRE_BiCGSTAB	
	bHYPRE_BiCGSTABcreateRemote (const char * url, sidl_BaseInterface *_ex)	
	RMI constructor function for the class	
	bHYPRE_BiCGSTAB bHYPRE_BiCGSTABwrapObj (void * data, sidl_BaseInterface *_ex) Wraps up the private data struct pointer (struct bHYPRE_BiCGSTABdata) passed in rather than running the constructor	
10.3.2	bHYPRE_BiCGSTAB bHYPRE_BiCGSTABconnect (const char *, sidl_BaseInterface *_ex) RMI connector function for the class	224
	bHYPRE_BiCGSTAB bHYPRE_BiCGSTAB_Create (bHYPRE_MPICommunicator mpi_comm, bHYPRE_Operator A, sidl_BaseInterface *_ex)	
	Method: Create[]	
	SIDL_C_INLINE_DECL_int32_t bHYPRE_BiCGSTAB_SetPreconditioner (bHYPRE_BiCGSTAB self, bHYPRE_Solver s,	
	sidl_BaseInterface *_ex)	
	Set the preconditioner	
	SIDL_C_INLINE_DECL int32_t bHYPRE_BiCGSTAB_GetPreconditioner (bHYPRE_BiCGSTAB self, bHYPRE_Solver* s, sidl_BaseInterface *_ex)	
	$Method: \ GetPreconditioner[]$	
	SIDL_C_INLINE_DECL int32_t bHYPRE_BiCGSTAB_Clone (bHYPRE_BiCGSTAB self, bHYPRE_PreconditionedSolver* x, sidl_BaseInterface *_ex)	
	Method: Clone[]	
10.3.3	SIDL_C_INLINE_DECL int32_t bHYPRE_BiCGSTAB_SetOperator (bHYPRE_BiCGSTAB self, bHYPRE_Operator A, sidl_BaseInterface *_ex)	
	Set the operator for the linear system being solved	224
10.3.4	SIDL_C_INLINE_DECL int32_t	

	bHYPRE_BiCGSTAB_SetTolerance (bHYPRE_BiCGSTAB self, double tolerance, sidl_BaseInterface *_ex)	
	(Optional) Set the convergence tolerance	225
10.3.5	SIDL_C_INLINE_DECL int32_t bHYPRE_BiCGSTAB_SetMaxIterations (bHYPRE_BiCGSTAB self, int32_t max_iterations, sidl_BaseInterface *_ex)	
	(Optional) Set maximum number of iterations	225
10.3.6	SIDL_C_INLINE_DECL int32_t bHYPRE_BiCGSTAB_SetLogging (bHYPRE_BiCGSTAB self,	
	(Optional) Set the logging level, specifying the degree of additional informational data to be accumulated	225
10.3.7	SIDL_C_INLINE_DECL int32_t bHYPRE_BiCGSTAB_SetPrintLevel (bHYPRE_BiCGSTAB self,	
	(Optional) Set the print level, specifying the degree of informational data to be printed either to the screen or to a file	225
	SIDL_C_INLINE_DECL int32_t bHYPRE_BiCGSTAB_GetNumIterations (bHYPRE_BiCGSTAB self, int32_t* num_iterations, sidl_BaseInterface *_ex)	
	(Optional) Return the number of iterations taken	
	SIDL_C_INLINE_DECL int32_t bHYPRE_BiCGSTAB_GetRelResidualNorm (bHYPRE_BiCGSTAB self, double* norm, sidl_BaseInterface *_ex) (Optional) Return the norm of the relative residual	
10.3.8	SIDL_C_INLINE_DECL int32_t bHYPRE_BiCGSTAB_SetCommunicator (bHYPRE_BiCGSTAB self, bHYPRE_MPICommunicator mpi_comm,	
	sidl_BaseInterface *_ex)	
	Set the MPI Communicator	226
10.3.9	SIDL_C_INLINE_DECL void bHYPRE_BiCGSTAB_Destroy (bHYPRE_BiCGSTAB self, sidl_BaseInterface *_ex)	
	The Destroy function doesn't necessarily destroy anything	226
	SIDL_C_INLINE_DECL int32_t bHYPRE_BiCGSTAB_SetIntParameter (bHYPRE_BiCGSTAB self,	
	Set the int parameter associated with name	
	SIDL_C_INLINE_DECL_int32_t	

```
bHYPRE_BiCGSTAB_SetDoubleParameter ( bHYPRE_BiCGSTAB self,
                                              const char* name,
                                              double value,
                                              sidl_BaseInterface *_ex)
      Set the double parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_BiCGSTAB_SetStringParameter ( bHYPRE_BiCGSTAB self,
                                             const char* name,
                                             const char* value.
                                             sidl_BaseInterface *_ex)
      Set the string parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_BiCGSTAB_SetIntArray1Parameter ( bHYPRE_BiCGSTAB
                                                 self, const char* name,
                                                 int32_t* value,
                                                 int32_t nvalues.
                                                 sidl_BaseInterface *_ex)
      Set the int 1-D array parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_BiCGSTAB_SetIntArray2Parameter ( bHYPRE_BiCGSTAB
                                                 self, const char* name,
                                                 struct sidl_int_array*
                                                 value,
                                                 sidl_BaseInterface *_ex)
      Set the int 2-D array parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_BiCGSTAB_SetDoubleArray1Parameter (
                                                     bHYPRE_BiCGSTAB
                                                     self.
                                                     const char* name,
                                                     double* value,
                                                     int32_t nvalues,
                                                     sidl\_BaseInterface
                                                     *_ex)
      Set the double 1-D array parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_BiCGSTAB_SetDoubleArray2Parameter (
                                                     bHYPRE_BiCGSTAB
                                                     self.
                                                     const char* name,
                                                     struct
                                                     sidl_double_array*
                                                     value.
                                                     sidl\_BaseInterface
                                                     *_ex)
      Set the double 2-D array parameter associated with name
```

SIDL_C_INLINE_DECL int32_t

```
bHYPRE_BiCGSTAB_GetIntValue ( bHYPRE_BiCGSTAB self,
                                    const char* name, int32_t* value,
                                    sidl_BaseInterface *_ex)
      Set the int parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_BiCGSTAB_GetDoubleValue ( bHYPRE_BiCGSTAB self,
                                         const char* name, double* value,
                                         sidl_BaseInterface *_ex)
      Get the double parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_BiCGSTAB_Setup ( bHYPRE_BiCGSTAB self,
                              bHYPRE_Vector b, bHYPRE_Vector x,
                              sidl\_BaseInterface *\_ex)
      (Optional) Do any preprocessing that may be necessary in order to execute
      Apply
SIDL_C_INLINE_DECL int32_t
bHYPRE_BiCGSTAB_Apply ( bHYPRE_BiCGSTAB self,
                              bHYPRE_Vector b, bHYPRE_Vector* x,
                              sidl_BaseInterface *_ex)
      Apply the operator to b, returning x
SIDL_C_INLINE_DECL_int32_t
bHYPRE_BiCGSTAB_ApplyAdjoint ( bHYPRE_BiCGSTAB self,
                                      bHYPRE_Vector b,
                                      bHYPRE_Vector* x,
                                      sidl_BaseInterface *_ex)
      Apply the adjoint of the operator to b, returning x
_ex
      Cast method for interface and class type conversions
void*
bHYPRE_BiCGSTAB__cast2 (void* obj, const char* type,
                              sidl_BaseInterface *_ex)
      String cast method for interface and class type conversions
SIDL_C_INLINE_DECL void
bHYPRE_BiCGSTAB__exec ( bHYPRE_BiCGSTAB self,
                             const char* methodName, sidl_rmi_Call inArgs,
                             sidl_rmi_Return outArgs,
                             sidl_BaseInterface *_ex)
      Select and execute a method by name
SIDL_C_INLINE_DECL char*
bHYPRE_BiCGSTAB__getURL ( bHYPRE_BiCGSTAB self,
                                 sidl_BaseInterface *_ex)
      Get the URL of the Implementation of this object (for RMI)
SIDL_C_INLINE_DECL void
bHYPRE_BiCGSTAB __raddRef ( bHYPRE_BiCGSTAB self,
```

sidl_BaseInterface *_ex)

