

hypre Reference Manual

— Version 1.11.1b —

Contents

1	Matrix and Vector Views (Conceptual Interfaces) —	4
1.1	IJ Matrix View —	4
1.2	IJ Vector View —	9
1.3	Struct Matrix View —	12
1.4	Struct Vector View —	14
1.5	SemiStructured Matrix View —	15
1.6	SemiStructured Vector View —	20
2	Operator Interface —	26
3	Vector Interface —	29
4	Matrices and Vectors —	31
4.1	IJParCSR Matrix —	31
4.2	IJParCSR Vector —	40
4.3	Struct Matrix —	45
4.4	Struct Vector —	50
4.5	SemiStructured Matrix —	53
4.6	SemiStructured Vector —	61
4.7	SemiStructured ParCSR Matrix —	68
4.8	SemiStructured ParCSR Vector —	77
5	Solver Interface —	85
5.7	Identity Solver (does nothing) —	87
5.8	Hybrid Solver —	93
6	ParCSR Matrix Solvers — <i>Linear solvers for sparse matrix systems</i>	98
6.1	ParCSRDiagScale Solver —	98
6.2	ParCSR BoomerAMG Solver —	104
6.3	ParCSR Euclid Solver —	111
6.4	ParCSR Schwarz Solver —	115
6.5	ParCSR ParaSails Solver —	120
6.6	ParCSR Pilut Solver —	125
7	Structured Matrix Solvers — <i>Linear solvers for struct matrix systems</i>	130
7.1	StructDiagScale Solver —	130
7.2	Struct Jacobi Solver —	135
7.3	Struct PFMG Solver —	140
7.4	Struct SMG Solver —	145
8	SemiStructured Matrix Solvers — <i>Linear solvers for semi-struct matrix systems</i>	151
8.1	SemiStruct DiagScale Solver —	151
8.2	Struct Split Solver —	156

9	PreconditionedSolver Interface —	162
10	Preconditioned Solvers —	164
10.1	PCG Preconditioned Solver —	164
10.2	GMRES Preconditioned Solver —	169
10.3	BiCGSTAB Preconditioned Solver —	174
10.4	CGNR Preconditioned Solver —	179
11	Other —	185
11.1	MPI Communicator —	185
12	Struct Grid, etc. —	187
12.1	Struct Grid —	187
12.2	Struct Stencil —	189
13	Semi-Structured Grid, etc. —	191
13.1	Semi-Structured Graph —	191
13.2	Semi-Structured Grid —	194
13.3	Semi-Structured Stencil —	198
	Class Graph	200

Matrix and Vector Views (Conceptual Interfaces)

Names

1.1	IJ Matrix View	4
1.2	IJ Vector View	9
1.3	Struct Matrix View	12
1.4	Struct Vector View	14
1.5	SemiStructured Matrix View	15
1.6	SemiStructured Vector View	20

IJ Matrix View

Names

1.1.1	struct bHYPRE_IJMatrixView__object <i>Symbol "bHYPRE"</i>	6
	extern C bHYPRE_IJMatrixView bHYPRE_IJMatrixView__connect (const char *, sidl_BaseInterface *_ex) <i>RMI connector function for the class</i>	
1.1.2	int32_t bHYPRE_IJMatrixView_SetLocalRange (bHYPRE_IJMatrixView self, int32_t ilower, int32_t iupper, int32_t jlower, int32_t jupper) <i>Set the local range for a matrix object</i>	6
1.1.3	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) <i>Sets values for nrows of the matrix</i>	7
1.1.4	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) <i>Adds to values for nrows of the matrix</i>	7
	int32_t bHYPRE_IJMatrixView_GetLocalRange (bHYPRE_IJMatrixView self, int32_t* ilower, int32_t* iupper, int32_t* jlower, int32_t* jupper) <i>Gets range of rows owned by this processor and range of column partitioning for this processor</i>	
	int32_t bHYPRE_IJMatrixView_GetRowCounts (bHYPRE_IJMatrixView self, int32_t nrows, int32_t* rows, int32_t* ncols) <i>Gets number of nonzeros elements for nrows rows specified in rows and returns them in ncols, which needs to be allocated by the user</i>	
1.1.5	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) <i>Gets values for nrows rows or partial rows of the matrix</i>	7
1.1.6	int32_t bHYPRE_IJMatrixView_SetRowSizes (bHYPRE_IJMatrixView self, int32_t* sizes, int32_t nrows) <i>(Optional) Set the max number of nonzeros to expect in each row</i>	8
1.1.7	int32_t bHYPRE_IJMatrixView_Print (bHYPRE_IJMatrixView self, const char* filename) <i>Print the matrix to file</i>	8
1.1.8	int32_t bHYPRE_IJMatrixView_Read (bHYPRE_IJMatrixView self, const char* filename, bHYPRE_MPICommunicator comm) <i>Read the matrix from file</i>	8
	struct bHYPRE_IJMatrixView__object* bHYPRE_IJMatrixView__cast void* obj <i>Cast method for interface and class type conversions</i>	
	void* bHYPRE_IJMatrixView__cast2 (void* obj, const char* type) <i>String cast method for interface and class type conversions</i>	
	void bHYPRE_IJMatrixView__exec (bHYPRE_IJMatrixView self, const char* methodName, sidl_io_Deserializer inArgs, sidl_io_Serializer outArgs) <i>Select and execute a method by name</i>	
	char*	

bHYPRE_IJMatrixView__getURL (bHYPRE_IJMatrixView self)
Get the URL of the Implementation of this object (for RMI)

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

```
int32_t
bHYPRE_IJMatrixView_SetLocalRange ( bHYPRE_IJMatrixView self,
int32_t ilower, int32_t iupper, int32_t jlower, int32_t jupper)
```

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.

1.1.3

```
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)
```

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 function 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.

1.1.4

```
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)
```

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.

1.1.5

```
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)
```

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

1.1.6

```
int32_t
bHYPRE_IJMatrixView_SetRowSizes ( bHYPRE_IJMatrixView self, int32_t*
sizes, int32_t nrows)
```

(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.

1.1.7

```
int32_t
bHYPRE_IJMatrixView_Print ( bHYPRE_IJMatrixView self, const char*
filename)
```

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

1.1.8

```
int32_t
bHYPRE_IJMatrixView_Read ( bHYPRE_IJMatrixView self, const char*
filename, bHYPRE_MPICommunicator comm)
```

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

IJ Vector View

1.2.1	<pre> struct bHYPRE_IJVectorView__object Symbol "bHYPRE" extern C bHYPRE_IJVectorView bHYPRE_IJVectorView__connect (const char *, sidl_BaseInterface *_ex) RMI connector function for the class </pre>	10
1.2.2	<pre> int32_t bHYPRE_IJVectorView_SetLocalRange (bHYPRE_IJVectorView self, int32_t jlower, int32_t jupper) Set the local range for a vector object </pre>	10
1.2.3	<pre> int32_t bHYPRE_IJVectorView_SetValues (bHYPRE_IJVectorView self, int32_t nvalues, int32_t* indices, double* values) Sets values in vector </pre>	10
1.2.4	<pre> int32_t bHYPRE_IJVectorView_AddToValues (bHYPRE_IJVectorView self, int32_t nvalues, int32_t* indices, double* values) Adds to values in vector </pre>	11
	<pre> int32_t bHYPRE_IJVectorView_GetLocalRange (bHYPRE_IJVectorView self, int32_t* jlower, int32_t* jupper) Returns range of the part of the vector owned by this processor </pre>	
1.2.5	<pre> int32_t bHYPRE_IJVectorView_GetValues (bHYPRE_IJVectorView self, int32_t nvalues, int32_t* indices, double* values) Gets values in vector </pre>	11
1.2.6	<pre> int32_t bHYPRE_IJVectorView_Print (bHYPRE_IJVectorView self, const char* filename) Print the vector to file </pre>	11
1.2.7	<pre> int32_t bHYPRE_IJVectorView_Read (bHYPRE_IJVectorView self, const char* filename, bHYPRE_MPICommunicator comm) Read the vector from file </pre>	12
	<pre> struct bHYPRE_IJVectorView__object* bHYPRE_IJVectorView__cast void* obj Cast method for interface and class type conversions void* </pre>	

```

bHYPRE_IJVectorView__cast2 ( void* obj, const char* type)
    String cast method for interface and class type conversions

void
bHYPRE_IJVectorView__exec ( bHYPRE_IJVectorView self,
                             const char* methodName,
                             sidl_io_Deserializer inArgs,
                             sidl_io_Serializer outArgs)
    Select and execute a method by name

char*
bHYPRE_IJVectorView__getURL ( bHYPRE_IJVectorView self)
    Get the URL of the Implementation of this object (for RMI)

```

1.2.1

```

struct bHYPRE_IJVectorView__object

```

Symbol "bHYPRE.IJVectorView" (version 1.0.0)

1.2.2

```

int32_t
bHYPRE_IJVectorView_SetLocalRange ( bHYPRE_IJVectorView self,
int32_t jlower, int32_t jupper)

```

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.