SIDL_C_INLINE_DECL sidl_bool

On a remote object, addrefs the remote instance

bHYPRE_BiCGSTAB__isRemote (bHYPRE_BiCGSTAB self,

sidl_BaseInterface *_ex)

TRUE if this object is remote, false if local

sidl_bool

${\bf bHYPRE_BiCGSTAB_isLocal} \ (\ \ {\bf bHYPRE_BiCGSTAB} \ {\bf self},$

sidl_BaseInterface *_ex)

TRUE if this object is remote, false if local

 $**_{-}ex$

Cast method for interface and class type conversions

10.3.10 **_ex

10.3.1 _

struct bHYPRE_BiCGSTAB__object

Symbol "bHYPRE.BiCGSTAB" (version 1.0.0)

Objects of this type can be cast to PreconditionedSolver objects using the __cast methods.

RDF: Documentation goes here.

BiCGSTAB solver calls Babel-interface functions

_ 10.3.2 _

bHYPRE_BiCGSTAB bHYPRE_BiCGSTAB__connect (const char *, sidl_BaseInterface *_ex)

RMI connector function for the class.(addrefs)

_ 10.3.3 ____

SIDL_C_INLINE_DECL int32_t bHYPRE_BiCGSTAB_SetOperator (bHYPRE_BiCGSTAB self, bHYPRE_Operator A, sidl_BaseInterface *_ex)

Set the operator for the linear system being solved. DEPRECATED. use Create

10.3.4

SIDL_C_INLINE_DECL int32_t bHYPRE_BiCGSTAB_SetTolerance (bHYPRE_BiCGSTAB self, double tolerance, sidl_BaseInterface *_ex)

(Optional) Set the convergence tolerance. DEPRECATED. use SetDoubleParameter

 $_{-}$ 10.3.5 $_{-}$

SIDL_C_INLINE_DECL int32_t bHYPRE_BiCGSTAB_SetMaxIterations (bHYPRE_BiCGSTAB self, int32_t max_iterations, sidl_BaseInterface *_ex)

(Optional) Set maximum number of iterations. DEPRECATED use SetIntParameter

_ 10.3.6 ___

SIDL_C_INLINE_DECL int32_t bHYPRE_BiCGSTAB_SetLogging (bHYPRE_BiCGSTAB self, int32_t level, sidl_BaseInterface *_ex)

(Optional) Set the $logging\ level$, specifying the degree of additional informational data to be accumulated. Does nothing by default (level = 0). Other levels (if any) are implementation-specific. Must be called before Setup and Apply. DEPRECATED use SetIntParameter

_ 10.3.7 ____

SIDL_C_INLINE_DECL int32_t bHYPRE_BiCGSTAB_SetPrintLevel (bHYPRE_BiCGSTAB self, int32_t level, sidl_BaseInterface *_ex)

(Optional) Set the *print level*, specifying the degree of informational data to be printed either to the screen or to a file. Does nothing by default (level=0). Other levels (if any) are implementation-specific. Must be called before Setup and Apply. DEPRECATED use SetIntParameter

10.3.8

SIDL_C_INLINE_DECL int32_t bHYPRE_BiCGSTAB_SetCommunicator (bHYPRE_BiCGSTAB self, bHYPRE_MPICommunicator mpi_comm, sidl_BaseInterface *_ex)

Set the MPI Communicator. DEPRECATED, use Create:

_ 10.3.9 _

The Destroy function doesn't necessarily destroy anything. It is just another name for deleteRef. Thus it decrements the object's reference count. The Babel memory management system will destroy the object if the reference count goes to zero.

10.3.10

struct bHYPRE_BiCGSTAB__object* bHYPRE_BiCGSTAB__connectI const char * url sidl_bool ar struct sidl_BaseInterface__object **_ex

10.4

CGNR Preconditioned Solver

Names		
10.4.1	struct bHYPRE_CGNRobject Symbol "bHYPRE	230
	_ex Constructor function for the class	
	bHYPRE_CGNR bHYPRE_CGNRcreateRemote (const char * url, sidl_BaseInterface *_ex) RMI constructor function for the class	
	bHYPRE_CGNR bHYPRE_CGNRwrapObj (void * data, sidl_BaseInterface *_ex) Wraps up the private data struct pointer (struct bHYPRE_CGNRdata) passed in rather than running the constructor	
10.4.2	bHYPRE_CGNR bHYPRE_CGNRconnect (const char *, sidl_BaseInterface *_ex) RMI connector function for the class	231
	bHYPRE_CGNR bHYPRE_CGNR_Create (bHYPRE_MPICommunicator mpi_comm,	
	SIDL_C_INLINE_DECL int32_t bHYPRE_CGNR_SetPreconditioner (bHYPRE_CGNR self,	
	Set the preconditioner SIDL_C_INLINE_DECL int32_t bHYPRE_CGNR_GetPreconditioner (bHYPRE_CGNR self, bHYPRE_Solver* s, sidl_BaseInterface *_ex) Method: GetPreconditioner[]	
	SIDL_C_INLINE_DECL_int32_t bHYPRE_CGNR_Clone (bHYPRE_CGNR self,	
10.4.3	SIDL_C_INLINE_DECL int32_t bHYPRE_CGNR_SetOperator (bHYPRE_CGNR self, bHYPRE_Operator A, sidl_BaseInterface *_ex)	
	Set the operator for the linear system being solved	231
10.4.4	SIDL_C_INLINE_DECL int32_t	

	bHYPRE_CGNR_SetTolerance (bHYPRE_CGNR self, double tolerance, sidl_BaseInterface *_ex)	
	(Optional) Set the convergence tolerance	231
10.4.5	SIDL_C_INLINE_DECL int32_t bHYPRE_CGNR_SetMaxIterations (bHYPRE_CGNR self,	
	(Optional) Set maximum number of iterations	231
10.4.6	SIDL_C_INLINE_DECL int32_t bHYPRE_CGNR_SetLogging (bHYPRE_CGNR self, int32_t level,	
	(Optional) Set the logging level, specifying the degree of additional informational data to be accumulated	232
10.4.7	SIDL_C_INLINE_DECL int32_t bHYPRE_CGNR_SetPrintLevel (bHYPRE_CGNR self, int32_t level,	
	(Optional) Set the print level, specifying the degree of informational data to be printed either to the screen or to a file	232
	SIDL_C_INLINE_DECL int32_t bHYPRE_CGNR_GetNumIterations (bHYPRE_CGNR self,	
	(Optional) Return the number of iterations taken	
	SIDL_C_INLINE_DECL int32_t bHYPRE_CGNR_GetRelResidualNorm (bHYPRE_CGNR self, double* norm, sidl_BaseInterface *_ex) (Optional) Return the norm of the relative residual	
10.4.8	SIDL_C_INLINE_DECL int32_t bHYPRE_CGNR_SetCommunicator (bHYPRE_CGNR self,	020
10.40	Set the MPI Communicator	232
10.4.9	SIDL_C_INLINE_DECL void bHYPRE_CGNR_Destroy (bHYPRE_CGNR self, sidl_BaseInterface *_ex) The Destroy function doesn't necessarily destroy anything	232
	SIDL_C_INLINE_DECL int32_t bHYPRE_CGNR_SetIntParameter (bHYPRE_CGNR self,	
	Set the int parameter associated with name	
	SIDL_C_INLINE_DECL int32_t bHYPRE_CGNR_SetDoubleParameter (bHYPRE_CGNR self,	
	Set the double parameter associated with name	
	SIDL_C_INLINE_DECL int32_t	

```
bHYPRE_CGNR_SetStringParameter ( bHYPRE_CGNR self,
                                        const char* name,
                                        const char* value,
                                        sidl_BaseInterface *_ex)
      Set the string parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_CGNR_SetIntArray1Parameter ( bHYPRE_CGNR self,
                                            const char* name,
                                            int32_t* value, int32_t nvalues,
                                            sidl_BaseInterface *_ex)
      Set the int 1-D array parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_CGNR_SetIntArray2Parameter ( bHYPRE_CGNR self,
                                            const char* name,
                                            struct sidl_int_array* value,
                                            sidl_BaseInterface *_ex)
      Set the int 2-D array parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_CGNR_SetDoubleArray1Parameter ( bHYPRE_CGNR self,
                                                const char* name,
                                                double* value,
                                                int32_t nvalues.
                                                sidl_BaseInterface *_ex)
      Set the double 1-D array parameter associated with name
SIDL_C_INLINE_DECL_int32_t
bHYPRE_CGNR_SetDoubleArray2Parameter ( bHYPRE_CGNR self,
                                                const char* name, struct
                                                sidl_double_array* value,
                                                sidl_BaseInterface *_ex)
      Set the double 2-D array parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_CGNR_GetIntValue ( bHYPRE_CGNR self, const char* name,
                                int32_t* value, sidl_BaseInterface *_ex)
      Set\ the\ int\ parameter\ associated\ with\ {\tt name}
SIDL_C_INLINE_DECL int32_t
bHYPRE_CGNR_GetDoubleValue ( bHYPRE_CGNR self,
                                    const char* name, double* value,
                                    sidl_BaseInterface *_ex)
      Get the double parameter associated with name
SIDL_C_INLINE_DECL int32_t
bHYPRE_CGNR_Setup ( bHYPRE_CGNR self, bHYPRE_Vector b,
                         bHYPRE_Vector x, sidl_BaseInterface *_ex)
       (Optional) Do any preprocessing that may be necessary in order to execute
      Apply
SIDL_C_INLINE_DECL int32_t
bHYPRE_CGNR_Apply (bHYPRE_CGNR self, bHYPRE_Vector b,
                          bHYPRE_Vector* x, sidl_BaseInterface *_ex)
      Apply the operator to b, returning x
SIDL_C_INLINE_DECL int32_t
```

bHYPRE_CGNR_ApplyAdjoint (bHYPRE_CGNR self,

```
bHYPRE_Vector b, bHYPRE_Vector* x,
                                              sidl_BaseInterface *_ex)
                   Apply the adjoint of the operator to b, returning x
            _{\mathbf{ex}}
                   Cast method for interface and class type conversions
            void*
            bHYPRE_CGNR__cast2 ( void* obj, const char* type,
                                      sidl_BaseInterface *_ex)
                   String cast method for interface and class type conversions
            SIDL_C_INLINE_DECL void
            bHYPRE_CGNR__exec ( bHYPRE_CGNR self, const char* methodName,
                                     sidl_rmi_Call inArgs, sidl_rmi_Return outArgs,
                                     sidl_BaseInterface *_ex)
                   Select and execute a method by name
            SIDL_C_INLINE_DECL char*
            bHYPRE_CGNR__getURL ( bHYPRE_CGNR self, sidl_BaseInterface *_ex)
                   Get the URL of the Implementation of this object (for RMI)
            SIDL_C_INLINE_DECL void
            bHYPRE_CGNR__raddRef ( bHYPRE_CGNR self, sidl_BaseInterface *_ex)
                   On a remote object, addrefs the remote instance
            SIDL_C_INLINE_DECL_sidl_bool
            bHYPRE_CGNR__isRemote ( bHYPRE_CGNR self,
                                          sidl_BaseInterface *_ex)
                   TRUE if this object is remote, false if local
            sidl_bool
            bHYPRE_CGNR__isLocal ( bHYPRE_CGNR self, sidl_BaseInterface *_ex)
                   TRUE if this object is remote, false if local
            **_ex
                   Cast method for interface and class type conversions
10.4.10
            **_ex
                                                                                             233
                   RMI connector function for the class .....
```

10.4.1

struct bHYPRE_CGNR__object

Symbol "bHYPRE.CGNR" (version 1.0.0)

Objects of this type can be cast to PreconditionedSolver objects using the __cast methods.

RDF: Documentation goes here.

CGNR solver calls Babel-interface functions

__ 10.4.2 ___

bHYPRE_CGNR bHYPRE_CGNR__connect (const char *, sidl_BaseInterface *_ex)

RMI connector function for the class.(addrefs)

___ 10.4.3 _____

Set the operator for the linear system being solved. DEPRECATED. use Create

_ 10.4.4 _

SIDL_C_INLINE_DECL int32_t bHYPRE_CGNR_SetTolerance (bHYPRE_CGNR self, double tolerance, sidl_BaseInterface *_ex)

(Optional) Set the convergence tolerance. DEPRECATED. use SetDoubleParameter

10.4.5

SIDL_C_INLINE_DECL int32_t bHYPRE_CGNR_SetMaxIterations (bHYPRE_CGNR self, int32_t max_iterations, sidl_BaseInterface *_ex)

(Optional) Set maximum number of iterations. DEPRECATED use SetIntParameter

10.4.6

SIDL_C_INLINE_DECL int32_t bHYPRE_CGNR_SetLogging (bHYPRE_CGNR self, int32_t level, sidl_BaseInterface *_ex)

(Optional) Set the *logging level*, specifying the degree of additional informational data to be accumulated. Does nothing by default (level = 0). Other levels (if any) are implementation-specific. Must be called before Setup and Apply. DEPRECATED use SetIntParameter

__ 10.4.7 ____

(Optional) Set the *print level*, specifying the degree of informational data to be printed either to the screen or to a file. Does nothing by default (level=0). Other levels (if any) are implementation-specific. Must be called before Setup and Apply. DEPRECATED use SetIntParameter

_ 10.4.8 ____

SIDL_C_INLINE_DECL int32_t bHYPRE_CGNR_SetCommunicator (bHYPRE_CGNR self, bHYPRE_MPICommunicator mpi_comm, sidl_BaseInterface *_ex)

Set the MPI Communicator. DEPRECATED, use Create:

_ 10.4.9 _

SIDL_C_INLINE_DECL void bHYPRE_CGNR_Destroy (bHYPRE_CGNR self, sidl_BaseInterface *_ex)

The Destroy function doesn't necessarily destroy anything. It is just another name for deleteRef. Thus it decrements the object's reference count. The Babel memory management system will destroy the object if the reference count goes to zero.