1.2.3

```

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

```

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.

1.2.4

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

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

Not collective.

1.2.5

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

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

Not collective.

1.2.6

```
int32_t  
bHYPRE_IJVectorView_Print ( bHYPRE_IJVectorView self, const char*  
filename)
```

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

1.2.7

```
int32_t
bHYPRE_IJVectorView_Read ( bHYPRE_IJVectorView self, const char*
filename, bHYPRE_MPICommunicator comm)
```

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

1.3**Struct Matrix View****Names**

```
1.3.1      struct bHYPRE_StructMatrixView__object
              Symbol "bHYPRE" ..... 14

extern C bHYPRE_StructMatrixView
bHYPRE_StructMatrixView__connect (const char *,
                                sidl_BaseInterface *_ex)
              RMI connector function for the class

int32_t
bHYPRE_StructMatrixView_SetGrid ( bHYPRE_StructMatrixView self,
                                bHYPRE_StructGrid grid)
              Method: SetGrid[]

int32_t
bHYPRE_StructMatrixView_SetStencil ( bHYPRE_StructMatrixView self,
                                bHYPRE_StructStencil stencil)
              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)
              Method: SetValues[]

int32_t
```

bHYPRE_StructMatrixView_SetBoxValues (bHYPRE_StructMatrixView self, int32_t* ilower, int32_t* iupper, int32_t dim, int32_t num_stencil_indices, int32_t* stencil_indices, double* values, int32_t nvalues)

Method: SetBoxValues[]

int32_t

bHYPRE_StructMatrixView_SetNumGhost (bHYPRE_StructMatrixView self, int32_t* num_ghost, int32_t dim2)

Method: SetNumGhost[]

int32_t

bHYPRE_StructMatrixView_SetSymmetric (bHYPRE_StructMatrixView self, int32_t symmetric)

Method: SetSymmetric[]

int32_t

bHYPRE_StructMatrixView_SetConstantEntries (bHYPRE_StructMatrixView self, int32_t num_stencil_constant_points, int32_t* stencil_constant_points)

Method: SetConstantEntries[]

int32_t

bHYPRE_StructMatrixView_SetConstantValues (bHYPRE_StructMatrixView self, int32_t num_stencil_indices, int32_t* stencil_indices, double* values)

Method: SetConstantValues[]

obj

Cast method for interface and class type conversions

void*

bHYPRE_StructMatrixView__cast2 (void* obj, const char* type)

String cast method for interface and class type conversions

void

bHYPRE_StructMatrixView__exec (bHYPRE_StructMatrixView self, const char* methodName, sidl_io_Deserializer inArgs, sidl_io_Serializer outArgs)

Select and execute a method by name

char*

bHYPRE_StructMatrixView__getURL (bHYPRE_StructMatrixView self)

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

1.3.1

```
struct bHYPRE_StructMatrixView__object
```

Symbol "bHYPRE.StructMatrixView" (version 1.0.0)

1.4

Struct Vector View

Names

```
1.4.1      struct bHYPRE_StructVectorView__object
              Symbol "bHYPRE" ..... 15

extern C bHYPRE_StructVectorView
bHYPRE_StructVectorView__connect (const char *, sidl_BaseInterface *_ex)
    RMI connector function for the class

int32_t
bHYPRE_StructVectorView_SetGrid ( bHYPRE_StructVectorView self,
                                   bHYPRE_StructGrid grid)
    Method: SetGrid[]

int32_t
bHYPRE_StructVectorView_SetNumGhost ( bHYPRE_StructVectorView
                                       self, int32_t* num_ghost,
                                       int32_t dim2)
    Method: SetNumGhost[]

int32_t
bHYPRE_StructVectorView_SetValue ( bHYPRE_StructVectorView self,
                                   int32_t* grid_index, int32_t dim,
                                   double value)
    Method: SetValue[]

int32_t
bHYPRE_StructVectorView_SetBoxValues ( bHYPRE_StructVectorView
                                       self, int32_t* ilower,
                                       int32_t* iupper, int32_t dim,
                                       double* values,
                                       int32_t nvalues)
    Method: SetBoxValues[]

obj
    Cast method for interface and class type conversions

void*
bHYPRE_StructVectorView__cast2 ( void* obj, const char* type)
    String cast method for interface and class type conversions

void
```

```
bHYPRE_StructVectorView__exec ( bHYPRE_StructVectorView self,
                                const char* methodName,
                                sidl_io_Deserializer inArgs,
                                sidl_io_Serializer outArgs)
```

Select and execute a method by name

char*

```
bHYPRE_StructVectorView__getURL ( bHYPRE_StructVectorView self)
Get the URL of the Implementation of this object (for RMI)
```

1.4.1

```
struct bHYPRE_StructVectorView__object
```

Symbol "bHYPRE.StructVectorView" (version 1.0.0)

1.5

SemiStructured Matrix View

Names

1.5.1	<pre>struct bHYPRE_SStructMatrixView__object <i>Symbol "bHYPRE"</i></pre>	17
	<pre>extern C bHYPRE_SStructMatrixView bHYPRE_SStructMatrixView__connect (const char *, sidl_BaseInterface *_ex) <i>RMI connector function for the class</i></pre>	
1.5.2	<pre>int32_t bHYPRE_SStructMatrixView_SetGraph (bHYPRE_SStructMatrixView self, bHYPRE_SStructGraph graph) <i>Set the matrix graph</i></pre>	17
1.5.3	<pre>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) <i>Set matrix coefficients index by index</i></pre>	18
1.5.4	<pre>int32_t</pre>	

		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) <i>Set matrix coefficients a box at a time</i>	18
1.5.5	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) <i>Add to matrix coefficients index by index</i>	19
1.5.6	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) <i>Add to matrix coefficients a box at a time</i>	19
1.5.7	int32_t	bHYPRE_SStructMatrixView_SetSymmetric (bHYPRE_SStructMatrixView self, int32_t part, int32_t var, int32_t to_var, int32_t symmetric) <i>Define symmetry properties for the stencil entries in the matrix</i>	20
	int32_t	bHYPRE_SStructMatrixView_SetNSSymmetric (bHYPRE_SStructMatrixView self, int32_t symmetric) <i>Define symmetry properties for all non-stencil matrix entries</i>	
	int32_t	bHYPRE_SStructMatrixView_SetComplex (bHYPRE_SStructMatrixView self) <i>Set the matrix to be complex</i>	
1.5.8	int32_t		

1.5.1

Symbol "bHYPRE.SStructMatrixView" (version 1.0.0)

Set the matrix graph. DEPRECATED Use Create

1.5.3

```
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)
```

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.

1.5.4

```
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)
```

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.

1.5.5

```
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)
```

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.

1.5.6

```
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)
```

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.

1.5.7

```
int32_t
bHYPRE_SStructMatrixView_SetSymmetric ( bHYPRE_SStructMatrixView
self, int32_t part, int32_t var, int32_t to_var, int32_t symmetric)
```

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.