10.4.10

11 Other

11		1
Otł	ner	
Names 11.1	MPI Communicator	234
	I.1	
Names 11.1.1	struct bHYPRE_MPICommunicator_object Symbol "bHYPRE	236
	_ex	
	Constructor function for the class bHYPRE_MPICommunicator bHYPRE_MPICommunicator_createRemote (const char * url, sidl_BaseInterface *_ex)	
	RMI constructor function for the class	
	bHYPRE_MPICommunicator bHYPRE_MPICommunicator_wrapObj (void * data,	
11.1.2	bHYPRE_MPICommunicator bHYPRE_MPICommunicator_connect (const char *, sidl_BaseInterface *_ex)	
	RMI connector function for the class	236
	bHYPRE_MPICommunicator bHYPRE_MPICommunicator_CreateC (void* mpi_comm, sidl_BaseInterface *_ex)	
	Method: CreateC[] bHYPRE_MPICommunicator	
	bHYPRE_MPICommunicator_CreateF (void* mpi_comm, sidl_BaseInterface *_ex)	
	Method: CreateF[]	
	bHYPRE_MPICommunicator	

11 Other

	$Method: \ Create_MPICommWorld[]$	
11.1.3	SIDL_C_INLINE_DECL void bHYPRE_MPICommunicator_Destroy (bHYPRE_MPICommunicator self, sidl_BaseInterface *_ex)	
	The Destroy function doesn't necessarily destroy anything	236
	_ex Cast method for interface and class type conversions	
	void^*	
	bHYPRE_MPICommunicator_cast2 (void* obj, const char* type, sidl_BaseInterface *_ex)	
	String cast method for interface and class type conversions	
	SIDL_C_INLINE_DECL void bHYPRE_MPICommunicator_exec (bHYPRE_MPICommunicator self, const char* methodName, sidl_rmi_Call inArgs, sidl_rmi_Return outArgs, sidl_BaseInterface *_ex)	
	Select and execute a method by name	
	SIDL_C_INLINE_DECL char* bHYPRE_MPICommunicatorgetURL (bHYPRE_MPICommunicator self, sidl_BaseInterface *_ex) Get the URL of the Implementation of this object (for RMI)	
	SIDL_C_INLINE_DECL_void	
	bHYPRE_MPICommunicator_raddRef (bHYPRE_MPICommunicator self, sidl_BaseInterface *_ex)	
	On a remote object, addrefs the remote instance	
	SIDL_C_INLINE_DECL sidl_bool bHYPRE_MPICommunicatorisRemote (bHYPRE_MPICommunicator self, sidl_BaseInterface *_ex)	
	TRUE if this object is remote, false if local	
	sidl_bool bHYPRE_MPICommunicatorisLocal (bHYPRE_MPICommunicator self,	
	sidl_BaseInterface *_ex) TRUE if this object is remote, false if local	
	**_ex	
	Cast method for interface and class type conversions	
11.1.4	**_ex RMI connector function for the class	236

236

11 Other

_ 11.1.1 _

struct bHYPRE_MPICommunicator_object

Symbol "bHYPRE.MPICommunicator" (version 1.0.0)

MPICommunicator class - two general Create functions: use CreateC if called from C code, CreateF if called from Fortran code. - Create_MPICommWorld will create a MPICommunicator to represent MPI_Comm_World, and can be called from any language.

11.1.2

bHYPRE_MPICommunicator bHYPRE_MPICommunicator_connect (const char *, sidl_BaseInterface *_ex)

RMI connector function for the class.(addrefs)

11.1.3

SIDL_C_INLINE_DECL void bHYPRE_MPICommunicator_Destroy (bHYPRE_MPICommunicator self, sidl_BaseInterface *_ex)

The Destroy function doesn't necessarily destroy anything. It is just another name for deleteRef. Thus it decrements the object's reference count. The Babel memory management system will destroy the object if the reference count goes to zero.

11.1.4

struct bHYPRE_MPICommunicator_object* bHYPRE_MPICommunicator_connectI const char * url sidl_bool ar struct sidl_BaseInterface_object **-ex

12Struct Grid, etc. Names 12.1 Struct Grid 237 12.2 Struct Stencil 240 Struct Grid Names 12.1.1 struct bHYPRE_StructGrid__object Symbol "bHYPRE 239 $_{\mathbf{ex}}$ Constructor function for the class bHYPRE_StructGrid bHYPRE_StructGrid__createRemote (const char * url, sidl_BaseInterface *_ex) RMI constructor function for the class bHYPRE_StructGrid bHYPRE_StructGrid__wrapObj (void * data, sidl_BaseInterface *_ex) $Wraps\ up\ the\ private\ data\ struct\ pointer\ (struct\ bHYPRE_StructGrid__data)$ passed in rather than running the constructor 12.1.2 bHYPRE_StructGrid bHYPRE_StructGrid__connect (const char *, sidl_BaseInterface *_ex) RMI connector function for the class 239 bHYPRE_StructGrid bHYPRE_StructGrid_Create (bHYPRE_MPICommunicator mpi_comm, int32_t dim, sidl_BaseInterface *_ex) Method: Create[] 12.1.3 SIDL_C_INLINE_DECL int32_t

bHYPRE_StructGrid_SetCommunicator (bHYPRE_StructGrid self,

Set the MPI Communicator

 $b HYPRE_MPIC ommunicator$

sidl_BaseInterface *_ex)

mpi_comm,

SIDL_C_INLINE_DECL void

12.1.4

239

```
bHYPRE_StructGrid_Destroy ( bHYPRE_StructGrid self,
                                sidl_BaseInterface *_ex)
       The Destroy function doesn't necessarily destroy anything ......
                                                                                 240
SIDL_C_INLINE_DECL int32_t
bHYPRE_StructGrid_SetDimension ( bHYPRE_StructGrid self,
                                      int32_t dim, sidl_BaseInterface *_ex)
      Method: SetDimension[]
int32_t
bHYPRE_StructGrid_SetExtents ( bHYPRE_StructGrid self,
                                   int32_t* ilower, int32_t* iupper,
                                   int32_t dim, sidl_BaseInterface *_ex)
      Method: SetExtents[]
SIDL_C_INLINE_DECL int32_t
bHYPRE_StructGrid_SetPeriodic ( bHYPRE_StructGrid self,
                                    int32_t* periodic, int32_t dim,
                                    sidl_BaseInterface *_ex)
      Method: SetPeriodic[]
SIDL_C_INLINE_DECL int32_t
bHYPRE_StructGrid_SetNumGhost ( bHYPRE_StructGrid self,
                                       int32_t* num_ghost, int32_t dim2,
                                       sidl_BaseInterface *_ex)
      Method: SetNumGhost[]
SIDL_C_INLINE_DECL int32_t
bHYPRE_StructGrid_Assemble ( bHYPRE_StructGrid self,
                                  sidl_BaseInterface *_ex)
      Method: Assemble[]
_{\mathbf{ex}}
       Cast method for interface and class type conversions
void*
bHYPRE_StructGrid__cast2 (void* obj, const char* type,
                              sidl_BaseInterface *_ex)
      String cast method for interface and class type conversions
SIDL_C_INLINE_DECL void
bHYPRE_StructGrid_exec ( bHYPRE_StructGrid self,
                             const char* methodName, sidl_rmi_Call inArgs,
                             sidl_rmi_Return outArgs,
                             sidl_BaseInterface *_ex)
      Select and execute a method by name
SIDL_C_INLINE_DECL char*
bHYPRE_StructGrid__getURL ( bHYPRE_StructGrid self,
                                 sidl_BaseInterface *_ex)
       Get the URL of the Implementation of this object (for RMI)
SIDL_C_INLINE_DECL_void
bHYPRE_StructGrid_raddRef ( bHYPRE_StructGrid self,
                                 sidl_BaseInterface *_ex)
       On a remote object, addrefs the remote instance
```

SIDL_C_INLINE_DECL_sidl_bool

 $\begin{tabular}{ll} \bf bHYPRE_StructGrid_isRemote (bHYPRE_StructGrid self, \\ sidl_BaseInterface *_ex) \end{tabular}$

TRUE if this object is remote, false if local

sidl_bool

bHYPRE_StructGrid__isLocal (bHYPRE_StructGrid self,

sidl_BaseInterface *_ex)

 $TRUE\ if\ this\ object\ is\ remote,\ false\ if\ local$

 $**_ex$

Cast method for interface and class type conversions

12.1.5 **_ex

12.1.1

struct bHYPRE_StructGrid_object

Symbol "bHYPRE.StructGrid" (version 1.0.0)

Define a structured grid class.

_ 12.1.2 _

bHYPRE_StructGrid bHYPRE_StructGrid__connect (const char *, sidl_BaseInterface *_ex)

RMI connector function for the class.(addrefs)

_ 12.1.3 __

SIDL_C_INLINE_DECL int32_t
bHYPRE_StructGrid_SetCommunicator (bHYPRE_StructGrid self,
bHYPRE_MPICommunicator mpi_comm, sidl_BaseInterface *_ex)

Set the MPI Communicator. DEPRECATED, use Create:

12.1.4

The Destroy function doesn't necessarily destroy anything. It is just another name for deleteRef. Thus it decrements the object's reference count. The Babel memory management system will destroy the object if the reference count goes to zero.

12.1.5

struct bHYPRE_StructGrid_object* bHYPRE_StructGrid_connectI const char * url sidl_bool ar struct sidl_BaseInterface_object **_ex

RMI connector function for the class. (no addref)

12.2

Struct Stencil

Names 12.2.1 struct bHYPRE_StructStencil__object Symbol "bHYPRE 242 _ex Constructor function for the class bHYPRE_StructStencil bHYPRE_StructStencil__createRemote (const char * url, sidl_BaseInterface *_ex) RMI constructor function for the class bHYPRE_StructStencil bHYPRE_StructStencil__wrapObj (void * data, sidl_BaseInterface *_ex) theprivatedatastructbHYPRE_StructStencil__data) passed in rather than running the constructor12.2.2 bHYPRE_StructStencil

bHYPRE_StructStencil__connect (const char *, sidl_BaseInterface *_ex)

RMI connector function for the class

bHYPRE_StructStencil

242

```
bHYPRE_StructStencil_Create ( int32_t ndim, int32_t size,
                                               sidl_BaseInterface *_ex)
                   Method: Create[]
12.2.3
             SIDL_C_INLINE_DECL void
             bHYPRE_StructStencil_Destroy ( bHYPRE_StructStencil self,
                                                sidl_BaseInterface *_ex)
                   The Destroy function doesn't necessarily destroy anything .....
                                                                                               242
             SIDL_C_INLINE_DECL int32_t
             bHYPRE_StructStencil_SetDimension ( bHYPRE_StructStencil self,
                                                      int32_t dim,
                                                      sidl_BaseInterface *_ex)
                   Method: SetDimension[]
             SIDL_C_INLINE_DECL int32_t
             bHYPRE_StructStencil_SetSize ( bHYPRE_StructStencil self, int32_t size,
                                               sidl_BaseInterface *_ex)
                   Method: SetSize[]
             SIDL_C_INLINE_DECL int32_t
             bHYPRE_StructStencil_SetElement ( bHYPRE_StructStencil self,
                                                    int32_t index, int32_t* offset,
                                                    int32_t dim, sidl_BaseInterface *_ex)
                   Method: SetElement//
             _{\mathbf{ex}}
                   Cast method for interface and class type conversions
             void*
             bHYPRE_StructStencil__cast2 (void* obj, const char* type,
                                              sidl_BaseInterface *_ex)
                   String cast method for interface and class type conversions
             SIDL_C_INLINE_DECL_void
             bHYPRE_StructStencil_exec ( bHYPRE_StructStencil self,
                                             const char* methodName,
                                             sidl_rmi_Call inArgs,
                                             sidl_rmi_Return outArgs,
                                             sidl_BaseInterface *_ex)
                   Select and execute a method by name
             SIDL_C_INLINE_DECL char*
             bHYPRE_StructStencil__getURL ( bHYPRE_StructStencil self,
                                                 sidl_BaseInterface *_ex)
                   Get the URL of the Implementation of this object (for RMI)
             SIDL_C_INLINE_DECL void
            bHYPRE_StructStencil__raddRef ( bHYPRE_StructStencil self,
                                                 sidl_BaseInterface *_ex)
                   On a remote object, addrefs the remote instance
             SIDL_C_INLINE_DECL_sidl_bool
             bHYPRE_StructStencil__isRemote ( bHYPRE_StructStencil self,
                                                  sidl_BaseInterface *_ex)
                   TRUE if this object is remote, false if local
            sidl\_bool
```

bHYPRE_StructStencil__isLocal (bHYPRE_StructStencil self, sidl_BaseInterface *_ex)

TRUE if this object is remote, false if local

**_ex

Cast method for interface and class type conversions

12.2.4 **_ex

_ 12.2.1 ___

struct bHYPRE_StructStencil_object

Symbol "bHYPRE.StructStencil" (version 1.0.0)

Define a structured stencil for a structured problem description. More than one implementation is not envisioned, thus the decision has been made to make this a class rather than an interface.

12.2.2

bHYPRE_StructStencil bHYPRE_StructStencil__connect (const char *, sidl_BaseInterface *_ex)

RMI connector function for the class.(addrefs)

12.2.3

SIDL_C_INLINE_DECL void bHYPRE_StructStencil_Destroy (bHYPRE_StructStencil self, sidl_BaseInterface *_ex)

The Destroy function doesn't necessarily destroy anything. It is just another name for deleteRef. Thus it decrements the object's reference count. The Babel memory management system will destroy the object if the reference count goes to zero.