1.5.8

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

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

1.6**SemiStructured Vector View****Names**

- | | | |
|-------|--|----|
| 1.6.1 | <pre>struct bHYPRE_SStructVectorView__object <i>Symbol "bHYPRE"</i> extern C bHYPRE_SStructVectorView bHYPRE_SStructVectorView__connect (const char *, sidl_BaseInterface *_ex) <i>RMI connector function for the class</i> int32_t bHYPRE_SStructVectorView_SetGrid (bHYPRE_SStructVectorView self, bHYPRE_SStructGrid grid) <i>Set the vector grid</i></pre> | 22 |
| 1.6.2 | <pre>int32_t</pre> | |

	bHYPRE_SStructVectorView_SetValues (bHYPRE_SStructVectorView self, int32_t part, int32_t* index, int32_t dim, int32_t var, double value)	
	<i>Set vector coefficients index by index</i>	22
1.6.3	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)	
	<i>Set vector coefficients a box at a time</i>	23
1.6.4	int32_t bHYPRE_SStructVectorView_AddToValues (bHYPRE_SStructVectorView self, int32_t part, int32_t* index, int32_t dim, int32_t var, double value)	
	<i>Set vector coefficients index by index</i>	23
1.6.5	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)	
	<i>Set vector coefficients a box at a time</i>	24
	int32_t bHYPRE_SStructVectorView_Gather (bHYPRE_SStructVectorView self) <i>Gather vector data before calling GetValues</i>	
1.6.6	int32_t bHYPRE_SStructVectorView_GetValues (bHYPRE_SStructVectorView self, int32_t part, int32_t* index, int32_t dim, int32_t var, double* value)	
	<i>Get vector coefficients index by index</i>	24
1.6.7	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)	
	<i>Get vector coefficients a box at a time</i>	24
	int32_t	

	bHYPRE_SStructVectorView_SetComplex (bHYPRE_SStructVectorView self)	
	<i>Set the vector to be complex</i>	
1.6.8	int32_t bHYPRE_SStructVectorView_Print (bHYPRE_SStructVectorView self, const char* filename, int32_t all)	
	<i>Print the vector to file</i>	25
	obj <i>Cast method for interface and class type conversions</i>	
	void* bHYPRE_SStructVectorView__cast2 (void* obj, const char* type)	
	<i>String cast method for interface and class type conversions</i>	
	void bHYPRE_SStructVectorView__exec (bHYPRE_SStructVectorView self, const char* methodName, sidl_io_Deserializer inArgs, sidl_io_Serializer outArgs)	
	<i>Select and execute a method by name</i>	
	char* bHYPRE_SStructVectorView__getURL (bHYPRE_SStructVectorView self)	
	<i>Get the URL of the Implementation of this object (for RMI)</i>	

1.6.1

```
struct bHYPRE_SStructVectorView__object
```

Symbol "bHYPRE.SStructVectorView" (version 1.0.0)

1.6.2

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

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.

1.6.3

```
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)
```

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.

1.6.4

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

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.

1.6.5

```
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)
```

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.

1.6.6

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

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.

1.6.7

```
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)
```


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.

1.6.8

```
int32_t  
bHYPRE_SStructVectorView_Print ( bHYPRE_SStructVectorView self, const  
char* filename, int32_t all)
```

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

Operator Interface

```

2.1      struct bHYPRE_Operator__object
          Symbol "bHYPRE" ..... 28

extern C bHYPRE_Operator
bHYPRE_Operator__connect (const char *, sidl_BaseInterface *_ex)
    RMI connector function for the class

2.2      int32_t
bHYPRE_Operator_SetCommunicator ( bHYPRE_Operator self,
                                   bHYPRE_MPICommunicator
                                   mpi_comm)
    Set the MPI Communicator ..... 28

      int32_t
bHYPRE_Operator_SetIntParameter ( bHYPRE_Operator self,
                                   const char* name, int32_t value)
    Set the int parameter associated with name

      int32_t
bHYPRE_Operator_SetDoubleParameter ( bHYPRE_Operator self,
                                       const char* name, double value)
    Set the double parameter associated with name

      int32_t
bHYPRE_Operator_SetStringParameter ( bHYPRE_Operator self,
                                       const char* name,
                                       const char* value)
    Set the string parameter associated with name

      int32_t
bHYPRE_Operator_SetIntArray1Parameter ( bHYPRE_Operator self,
                                           const char* name,
                                           int32_t* value,
                                           int32_t nvalues)
    Set the int 1-D array parameter associated with name

      int32_t
bHYPRE_Operator_SetIntArray2Parameter ( bHYPRE_Operator self,
                                           const char* name,
                                           struct sidl_int__array* value)
    Set the int 2-D array parameter associated with name

      int32_t
bHYPRE_Operator_SetDoubleArray1Parameter ( bHYPRE_Operator self,
                                              const char* name,
                                              double* value,
                                              int32_t nvalues)
    Set the double 1-D array parameter associated with name

      int32_t

```

```

bHYPRE_Operator_SetDoubleArray2Parameter ( bHYPRE_Operator self,
                                             const char* name,
                                             struct sidl_double_array*
                                             value)
    Set the double 2-D array parameter associated with name

int32_t
bHYPRE_Operator_GetIntValue ( bHYPRE_Operator self,
                               const char* name, int32_t* value)
    Set the int parameter associated with name

int32_t
bHYPRE_Operator_GetDoubleValue ( bHYPRE_Operator self,
                                   const char* name, double* value)
    Get the double parameter associated with name

int32_t
bHYPRE_Operator_Setup ( bHYPRE_Operator self, bHYPRE_Vector b,
                        bHYPRE_Vector x)
    (Optional) Do any preprocessing that may be necessary in order to execute
    Apply

int32_t
bHYPRE_Operator_Apply ( bHYPRE_Operator self, bHYPRE_Vector b,
                        bHYPRE_Vector* x)
    Apply the operator to b, returning x

int32_t
bHYPRE_Operator_ApplyAdjoint ( bHYPRE_Operator self,
                                bHYPRE_Vector b,
                                bHYPRE_Vector* x)
    Apply the adjoint of the operator to b, returning x

struct bHYPRE_Operator__object* bHYPRE_Operator__cast void* obj
    Cast method for interface and class type conversions

void*
bHYPRE_Operator__cast2 ( void* obj, const char* type)
    String cast method for interface and class type conversions

void
bHYPRE_Operator__exec ( bHYPRE_Operator self,
                        const char* methodName,
                        sidl_io_Deserializer inArgs,
                        sidl_io_Serializer outArgs)
    Select and execute a method by name

char*
bHYPRE_Operator__getURL ( bHYPRE_Operator self)
    Get the URL of the Implementation of this object (for RMI)

```

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

```
int32_t  
bHYPRE_Operator_SetCommunicator ( bHYPRE_Operator self,  
bHYPRE_MPICommunicator mpi_comm)
```

Set the MPI Communicator. DEPRECATED, use Create:

Vector Interface

Names

3.1	struct bHYPRE_Vector__object <i>Symbol "bHYPRE"</i> extern C bHYPRE_Vector bHYPRE_Vector__connect (const char *, sidl_BaseInterface *_ex) <i>RMI connector function for the class</i> int32_t bHYPRE_Vector_Clear (bHYPRE_Vector self) <i>Set self to 0</i> int32_t bHYPRE_Vector_Copy (bHYPRE_Vector self, bHYPRE_Vector x) <i>Copy data from x into self</i>	30
3.2	int32_t bHYPRE_Vector_Clone (bHYPRE_Vector self, bHYPRE_Vector* x) <i>Create an x compatible with self</i> int32_t bHYPRE_Vector_Scale (bHYPRE_Vector self, double a) <i>Scale self by a</i> int32_t bHYPRE_Vector_Dot (bHYPRE_Vector self, bHYPRE_Vector x, double* d) <i>Compute d, the inner-product of self and x</i> int32_t bHYPRE_Vector_Axpy (bHYPRE_Vector self, double a, bHYPRE_Vector x) <i>Add a*x to self</i> struct bHYPRE_Vector__object* bHYPRE_Vector__cast void* obj <i>Cast method for interface and class type conversions</i> void* bHYPRE_Vector__cast2 (void* obj, const char* type) <i>String cast method for interface and class type conversions</i> void bHYPRE_Vector__exec (bHYPRE_Vector self, const char* methodName, sidl_io_Deserializer inArgs, sidl_io_Serializer outArgs) <i>Select and execute a method by name</i> char* bHYPRE_Vector__getURL (bHYPRE_Vector self) <i>Get the URL of the Implementation of this object (for RMI)</i>	30

3.1

```
struct bHYPRE_Vector__object
```

Symbol "bHYPRE.Vector" (version 1.0.0)

3.2

```
int32_t bHYPRE_Vector_Clone ( bHYPRE_Vector self, bHYPRE_Vector* x)
```

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.