1224

struct bHYPRE_StructStencil_object* bHYPRE_StructStencil_connectI const char * url sidl_bool ar struct sidl_BaseInterface_object ** $\bf ex$

__ 13

Semi-Structured Grid, etc.

Names		
13.1	Semi-Structured Graph	
		244
13.2	Semi-Structured Grid	
		249
13.3	Semi-Structured Stencil	
		254
	Semi-Structured Variable	

_ 13.1 __

${\bf Semi\text{-}Structured\ Graph}$

Names		
13.1.1	struct bHYPRE_SStructGraphobject Symbol "bHYPRE	246
	_ex	
	Constructor function for the class	
	bHYPRE_SStructGraph	
	bHYPRE_SStructGraphcreateRemote (const char * url, sidl_BaseInterface *_ex)	
	RMI constructor function for the class	
	bHYPRE_SStructGraph bHYPRE_SStructGraphwrapObj (void * data, sidl_BaseInterface *_ex) Wraps up the private data struct pointer (struct bHYPRE_SStructGraphdata) passed in rather than running the con- structor	
13.1.2	bHYPRE_SStructGraph bHYPRE_SStructGraphconnect (const char *, sidl_BaseInterface *_ex) RMI connector function for the class	247
	bHYPRE_SStructGraph bHYPRE_SStructGraph_Create (bHYPRE_MPICommunicator mpi_comm, bHYPRE_SStructGrid grid, sidl_BaseInterface *_ex)	
	Method: Create[]	
13.1.3	SIDL_C_INLINE_DECL_int32_t	

	bHYPRE_SStructGraph_SetCommGrid (bHYPRE_SStructGraph self, bHYPRE_MPICommunicator	
	mpi_comm,	
	bHYPRE_SStructGrid grid,	
	sidl_BaseInterface *_ex)	
	Set the grid and communicator	247
	SIDL_C_INLINE_DECL int32_t	
	bHYPRE_SStructGraph_SetStencil (bHYPRE_SStructGraph self,	
	int32_t part, int32_t var,	
	bHYPRE_SStructStencil stencil,	
	sidl_BaseInterface *_ex)	
	Set the stencil for a variable on a structured part of the grid	
13.1.4	$\mathrm{int}32$ _t	
	bHYPRE_SStructGraph_AddEntries (bHYPRE_SStructGraph self,	
	int32_t part, int32_t* index,	
	$int32_{-t} dim$, $int32_{-t} var$,	
	int32_t to_part, int32_t* to_index,	
	$int32_{-}t to_{-}var,$	
	sidl_BaseInterface *_ex)	
	Add a non-stencil graph entry at a particular index	247
	SIDL_C_INLINE_DECL_int32_t	
	bHYPRE_SStructGraph_SetObjectType (bHYPRE_SStructGraph self,	
	$int32_{-}t type,$	
	sidl_BaseInterface *_ex)	
	$Method:\ SetObjectType[]$	
13.1.5	SIDL_C_INLINE_DECL_int32_t	
	bHYPRE_SStructGraph_SetCommunicator (bHYPRE_SStructGraph self, bHYPRE_MPICommunicator	
	mpi_comm,	
	sidl_BaseInterface *_ex)	
	Set the MPI Communicator	247
13.1.6	SIDL_C_INLINE_DECL void bHYPRE_SStructGraph_Destroy (bHYPRE_SStructGraph self,	
	sidl_BaseInterface *_ex)	
	The Destroy function doesn't necessarily destroy anything	248
	SIDL_C_INLINE_DECL int32_t bHYPRE_SStructGraph_Initialize (bHYPRE_SStructGraph self,	
	sidl_BaseInterface *_ex)	
	Prepare an object for setting coefficient values, whether for the first time or subsequently	
13.1.7	SIDL_C_INLINE_DECL int32_t	
	$\mathbf{bHYPRE_SStructGraph_Assemble} \ (\ \ \mathbf{bHYPRE_SStructGraph} \ \mathbf{self},$	
	sidl_BaseInterface *_ex)	
	Finalize the construction of an object before using, either for the first time	
	or on subsequent uses	248
	_ex	
	Cast method for interface and class type conversions	
	void^*	
	1010	

 $\mathbf{bHYPRE_SStructGraph__cast2} \ (\ \mathrm{void}^* \ \mathrm{obj}, \ \ \mathrm{const} \ \mathrm{char}^* \ \mathrm{type},$ sidl_BaseInterface *_ex) String cast method for interface and class type conversions SIDL_C_INLINE_DECL_void bHYPRE_SStructGraph__exec (bHYPRE_SStructGraph self, const char* methodName, sidl_rmi_Call inArgs, sidl_rmi_Return outArgs, sidl_BaseInterface *_ex) Select and execute a method by name SIDL_C_INLINE_DECL char* bHYPRE_SStructGraph__getURL (bHYPRE_SStructGraph self, sidl_BaseInterface *_ex) Get the URL of the Implementation of this object (for RMI) SIDL_C_INLINE_DECL void bHYPRE_SStructGraph_raddRef (bHYPRE_SStructGraph self, sidl_BaseInterface *_ex) On a remote object, addrefs the remote instance SIDL_C_INLINE_DECL sidl_bool bHYPRE_SStructGraph__isRemote (bHYPRE_SStructGraph self, sidl_BaseInterface *_ex) TRUE if this object is remote, false if local $sidl_bool$ bHYPRE_SStructGraph__isLocal (bHYPRE_SStructGraph self, sidl_BaseInterface *_ex) TRUE if this object is remote, false if local **_ex Cast method for interface and class type conversions $**_ex$ 13.1.8 RMI connector function for the class 248

13.1.1

struct bHYPRE_SStructGraph_object

Symbol "bHYPRE.SStructGraph" (version 1.0.0)

The semi-structured grid graph class.

_ 13.1.2 _

bHYPRE_SStructGraph
bHYPRE_SStructGraph__connect (const char *, sidl_BaseInterface *_ex)

RMI connector function for the class.(addrefs)

__ 13.1.3 _____

SIDL_C_INLINE_DECL int32_t bHYPRE_SStructGraph_SetCommGrid (bHYPRE_SStructGraph self, bHYPRE_MPICommunicator mpi_comm, bHYPRE_SStructGrid grid, sidl_BaseInterface *_ex)

Set the grid and communicator. DEPRECATED, use Create:

13.1.4

int32_t bHYPRE_SStructGraph_AddEntries (bHYPRE_SStructGraph self, int32_t part, int32_t* index, int32_t dim, int32_t var, int32_t to_part, int32_t* to_index, int32_t to_var, sidl_BaseInterface *_ex)

Add a non-stencil graph entry at a particular index. This graph entry is appended to the existing graph entries, and is referenced as such.

NOTE: Users are required to set graph entries on all processes that own the associated variables. This means that some data will be multiply defined.

_ 13.1.5 _

SIDL_C_INLINE_DECL int32_t
bHYPRE_SStructGraph_SetCommunicator (bHYPRE_SStructGraph self,
bHYPRE_MPICommunicator mpi_comm, sidl_BaseInterface *_ex)

Set the MPI Communicator. DEPRECATED, Use Create()

___ 13.1.6 ____

SIDL_C_INLINE_DECL void bHYPRE_SStructGraph_Destroy (bHYPRE_SStructGraph self, sidl_BaseInterface *_ex)

The Destroy function doesn't necessarily destroy anything. It is just another name for deleteRef. Thus it decrements the object's reference count. The Babel memory management system will destroy the object if the reference count goes to zero.

13.1.7

SIDL_C_INLINE_DECL int32_t bHYPRE_SStructGraph_Assemble (bHYPRE_SStructGraph self, sidl_BaseInterface *_ex)

Finalize the construction of an object before using, either for the first time or on subsequent uses. Initialize and Assemble always appear in a matched set, with Initialize preceding Assemble. Values can only be set in between a call to Initialize and Assemble.

13.1.8

struct bHYPRE_SStructGraph__object* bHYPRE_SStructGraph__connectI const char * url sidl_bool ar struct sidl_BaseInterface__object **_ex

13 2

Semi-Structured Grid

Names		
13.2.1	struct bHYPRE_SStructGridobject Symbol "bHYPRE	251
	_ex	
	Constructor function for the class	
	bHYPRE_SStructGrid bHYPRE_SStructGridcreateRemote (const char * url,	
	RMI constructor function for the class	
	bHYPRE_SStructGrid bHYPRE_SStructGridwrapObj (void * data, sidl_BaseInterface *_ex) Wraps up the private data struct pointer (struct bHYPRE_SStructGrid_data) passed in rather than running the constructor	
13.2.2	bHYPRE_SStructGrid bHYPRE_SStructGridconnect (const char *, sidl_BaseInterface *_ex) RMI connector function for the class	252
	bHYPRE_SStructGrid bHYPRE_MPICommunicator mpi_comm,	
	SIDL_C_INLINE_DECL_int32_t bHYPRE_SStructGrid_SetNumDimParts (bHYPRE_SStructGrid self, int32_t ndim, int32_t nparts, sidl_BaseInterface *_ex)	
	$Method:\ SetNumDimParts[]$	
	SIDL_C_INLINE_DECL int32_t bHYPRE_SStructGrid_SetCommunicator (bHYPRE_SStructGrid self, bHYPRE_MPICommunicator mpi_comm, sidl_BaseInterface *_ex)	
	Method: SetCommunicator//	
13.2.3	SIDL_C_INLINE_DECL_void bHYPRE_SStructGrid_Destroy (bHYPRE_SStructGrid self, sidl_BaseInterface *_ex)	
	The Destroy function doesn't necessarily destroy anything	252
	$\mathrm{int}32_\mathrm{t}$	

	bHYPRE_SStructGrid_SetExtents (bHYPRE_SStructGrid self, int32_t part, int32_t* ilower, int32_t* iupper, int32_t dim,	
	sidl_BaseInterface *_ex) Set the extents for a box on a structured part of the grid	
13.2.4	SIDL_C_INLINE_DECL_int32_t bHYPRE_SStructGrid_SetVariable (bHYPRE_SStructGrid self,	
	int32_t part, int32_t var, int32_t nvars, enum bHYPRE_SStructVariableenum vartype, sidl_BaseInterface *_ex)	
	Describe the variables that live on a structured part of the grid	252
13.2.5	SIDL_C_INLINE_DECL int32_t bHYPRE_SStructGrid_AddVariable (bHYPRE_SStructGrid self,	
	Describe additional variables that live at a particular index	252
13.2.6	${ m int}32$ _t	
10.2.0	bHYPRE_SStructGrid_SetNeighborBox (bHYPRE_SStructGrid self, int32_t part, int32_t* ilower, int32_t* iupper, int32_t* nbor_part, int32_t* nbor_ilower, int32_t* nbor_iupper, int32_t* nbor_iupper, int32_t* nbor_iupper, int32_t* index_map, int32_t dim,	
	sidl_BaseInterface *_ex) Describe how regions just outside of a part relate to other parts	253
13.2.7	SIDL_C_INLINE_DECL int32_t bHYPRE_SStructGrid_AddUnstructuredPart (bHYPRE_SStructGrid self, int32_t ilower, int32_t iupper, sidl_BaseInterface *_ex)	200
	Add an unstructured part to the grid	253
	SIDL_C_INLINE_DECL int32_t bHYPRE_SStructGrid_SetPeriodic (bHYPRE_SStructGrid self,	
	int32_t dim, sidl_BaseInterface *_ex)	
	(Optional) Set periodic for a particular part	
	SIDL_C_INLINE_DECL int32_t bHYPRE_SStructGrid_SetNumGhost (bHYPRE_SStructGrid self,	
	Setting ghost in the sgrids	
	SIDL_C_INLINE_DECL_int32_t bHYPRE_SStructGrid_Assemble (bHYPRE_SStructGrid self,	
	$Method: \ Assemble[]$	
	_ex	

Cast method for interface and class type conversions

void*

bHYPRE_SStructGrid__cast2 (void* obj, const char* type, sidl_BaseInterface *_ex)

String cast method for interface and class type conversions

SIDL_C_INLINE_DECL void

bHYPRE_SStructGrid_exec (bHYPRE_SStructGrid self,

const char* methodName,

sidl_rmi_Call inArgs, sidl_rmi_Return outArgs,

sidl_BaseInterface *_ex)

Select and execute a method by name

SIDL_C_INLINE_DECL char*

 ${\bf bHYPRE_SStructGrid__getURL} \ (\ \ {\bf bHYPRE_SStructGrid} \ {\bf self},$

sidl_BaseInterface *_ex)

Get the URL of the Implementation of this object (for RMI)

SIDL_C_INLINE_DECL_void

bHYPRE_SStructGrid_raddRef (bHYPRE_SStructGrid self,

sidl_BaseInterface *_ex)

On a remote object, addrefs the remote instance

SIDL_C_INLINE_DECL sidl_bool

bHYPRE_SStructGrid__isRemote (bHYPRE_SStructGrid self,

sidl_BaseInterface *_ex)

TRUE if this object is remote, false if local

sidl_bool

 $\mathbf{bHYPRE_SStructGrid__isLocal} \ (\ \ \mathbf{bHYPRE_SStructGrid} \ \mathbf{self},$

sidl_BaseInterface *_ex)

TRUE if this object is remote, false if local

 $**_ex$

Cast method for interface and class type conversions

13.2.8 **_ex

13.2.1

struct bHYPRE_SStructGrid_object

Symbol "bHYPRE.SStructGrid" (version 1.0.0)

The semi-structured grid class.

13.2.2

bHYPRE_SStructGrid bHYPRE_SStructGrid_connect (const char *, sidl_BaseInterface *_ex)

RMI connector function for the class.(addrefs)

_ 13.2.3 _

SIDL_C_INLINE_DECL void bHYPRE_SStructGrid_Destroy (bHYPRE_SStructGrid self, sidl_BaseInterface *_ex)

The Destroy function doesn't necessarily destroy anything. It is just another name for deleteRef. Thus it decrements the object's reference count. The Babel memory management system will destroy the object if the reference count goes to zero.