Matrices and Vectors

Names

4.1	IJParCSR Matrix	31
4.2	IJParCSR Vector	40
4.3	Struct Matrix	45
4.4	Struct Vector	50
4.5	SemiStructured Matrix	53
4.6	SemiStructured Vector	61
4.7	SemiStructured ParCSR Matrix	68
4.8	SemiStructured ParCSR Vector	77

IJParCSR Matrix

Names

4.1.1	struct bHYPRE_IJParCSRMatrix__object <i>Symbol "bHYPRE"</i>	36
	void <i>Constructor function for the class</i>	
	bHYPRE_IJParCSRMatrix bHYPRE_IJParCSRMatrix__createRemote (const char *, sidl_BaseInterface *_ex) <i>RMI constructor function for the class</i>	
	bHYPRE_IJParCSRMatrix bHYPRE_IJParCSRMatrix__connect (const char *, sidl_BaseInterface *_ex) <i>RMI connector function for the class</i>	
	bHYPRE_IJParCSRMatrix	

bHYPRE_IJParCSRMatrix_Create (bHYPRE_MPICommunicator
mpi_comm, int32_t ilower,
int32_t iupper, int32_t jlower,
int32_t jupper)

Method: Create[]

bHYPRE_IJParCSRMatrix

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)

Method: GenerateLaplacian[]

- 4.1.2 int32_t
bHYPRE_IJParCSRMatrix_SetDiagOffdSizes (bHYPRE_IJParCSRMatrix
self, int32_t* diag_sizes,
int32_t* offdiag_sizes,
int32_t local_nrows)
*(Optional) Set the max number of nonzeros to expect in each row of the
diagonal and off-diagonal blocks 36*
- 4.1.3 int32_t
bHYPRE_IJParCSRMatrix_SetCommunicator (
bHYPRE_IJParCSRMatrix
self,
bHYPRE_MPICommunicator
mpi_comm)
Set the MPI Communicator 36
- int32_t
bHYPRE_IJParCSRMatrix_Initialize (bHYPRE_IJParCSRMatrix self)
*Prepare an object for setting coefficient values, whether for the first time or
subsequently*
- 4.1.4 int32_t
bHYPRE_IJParCSRMatrix_Assemble (bHYPRE_IJParCSRMatrix self)
*Finalize the construction of an object before using, either for the first time
or on subsequent uses 37*
- 4.1.5 int32_t
bHYPRE_IJParCSRMatrix_SetLocalRange (bHYPRE_IJParCSRMatrix
self, int32_t ilower,
int32_t iupper, int32_t jlower,
int32_t jupper)
Set the local range for a matrix object 37
- 4.1.6 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)	
		<i>Sets values for nrows of the matrix</i>	37
4.1.7	int32_t	bHYPRE_IJParCSRMatrix_AddToValues (bHYPRE_IJParCSRMatrix self, int32_t nrows, int32_t* ncols, int32_t* rows, int32_t* cols, double* values, int32_t nnonzeros)	
		<i>Adds to values for nrows of the matrix</i>	38
	int32_t	bHYPRE_IJParCSRMatrix_GetLocalRange (bHYPRE_IJParCSRMatrix self, int32_t* ilower, int32_t* iupper, int32_t* jlower, int32_t* jupper)	
		<i>Gets range of rows owned by this processor and range of column partitioning for this processor</i>	
	int32_t	bHYPRE_IJParCSRMatrix_GetRowCounts (bHYPRE_IJParCSRMatrix self, int32_t nrows, int32_t* rows, int32_t* ncols)	
		<i>Gets number of nonzeros elements for nrows rows specified in rows and returns them in ncols, which needs to be allocated by the user</i>	
4.1.8	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)	
		<i>Gets values for nrows rows or partial rows of the matrix</i>	38
4.1.9	int32_t	bHYPRE_IJParCSRMatrix_SetRowSizes (bHYPRE_IJParCSRMatrix self, int32_t* sizes, int32_t nrows)	
		<i>(Optional) Set the max number of nonzeros to expect in each row</i>	38
4.1.10	int32_t	bHYPRE_IJParCSRMatrix_Print (bHYPRE_IJParCSRMatrix self, const char* filename)	
		<i>Print the matrix to file</i>	39
4.1.11	int32_t	bHYPRE_IJParCSRMatrix_Read (bHYPRE_IJParCSRMatrix self, const char* filename, bHYPRE_MPICommunicator comm)	
		<i>Read the matrix from file</i>	39
	int32_t		

bHYPRE_IJParCSRMatrix_SetIntParameter (bHYPRE_IJParCSRMatrix
self, const char* name,
int32_t value)

Set the int parameter associated with name

int32_t

bHYPRE_IJParCSRMatrix_SetDoubleParameter (
bHYPRE_IJParCSRMatrix
self, const char* name,
double value)

Set the double parameter associated with name

int32_t

bHYPRE_IJParCSRMatrix_SetStringParameter (
bHYPRE_IJParCSRMatrix
self, const char* name,
const char* value)

Set the string parameter associated with name

int32_t

bHYPRE_IJParCSRMatrix_SetIntArray1Parameter (
bHYPRE_IJParCSRMatrix
self,
const char* name,
int32_t* value,
int32_t nvalues)

Set the int 1-D array parameter associated with name

int32_t

bHYPRE_IJParCSRMatrix_SetIntArray2Parameter (
bHYPRE_IJParCSRMatrix
self,
const char* name,
struct
sidl_int__array*
value)

Set the int 2-D array parameter associated with name

int32_t

bHYPRE_IJParCSRMatrix_SetDoubleArray1Parameter (
bHYPRE_IJParCSRMatrix
self, const
char* name,
double* value,
int32_t nvalues)

Set the double 1-D array parameter associated with name

int32_t

bHYPRE_IJParCSRMatrix_SetDoubleArray2Parameter (
bHYPRE_IJParCSRMatrix
self, const
char* name,
struct
sidl_double__array*
value)

Set the double 2-D array parameter associated with name

int32_t

bHYPRE_IJParCSRMatrix_GetIntValue (bHYPRE_IJParCSRMatrix self,
 const char* name,
 int32_t* value)

Set the int parameter associated with name

int32_t

bHYPRE_IJParCSRMatrix_GetDoubleValue (bHYPRE_IJParCSRMatrix
 self, const char* name,
 double* value)

Get the double parameter associated with name

int32_t

bHYPRE_IJParCSRMatrix_Setup (bHYPRE_IJParCSRMatrix self,
 bHYPRE_Vector b, bHYPRE_Vector x)

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

int32_t

bHYPRE_IJParCSRMatrix_Apply (bHYPRE_IJParCSRMatrix self,
 bHYPRE_Vector b,
 bHYPRE_Vector* x)

Apply the operator to b, returning x

int32_t

bHYPRE_IJParCSRMatrix_ApplyAdjoint (bHYPRE_IJParCSRMatrix
 self, bHYPRE_Vector b,
 bHYPRE_Vector* x)

Apply the adjoint of the operator to b, returning x

4.1.12

int32_t

bHYPRE_IJParCSRMatrix_GetRow (bHYPRE_IJParCSRMatrix self,
 int32_t row, int32_t* size,
 struct sidl_int_array** colInd,
 struct sidl_double_array** values)

*The GetRow method will allocate space for its two output arrays on the first
 call*

39

obj

Cast method for interface and class type conversions

void*

bHYPRE_IJParCSRMatrix__cast2 (void* obj, const char* type)

String cast method for interface and class type conversions

void

bHYPRE_IJParCSRMatrix__exec (bHYPRE_IJParCSRMatrix self,
 const char* methodName,
 sidl_io_Deserializer inArgs,
 sidl_io_Serializer outArgs)

Select and execute a method by name

char*

bHYPRE_IJParCSRMatrix__getURL (bHYPRE_IJParCSRMatrix self)

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

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 `__cast` methods.

4.1.2

```
int32_t
bHYPRE_IJParCSRMatrix_SetDiagOffdSizes ( bHYPRE_IJParCSRMatrix
self, int32_t* diag_sizes, int32_t* offdiag_sizes, int32_t local_nrows)
```

(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.