_ 13.2.4 _

SIDL_C_INLINE_DECL int32_t bHYPRE_SStructGrid_SetVariable (bHYPRE_SStructGrid self, int32_t part, int32_t var, int32_t nvars, enum bHYPRE_SStructVariable_enum vartype, sidl_BaseInterface *_ex)

Describe the variables that live on a structured part of the grid. Input: part number, variable number, total number of variables on that part (needed for memory allocation), variable type.

$_{-}$ 13.2.5 $_{-}$

SIDL_C_INLINE_DECL int32_t bHYPRE_SStructGrid_AddVariable (bHYPRE_SStructGrid self, int32_t part, int32_t* index, int32_t dim, int32_t var, enum bHYPRE_SStructVariable_enum vartype, sidl_BaseInterface *_ex)

Describe additional variables that live at a particular index. These variables are appended to the array of variables set in SetVariables, and are referenced as such.

13.2.6

 $int32_t$

bHYPRE_SStructGrid_SetNeighborBox (bHYPRE_SStructGrid self, int32_t part, int32_t* ilower, int32_t* iupper, int32_t nbor_part, int32_t* nbor_ilower, int32_t* nbor_iupper, int32_t* index_map, int32_t dim, sidl_BaseInterface *_ex)

Describe how regions just outside of a part relate to other parts. This is done a box at a time.

The indexes ilower and iupper map directly to the indexes nbor_ilower and nbor_iupper. Although, it is required that indexes increase from ilower to iupper, indexes may increase and/or decrease from nbor_ilower to nbor_iupper.

The index_map describes the mapping of indexes 0, 1, and 2 on part part to the corresponding indexes on part nbor_part. For example, triple (1, 2, 0) means that indexes 0, 1, and 2 on part part map to indexes 1, 2, and 0 on part nbor_part, respectively.

NOTE: All parts related to each other via this routine must have an identical list of variables and variable types. For example, if part 0 has only two variables on it, a cell centered variable and a node centered variable, and we declare part 1 to be a neighbor of part 0, then part 1 must also have only two variables on it, and they must be of type cell and node.

13.2.7

SIDL_C_INLINE_DECL int32_t bHYPRE_SStructGrid_AddUnstructuredPart (bHYPRE_SStructGrid self, int32_t ilower, int32_t iloper, sidl_BaseInterface *_ex)

Add an unstructured part to the grid. The variables in the unstructured part of the grid are referenced by a global rank between 0 and the total number of unstructured variables minus one. Each process owns some unique consecutive range of variables, defined by ilower and iupper.

NOTE: This is just a placeholder. This part of the interface is not finished.

_ 13.2.8 _

struct bHYPRE_SStructGrid_object* bHYPRE_SStructGrid_connectI const char * url sidl_bool ar struct sidl_BaseInterface_object **_ex

_ 13.3 _

Semi-Structured Stencil

Names		
13.3.1	struct bHYPRE_SStructStencil_object Symbol "bHYPRE	255
	_ex	
	Constructor function for the class	
	bHYPRE_SStructStencil	
	bHYPRE_SStructStencilcreateRemote (const char * url, sidl_BaseInterface *_ex)	
	RMI constructor function for the class	
	bHYPRE_SStructStencil bHYPRE_SStructStencilwrapObj (void * data, sidl_BaseInterface *_ex) Wraps up the private data struct pointer (struct bHYPRE_SStructStencildata) passed in rather than running the con- structor	
13.3.2	bHYPRE_SStructStencil bHYPRE_SStructStencilconnect (const char *, sidl_BaseInterface *_ex) RMI connector function for the class	255
	bHYPRE_SStructStencil bHYPRE_SStructStencil_Create (int32_t ndim, int32_t size, sidl_BaseInterface *_ex)	
	Method: Create[]	
13.3.3	SIDL_C_INLINE_DECL_void bHYPRE_SStructStencil_Destroy (bHYPRE_SStructStencil self, sidl_BaseInterface *_ex)	
	The Destroy function doesn't necessarily destroy anything	256
13.3.4	SIDL_C_INLINE_DECL int32_t	
	bHYPRE_SStructStencil_SetNumDimSize (bHYPRE_SStructStencil self, int32_t ndim, int32_t size, sidl_BaseInterface *_ex)	
	Set the number of spatial dimensions and stencil entries	256
	SIDL_C_INLINE_DECL int32_t bHYPRE_SStructStencil_SetEntry (bHYPRE_SStructStencil self,	
	Set a stencil entry	
	_ex Cast method for interface and class type conversions	
	void^*	
	bHYPRE_SStructStencilcast2 (void* obj, const char* type, sidl_BaseInterface *_ex)	
	String cast method for interface and class type conversions	
	SIDL_C_INLINE_DECL void	

bHYPRE_SStructStencil__exec (bHYPRE_SStructStencil self, const char* methodName, sidl_rmi_Call inArgs, sidl_rmi_Return outArgs, sidl_BaseInterface *_ex)

Select and execute a method by name

SIDL_C_INLINE_DECL char*

 $\label{eq:bhypre_structStencil} \begin{tabular}{ll} \mathbf{bhypre_SStructStencil\ self}, \\ \mathbf{sidl_BaseInterface\ *_ex}) \end{tabular}$

Get the URL of the Implementation of this object (for RMI)

SIDL_C_INLINE_DECL_void

bHYPRE_SStructStencil__raddRef (bHYPRE_SStructStencil self, sidl_BaseInterface *_ex)

On a remote object, addrefs the remote instance

SIDL_C_INLINE_DECL sidl_bool

bHYPRE_SStructStencil__isRemote (bHYPRE_SStructStencil self, sidl_BaseInterface *_ex)

TRUE if this object is remote, false if local

 $sidl_bool$

bHYPRE_SStructStencil__isLocal (bHYPRE_SStructStencil self,

sidl_BaseInterface *_ex)

TRUE if this object is remote, false if local

**_ex

 $Cast\ method\ for\ interface\ and\ class\ type\ conversions$

13.3.5 **_ex

RMI connector function for the class

13.3.1

struct bHYPRE_SStructStencil_object

Symbol "bHYPRE.SStructStencil" (version 1.0.0)

The semi-structured grid stencil class.

 $_{-}$ 13.3.2 $_{-}$

bHYPRE_SStructStencil

bHYPRE_SStructStencil__connect (const char *, sidl_BaseInterface *_ex)

256

RMI connector function for the class.(addrefs)

13.3.3

SIDL_C_INLINE_DECL void bHYPRE_SStructStencil_Destroy (bHYPRE_SStructStencil self, sidl_BaseInterface *_ex)

The Destroy function doesn't necessarily destroy anything. It is just another name for deleteRef. Thus it decrements the object's reference count. The Babel memory management system will destroy the object if the reference count goes to zero.

13.3.4

SIDL_C_INLINE_DECL int32_t bHYPRE_SStructStencil_SetNumDimSize (bHYPRE_SStructStencil self, int32_t ndim, int32_t size, sidl_BaseInterface *_ex)

Set the number of spatial dimensions and stencil entries. DEPRECATED, use Create:

_ 13.3.5 ___

Class Graph