4.1.3

```
int32_t
bHYPRE_IJParCSRMatrix_SetCommunicator ( bHYPRE_IJParCSRMatrix
self, bHYPRE_MPICommunicator mpi_comm)
```

Set the MPI Communicator. DEPRECATED, Use Create()

4.1.4

```
int32_t
bHYPRE_IJParCSRMatrix_Assemble ( bHYPRE_IJParCSRMatrix self)
```

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**.

4.1.5

```
int32_t
bHYPRE_IJParCSRMatrix_SetLocalRange ( bHYPRE_IJParCSRMatrix
self, int32_t ilower, int32_t iupper, int32_t jlower, int32_t jupper)
```

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.

4.1.6

```
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)
```

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 function 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.

4.1.7

```
int32_t
bHYPRE_IJParCSRMatrix_AddToValues ( bHYPRE_IJParCSRMatrix self,
int32_t nrows, int32_t* ncols, int32_t* rows, int32_t* cols, double* values, int32_t
nnonzeros)
```

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.

4.1.8

```
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)
```

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

4.1.9

```
int32_t
bHYPRE_IJParCSRMatrix_SetRowSizes ( bHYPRE_IJParCSRMatrix self,
int32_t* sizes, int32_t nrows)
```

(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.

4.1.10

```
int32_t
bHYPRE_IJParCSRMatrix_Print ( bHYPRE_IJParCSRMatrix self, const
char* filename)
```

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

4.1.11

```
int32_t
bHYPRE_IJParCSRMatrix_Read ( bHYPRE_IJParCSRMatrix self, const
char* filename, bHYPRE_MPICommunicator comm)
```

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

4.1.12

```
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)
```

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.

IJParCSR Vector

4.2.1	<pre> struct bHYPRE_IJParCSRVector__object Symbol "bHYPRE" </pre>	42
	<pre> void Constructor function for the class bHYPRE_IJParCSRVector bHYPRE_IJParCSRVector__createRemote (const char *, sidl_BaseInterface *_ex) RMI constructor function for the class bHYPRE_IJParCSRVector bHYPRE_IJParCSRVector__connect (const char *, sidl_BaseInterface *_ex) RMI connector function for the class bHYPRE_IJParCSRVector bHYPRE_IJParCSRVector__Create (bHYPRE_MPICommunicator mpi_comm, int32_t jlower, int32_t jupper) Method: Create[] </pre>	
4.2.2	<pre> int32_t bHYPRE_IJParCSRVector_SetCommunicator (bHYPRE_IJParCSRVector self, bHYPRE_MPICommunicator mpi_comm) Set the MPI Communicator </pre>	42
	<pre> int32_t bHYPRE_IJParCSRVector_Initialize (bHYPRE_IJParCSRVector self) Prepare an object for setting coefficient values, whether for the first time or subsequently </pre>	
4.2.3	<pre> int32_t bHYPRE_IJParCSRVector_Assemble (bHYPRE_IJParCSRVector self) Finalize the construction of an object before using, either for the first time or on subsequent uses </pre>	43
4.2.4	<pre> int32_t bHYPRE_IJParCSRVector_SetLocalRange (bHYPRE_IJParCSRVector self, int32_t jlower, int32_t jupper) Set the local range for a vector object </pre>	43
4.2.5	<pre> int32_t </pre>	

	bHYPRE_IJParCSRVector_SetValues (bHYPRE_IJParCSRVector self, int32_t nvalues, int32_t* indices, double* values)	
	<i>Sets values in vector</i>	43
4.2.6	int32_t bHYPRE_IJParCSRVector_AddToValues (bHYPRE_IJParCSRVector self, int32_t nvalues, int32_t* indices, double* values)	
	<i>Adds to values in vector</i>	44
	int32_t bHYPRE_IJParCSRVector_GetLocalRange (bHYPRE_IJParCSRVector self, int32_t* jlower, int32_t* jupper)	
	<i>Returns range of the part of the vector owned by this processor</i>	
4.2.7	int32_t bHYPRE_IJParCSRVector_GetValues (bHYPRE_IJParCSRVector self, int32_t nvalues, int32_t* indices, double* values)	
	<i>Gets values in vector</i>	44
4.2.8	int32_t bHYPRE_IJParCSRVector_Print (bHYPRE_IJParCSRVector self, const char* filename)	
	<i>Print the vector to file</i>	44
4.2.9	int32_t bHYPRE_IJParCSRVector_Read (bHYPRE_IJParCSRVector self, const char* filename, bHYPRE_MPICommunicator comm)	
	<i>Read the vector from file</i>	45
	int32_t bHYPRE_IJParCSRVector_Clear (bHYPRE_IJParCSRVector self)	
	<i>Set self to 0</i>	
	int32_t bHYPRE_IJParCSRVector_Copy (bHYPRE_IJParCSRVector self, bHYPRE_Vector x)	
	<i>Copy data from x into self</i>	
4.2.10	int32_t bHYPRE_IJParCSRVector_Clone (bHYPRE_IJParCSRVector self, bHYPRE_Vector* x)	
	<i>Create an x compatible with self</i>	45
	int32_t bHYPRE_IJParCSRVector_Scale (bHYPRE_IJParCSRVector self, double a)	
	<i>Scale self by a</i>	
	int32_t bHYPRE_IJParCSRVector_Dot (bHYPRE_IJParCSRVector self, bHYPRE_Vector x, double* d)	
	<i>Compute d, the inner-product of self and x</i>	
	int32_t	

```

bHYPRE_IJParCSRVector_Axpy ( bHYPRE_IJParCSRVector self,
                               double a, bHYPRE_Vector x)
    Add a*x to self

obj
    Cast method for interface and class type conversions

void*
bHYPRE_IJParCSRVector__cast2 ( void* obj, const char* type)
    String cast method for interface and class type conversions

void
bHYPRE_IJParCSRVector__exec ( bHYPRE_IJParCSRVector self,
                               const char* methodName,
                               sidl_io_Deserializer inArgs,
                               sidl_io_Serializer outArgs)
    Select and execute a method by name

char*
bHYPRE_IJParCSRVector__getURL ( bHYPRE_IJParCSRVector self)
    Get the URL of the Implementation of this object (for RMI)

```

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

```

int32_t
bHYPRE_IJParCSRVector_SetCommunicator ( bHYPRE_IJParCSRVector
self, bHYPRE_MPICommunicator mpi_comm)

```

Set the MPI Communicator. DEPRECATED, Use Create()

4.2.3

```
int32_t
bHYPRE_IJParCSRVector_Assemble ( bHYPRE_IJParCSRVector self)
```

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**.

4.2.4

```
int32_t
bHYPRE_IJParCSRVector_SetLocalRange ( bHYPRE_IJParCSRVector self,
int32_t jlower, int32_t jupper)
```

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.

4.2.5

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

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.

4.2.6

```
int32_t  
bHYPRE_IJParCSRVector_AddToValues ( bHYPRE_IJParCSRVector self,  
int32_t nvalues, int32_t* indices, double* values)
```

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

Not collective.

4.2.7

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

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

Not collective.

4.2.8

```
int32_t  
bHYPRE_IJParCSRVector_Print ( bHYPRE_IJParCSRVector self, const  
char* filename)
```

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

4.2.9

```
int32_t
bHYPRE_IJParCSRVector_Read ( bHYPRE_IJParCSRVector self, const
char* filename, bHYPRE_MPICommunicator comm)
```

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

4.2.10

```
int32_t
bHYPRE_IJParCSRVector_Clone ( bHYPRE_IJParCSRVector self,
bHYPRE_Vector* x)
```

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.

4.3**Struct Matrix****Names**

4.3.1	struct bHYPRE_StructMatrix__object <i>Symbol "bHYPRE"</i>	49
	void <i>Constructor function for the class</i>	
	bHYPRE_StructMatrix bHYPRE_StructMatrix__createRemote (const char *, sidl_BaseInterface *_ex) <i>RMI constructor function for the class</i>	
	bHYPRE_StructMatrix bHYPRE_StructMatrix__connect (const char *, sidl_BaseInterface *_ex) <i>RMI connector function for the class</i>	
	bHYPRE_StructMatrix	

	bHYPRE_StructMatrix_Create (bHYPRE_MPICommunicator mpi_comm, bHYPRE_StructGrid grid, bHYPRE_StructStencil stencil)	
	<i>Method: Create[]</i>	
4.3.2	int32_t bHYPRE_StructMatrix_SetCommunicator (bHYPRE_StructMatrix self, bHYPRE_MPICommunicator mpi_comm)	
	<i>Set the MPI Communicator</i>	49
	int32_t bHYPRE_StructMatrix_Initialize (bHYPRE_StructMatrix self)	
	<i>Prepare an object for setting coefficient values, whether for the first time or subsequently</i>	
4.3.3	int32_t bHYPRE_StructMatrix_Assemble (bHYPRE_StructMatrix self)	
	<i>Finalize the construction of an object before using, either for the first time or on subsequent uses</i>	49
	int32_t bHYPRE_StructMatrix_SetGrid (bHYPRE_StructMatrix self, bHYPRE_StructGrid grid)	
	<i>Method: SetGrid[]</i>	
	int32_t bHYPRE_StructMatrix_SetStencil (bHYPRE_StructMatrix self, bHYPRE_StructStencil stencil)	
	<i>Method: SetStencil[]</i>	
	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)	
	<i>Method: SetValues[]</i>	
	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)	
	<i>Method: SetBoxValues[]</i>	
	int32_t bHYPRE_StructMatrix_SetNumGhost (bHYPRE_StructMatrix self, int32_t* num_ghost, int32_t dim2)	
	<i>Method: SetNumGhost[]</i>	
	int32_t bHYPRE_StructMatrix_SetSymmetric (bHYPRE_StructMatrix self, int32_t symmetric)	
	<i>Method: SetSymmetric[]</i>	
	int32_t	

int32_t

```

bHYPRE_StructMatrix_SetDoubleArray2Parameter (
    bHYPRE_StructMatrix
    self,
    const char* name,
    struct
    sidl_double__array*
    value)
    Set the double 2-D array parameter associated with name

int32_t
bHYPRE_StructMatrix_GetIntValue ( bHYPRE_StructMatrix self,
    const char* name, int32_t* value)
    Set the int parameter associated with name

int32_t
bHYPRE_StructMatrix_GetDoubleValue ( bHYPRE_StructMatrix self,
    const char* name,
    double* value)
    Get the double parameter associated with name

int32_t
bHYPRE_StructMatrix_Setup ( bHYPRE_StructMatrix self,
    bHYPRE_Vector b, bHYPRE_Vector x)
    (Optional) Do any preprocessing that may be necessary in order to execute
    Apply

int32_t
bHYPRE_StructMatrix_Apply ( bHYPRE_StructMatrix self,
    bHYPRE_Vector b, bHYPRE_Vector* x)
    Apply the operator to b, returning x

int32_t
bHYPRE_StructMatrix_ApplyAdjoint ( bHYPRE_StructMatrix self,
    bHYPRE_Vector b,
    bHYPRE_Vector* x)
    Apply the adjoint of the operator to b, returning x

struct bHYPRE_StructMatrix__object* bHYPRE_StructMatrix__cast void* obj
    Cast method for interface and class type conversions

void*
bHYPRE_StructMatrix__cast2 ( void* obj, const char* type)
    String cast method for interface and class type conversions

void
bHYPRE_StructMatrix__exec ( bHYPRE_StructMatrix self,
    const char* methodName,
    sidl_io_Deserializer inArgs,
    sidl_io_Serializer outArgs)
    Select and execute a method by name

char*
bHYPRE_StructMatrix__getURL ( bHYPRE_StructMatrix self)
    Get the URL of the Implementation of this object (for RMI)

```


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

```
int32_t  
bHYPRE_StructMatrix_SetCommunicator ( bHYPRE_StructMatrix self,  
bHYPRE_MPICommunicator mpi_comm)
```

Set the MPI Communicator. DEPRECATED, Use Create()

4.3.3

```
int32_t bHYPRE_StructMatrix_Assemble ( bHYPRE_StructMatrix self)
```

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.

```

bHYPRE_StructVector_SetBoxValues ( bHYPRE_StructVector self,
                                     int32_t* ilower, int32_t* iupper,
                                     int32_t dim, double* values,
                                     int32_t nvalues)

    Method: SetBoxValues[]

int32_t
bHYPRE_StructVector_Clear ( bHYPRE_StructVector self)
    Set self to 0

int32_t
bHYPRE_StructVector_Copy ( bHYPRE_StructVector self,
                             bHYPRE_Vector x)
    Copy data from x into self

4.4.4 int32_t
bHYPRE_StructVector_Clone ( bHYPRE_StructVector self,
                             bHYPRE_Vector* x)
    Create an x compatible with self ..... 52

int32_t
bHYPRE_StructVector_Scale ( bHYPRE_StructVector self, double a)
    Scale self by a

int32_t
bHYPRE_StructVector_Dot ( bHYPRE_StructVector self,
                             bHYPRE_Vector x, double* d)
    Compute d, the inner-product of self and x

int32_t
bHYPRE_StructVector_Axpy ( bHYPRE_StructVector self, double a,
                             bHYPRE_Vector x)
    Add a*x to self

struct bHYPRE_StructVector__object* bHYPRE_StructVector__cast void* obj
    Cast method for interface and class type conversions

void*
bHYPRE_StructVector__cast2 ( void* obj, const char* type)
    String cast method for interface and class type conversions

void
bHYPRE_StructVector__exec ( bHYPRE_StructVector self,
                             const char* methodName,
                             sidl_io_Deserializer inArgs,
                             sidl_io_Serializer outArgs)
    Select and execute a method by name

char*
bHYPRE_StructVector__getURL ( bHYPRE_StructVector self)
    Get the URL of the Implementation of this object (for RMI)

```

4.4.1

```
struct bHYPRE_StructVector__object
```

Symbol "bHYPRE.StructVector" (version 1.0.0)

4.4.2

```
int32_t  
bHYPRE_StructVector_SetCommunicator ( bHYPRE_StructVector self,  
bHYPRE_MPICommunicator mpi_comm)
```

Set the MPI Communicator. DEPRECATED, Use Create()

4.4.3

```
int32_t bHYPRE_StructVector_Assemble ( bHYPRE_StructVector self)
```

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.

4.4.4

```
int32_t  
bHYPRE_StructVector_Clone ( bHYPRE_StructVector self, bHYPRE_Vector*  
x)
```

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

4.5 SemiStructured Matrix

4.5.1	<pre> struct bHYPRE_SStructMatrix__object Symbol "bHYPRE" </pre>	57
	<pre> void Constructor function for the class bHYPRE_SStructMatrix bHYPRE_SStructMatrix__createRemote (const char *, sidl_BaseInterface *_ex) RMI constructor function for the class bHYPRE_SStructMatrix bHYPRE_SStructMatrix__connect (const char *, sidl_BaseInterface *_ex) RMI connector function for the class bHYPRE_SStructMatrix bHYPRE_SStructMatrix_Create (bHYPRE_MPICommunicator mpi_comm, bHYPRE_SStructGraph graph) Method: Create[] int32_t bHYPRE_SStructMatrix_SetObjectType (bHYPRE_SStructMatrix self, int32_t type) Method: SetObjectType[] </pre>	
4.5.2	<pre> int32_t bHYPRE_SStructMatrix_SetCommunicator (bHYPRE_SStructMatrix self, bHYPRE_MPICommunicator mpi_comm) Set the MPI Communicator </pre>	57
	<pre> int32_t bHYPRE_SStructMatrix_Initialize (bHYPRE_SStructMatrix self) Prepare an object for setting coefficient values, whether for the first time or subsequently </pre>	
4.5.3	<pre> int32_t bHYPRE_SStructMatrix_Assemble (bHYPRE_SStructMatrix self) Finalize the construction of an object before using, either for the first time or on subsequent uses </pre>	57
4.5.4	<pre> int32_t </pre>	

	bHYPRE_SStructMatrix_GetObject (bHYPRE_SStructMatrix self, sidl_BaseInterface* A) <i>A semi-structured matrix or vector contains a Struct or IJ matrix or vector</i>	58
4.5.5	int32_t bHYPRE_SStructMatrix_SetGraph (bHYPRE_SStructMatrix self, bHYPRE_SStructGraph graph) <i>Set the matrix graph</i>	58
4.5.6	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) <i>Set matrix coefficients index by index</i>	58
4.5.7	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) <i>Set matrix coefficients a box at a time</i>	59
4.5.8	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) <i>Add to matrix coefficients index by index</i>	59
4.5.9	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) <i>Add to matrix coefficients a box at a time</i>	60
4.5.10	int32_t bHYPRE_SStructMatrix_SetSymmetric (bHYPRE_SStructMatrix self, int32_t part, int32_t var, int32_t to_var, int32_t symmetric) <i>Define symmetry properties for the stencil entries in the matrix</i>	60
	int32_t bHYPRE_SStructMatrix_SetNSSymmetric (bHYPRE_SStructMatrix self, int32_t symmetric) <i>Define symmetry properties for all non-stencil matrix entries</i> int32_t	

	bHYPRE_SStructMatrix_SetComplex (bHYPRE_SStructMatrix self)	
	<i>Set the matrix to be complex</i>	
4.5.11	int32_t bHYPRE_SStructMatrix_Print (bHYPRE_SStructMatrix self, const char* filename, int32_t all)	
	<i>Print the matrix to file</i>	60
	int32_t bHYPRE_SStructMatrix_SetIntParameter (bHYPRE_SStructMatrix self, const char* name, int32_t value)	
	<i>Set the int parameter associated with name</i>	
	int32_t bHYPRE_SStructMatrix_SetDoubleParameter (bHYPRE_SStructMatrix self, const char* name, double value)	
	<i>Set the double parameter associated with name</i>	
	int32_t bHYPRE_SStructMatrix_SetStringParameter (bHYPRE_SStructMatrix self, const char* name, const char* value)	
	<i>Set the string parameter associated with name</i>	
	int32_t bHYPRE_SStructMatrix_SetIntArray1Parameter (
	bHYPRE_SStructMatrix self, const char* name, int32_t* value, int32_t nvalues)	
	<i>Set the int 1-D array parameter associated with name</i>	
	int32_t bHYPRE_SStructMatrix_SetIntArray2Parameter (
	bHYPRE_SStructMatrix self, const char* name, struct sidl_int_array* value)	
	<i>Set the int 2-D array parameter associated with name</i>	
	int32_t bHYPRE_SStructMatrix_SetDoubleArray1Parameter (
	bHYPRE_SStructMatrix self, const char* name, double* value, int32_t nvalues)	
	<i>Set the double 1-D array parameter associated with name</i>	
	int32_t	

```

bHYPRE_SStructMatrix_SetDoubleArray2Parameter (
    bHYPRE_SStructMatrix
    self,
    const char* name,
    struct
    sidl_double__array*
    value)
    Set the double 2-D array parameter associated with name

int32_t
bHYPRE_SStructMatrix_GetIntValue ( bHYPRE_SStructMatrix self,
    const char* name, int32_t* value)
    Set the int parameter associated with name

int32_t
bHYPRE_SStructMatrix_GetDoubleValue ( bHYPRE_SStructMatrix self,
    const char* name,
    double* value)
    Get the double parameter associated with name

int32_t
bHYPRE_SStructMatrix_Setup ( bHYPRE_SStructMatrix self,
    bHYPRE_Vector b, bHYPRE_Vector x)
    (Optional) Do any preprocessing that may be necessary in order to execute
    Apply

int32_t
bHYPRE_SStructMatrix_Apply ( bHYPRE_SStructMatrix self,
    bHYPRE_Vector b, bHYPRE_Vector* x)
    Apply the operator to b, returning x

int32_t
bHYPRE_SStructMatrix_ApplyAdjoint ( bHYPRE_SStructMatrix self,
    bHYPRE_Vector b,
    bHYPRE_Vector* x)
    Apply the adjoint of the operator to b, returning x

obj
    Cast method for interface and class type conversions

void*
bHYPRE_SStructMatrix__cast2 ( void* obj, const char* type)
    String cast method for interface and class type conversions

void
bHYPRE_SStructMatrix__exec ( bHYPRE_SStructMatrix self,
    const char* methodName,
    sidl_io_Deserializer inArgs,
    sidl_io_Serializer outArgs)
    Select and execute a method by name

char*
bHYPRE_SStructMatrix__getURL ( bHYPRE_SStructMatrix self)
    Get the URL of the Implementation of this object (for RMI)

```


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

```
int32_t  
bHYPRE_SStructMatrix_SetCommunicator ( bHYPRE_SStructMatrix self,  
bHYPRE_MPICommunicator mpi_comm)
```

Set the MPI Communicator. DEPRECATED, Use Create()

4.5.3

```
int32_t bHYPRE_SStructMatrix_Assemble ( bHYPRE_SStructMatrix self)
```

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.

4.5.4

```
int32_t
bHYPRE_SStructMatrix_GetObject ( bHYPRE_SStructMatrix self,
sidl_BaseInterface* A)
```

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

4.5.5

```
int32_t
bHYPRE_SStructMatrix_SetGraph ( bHYPRE_SStructMatrix self,
bHYPRE_SStructGraph graph)
```

Set the matrix graph. DEPRECATED Use `Create`

4.5.6

```
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)
```

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.

4.5.7

```
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)
```

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.

4.5.8

```
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)
```

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.

4.5.9

```
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)
```

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.

4.5.10

```
int32_t
bHYPRE_SStructMatrix_SetSymmetric ( bHYPRE_SStructMatrix self,
int32_t part, int32_t var, int32_t to_var, int32_t symmetric)
```

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.

4.5.11

```
int32_t
bHYPRE_SStructMatrix_Print ( bHYPRE_SStructMatrix self, const char*
filename, int32_t all)
```

SemiStructured Vector

4.6.1	<pre> struct bHYPRE_SStructVector__object Symbol "bHYPRE" </pre>	64
	<pre> void Constructor function for the class </pre>	
	<pre> bHYPRE_SStructVector bHYPRE_SStructVector__createRemote (const char *, sidl_BaseInterface *_ex) RMI constructor function for the class </pre>	
	<pre> bHYPRE_SStructVector bHYPRE_SStructVector__connect (const char *, sidl_BaseInterface *_ex) RMI connector function for the class </pre>	
	<pre> bHYPRE_SStructVector bHYPRE_SStructVector_Create (bHYPRE_MPICommunicator mpi_comm, bHYPRE_SStructGrid grid) Method: Create[] </pre>	
	<pre> int32_t bHYPRE_SStructVector_SetObjectType (bHYPRE_SStructVector self, int32_t type) Method: SetObjectType[] </pre>	
4.6.2	<pre> int32_t bHYPRE_SStructVector_SetCommunicator (bHYPRE_SStructVector self, bHYPRE_MPICommunicator mpi_comm) Set the MPI Communicator </pre>	64
	<pre> int32_t bHYPRE_SStructVector_Initialize (bHYPRE_SStructVector self) Prepare an object for setting coefficient values, whether for the first time or subsequently </pre>	
4.6.3	<pre> int32_t bHYPRE_SStructVector_Assemble (bHYPRE_SStructVector self) Finalize the construction of an object before using, either for the first time or on subsequent uses </pre>	64
4.6.4	<pre> int32_t </pre>	

	bHYPRE_SStructVector_GetObject (bHYPRE_SStructVector self, sidl_BaseInterface* A) <i>A semi-structured matrix or vector contains a Struct or IJ matrix or vector</i>	65
	int32_t bHYPRE_SStructVector_SetGrid (bHYPRE_SStructVector self, bHYPRE_SStructGrid grid) <i>Set the vector grid</i>	
4.6.5	int32_t bHYPRE_SStructVector_SetValues (bHYPRE_SStructVector self, int32_t part, int32_t* index, int32_t dim, int32_t var, double value) <i>Set vector coefficients index by index</i>	65
4.6.6	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) <i>Set vector coefficients a box at a time</i>	65
4.6.7	int32_t bHYPRE_SStructVector_AddToValues (bHYPRE_SStructVector self, int32_t part, int32_t* index, int32_t dim, int32_t var, double value) <i>Set vector coefficients index by index</i>	66
4.6.8	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 nvalues) <i>Set vector coefficients a box at a time</i>	66
	int32_t bHYPRE_SStructVector_Gather (bHYPRE_SStructVector self) <i>Gather vector data before calling GetValues</i>	
4.6.9	int32_t bHYPRE_SStructVector_GetValues (bHYPRE_SStructVector self, int32_t part, int32_t* index, int32_t dim, int32_t var, double* value) <i>Get vector coefficients index by index</i>	67
4.6.10	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) <i>Get vector coefficients a box at a time</i>	67
	int32_t	

	bHYPRE_SStructVector_SetComplex (bHYPRE_SStructVector self) <i>Set the vector to be complex</i>	
4.6.11	int32_t bHYPRE_SStructVector_Print (bHYPRE_SStructVector self, const char* filename, int32_t all) <i>Print the vector to file</i>	67
	int32_t bHYPRE_SStructVector_Clear (bHYPRE_SStructVector self) <i>Set self to 0</i>	
	int32_t bHYPRE_SStructVector_Copy (bHYPRE_SStructVector self, bHYPRE_Vector x) <i>Copy data from x into self</i>	
4.6.12	int32_t bHYPRE_SStructVector_Clone (bHYPRE_SStructVector self, bHYPRE_Vector* x) <i>Create an x compatible with self</i>	68
	int32_t bHYPRE_SStructVector_Scale (bHYPRE_SStructVector self, double a) <i>Scale self by a</i>	
	int32_t bHYPRE_SStructVector_Dot (bHYPRE_SStructVector self, bHYPRE_Vector x, double* d) <i>Compute d, the inner-product of self and x</i>	
	int32_t bHYPRE_SStructVector_Axpy (bHYPRE_SStructVector self, double a, bHYPRE_Vector x) <i>Add a*x to self</i>	
	struct bHYPRE_SStructVector__object* bHYPRE_SStructVector__cast void* obj <i>Cast method for interface and class type conversions</i>	
	void* bHYPRE_SStructVector__cast2 (void* obj, const char* type) <i>String cast method for interface and class type conversions</i>	
	void bHYPRE_SStructVector__exec (bHYPRE_SStructVector self, const char* methodName, sidl_io_Deserializer inArgs, sidl_io_Serializer outArgs) <i>Select and execute a method by name</i>	
	char* bHYPRE_SStructVector__getURL (bHYPRE_SStructVector self) <i>Get the URL of the Implementation of this object (for RMI)</i>	

4.6.1

```
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

```
int32_t  
bHYPRE_SStructVector_SetCommunicator ( bHYPRE_SStructVector self,  
bHYPRE_MPICommunicator mpi_comm)
```

Set the MPI Communicator. DEPRECATED, Use Create()

4.6.3

```
int32_t bHYPRE_SStructVector_Assemble ( bHYPRE_SStructVector self)
```

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.

4.6.4

```
int32_t
bHYPRE_SStructVector_GetObject ( bHYPRE_SStructVector self,
sidl_BaseInterface* A)
```

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

4.6.5

```
int32_t
bHYPRE_SStructVector_SetValues ( bHYPRE_SStructVector self, int32_t
part, int32_t* index, int32_t dim, int32_t var, double value)
```

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_SetBoxValues ( bHYPRE_SStructVector self,
int32_t part, int32_t* ilower, int32_t* iupper, int32_t dim, int32_t var, double*
values, int32_t nvalues)
```

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

```
int32_t
bHYPRE_SStructVector_AddToValues ( bHYPRE_SStructVector self, int32_t
part, int32_t* index, int32_t dim, int32_t var, double value)
```

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.

4.6.8

```
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 nvalues)
```

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.

4.6.9

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

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.

4.6.10

```
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)
```

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.

4.6.11

```
int32_t
bHYPRE_SStructVector_Print ( bHYPRE_SStructVector self, const char*
filename, int32_t all)
```

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

4.6.12

```
int32_t
bHYPRE_SStructVector_Clone ( bHYPRE_SStructVector self,
bHYPRE_Vector* x)
```

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.

4.7

SemiStructured ParCSR Matrix

Names

4.7.1	struct bHYPRE_SStructParCSRMatrix__object <i>Symbol "bHYPRE"</i>	73
	void <i>Constructor function for the class</i>	
	bHYPRE_SStructParCSRMatrix bHYPRE_SStructParCSRMatrix__createRemote (const char *, sidl_BaseInterface *_ex) <i>RMI constructor function for the class</i>	
	bHYPRE_SStructParCSRMatrix bHYPRE_SStructParCSRMatrix__connect (const char *, sidl_BaseInterface *_ex) <i>RMI connector function for the class</i>	
	bHYPRE_SStructParCSRMatrix bHYPRE_SStructParCSRMatrix_Create (bHYPRE_MPICommunicator mpi_comm, bHYPRE_SStructGraph graph) <i>Method: Create[]</i>	
4.7.2	int32_t bHYPRE_SStructParCSRMatrix_SetCommunicator (bHYPRE_SStructParCSRMatrix self, bHYPRE_MPICommunicator mpi_comm) <i>Set the MPI Communicator</i>	73
	int32_t	

	bHYPRE_SStructParCSRMatrix_Initialize (bHYPRE_SStructParCSRMatrix self) <i>Prepare an object for setting coefficient values, whether for the first time or subsequently</i>	
4.7.3	int32_t bHYPRE_SStructParCSRMatrix_Assemble (bHYPRE_SStructParCSRMatrix self) <i>Finalize the construction of an object before using, either for the first time or on subsequent uses</i>	73
4.7.4	int32_t bHYPRE_SStructParCSRMatrix_GetObject (bHYPRE_SStructParCSRMatrix self, sidl_BaseInterface* A) <i>A semi-structured matrix or vector contains a Struct or IJ matrix or vector</i>	74
4.7.5	int32_t bHYPRE_SStructParCSRMatrix_SetGraph (bHYPRE_SStructParCSRMatrix self, bHYPRE_SStructGraph graph) <i>Set the matrix graph</i>	74
4.7.6	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) <i>Set matrix coefficients index by index</i>	74
4.7.7	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) <i>Set matrix coefficients a box at a time</i>	75
4.7.8	int32_t	

	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)	
	<i>Add to matrix coefficients index by index</i>		75
4.7.9	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)	
	<i>Add to matrix coefficients a box at a time</i>		76
4.7.10	int32_t bHYPRE_SStructParCSRMatrix_SetSymmetric (bHYPRE_SStructParCSRMatrix self, int32_t part, int32_t var, int32_t to-var, int32_t symmetric)	
	<i>Define symmetry properties for the stencil entries in the matrix</i>		76
	int32_t bHYPRE_SStructParCSRMatrix_SetNSSymmetric (bHYPRE_SStructParCSRMatrix self, int32_t symmetric)	
	<i>Define symmetry properties for all non-stencil matrix entries</i>		
	int32_t bHYPRE_SStructParCSRMatrix_SetComplex (bHYPRE_SStructParCSRMatrix self)	
	<i>Set the matrix to be complex</i>		
4.7.11	int32_t bHYPRE_SStructParCSRMatrix_Print (bHYPRE_SStructParCSRMatrix self, const char* filename, int32_t all)		
	<i>Print the matrix to file</i>		77
	int32_t		

```

bHYPRE_SStructParCSRMatrix_SetIntParameter (
    bHYPRE_SStructParCSRMatrix
    self,
    const char* name,
    int32_t value)

    Set the int parameter associated with name

int32_t
bHYPRE_SStructParCSRMatrix_SetDoubleParameter (
    bHYPRE_SStructParCSRMatrix
    self,
    const char* name,
    double value)

    Set the double parameter associated with name

int32_t
bHYPRE_SStructParCSRMatrix_SetStringParameter (
    bHYPRE_SStructParCSRMatrix
    self,
    const char* name,
    const char* value)

    Set the string parameter associated with name

int32_t
bHYPRE_SStructParCSRMatrix_SetIntArray1Parameter (
    bHYPRE_SStructParCSRMatrix
    self, const
    char* name,
    int32_t* value,
    int32_t
    nvalues)

    Set the int 1-D array parameter associated with name

int32_t
bHYPRE_SStructParCSRMatrix_SetIntArray2Parameter (
    bHYPRE_SStructParCSRMatrix
    self, const
    char* name,
    struct
    sidl_int_array*
    value)

    Set the int 2-D array parameter associated with name

int32_t
bHYPRE_SStructParCSRMatrix_SetDoubleArray1Parameter (
    bHYPRE_SStructParCSRMatrix
    self,
    const
    char*
    name,
    double*
    value,
    int32_t
    nvalues)

    Set the double 1-D array parameter associated with name

int32_t

```

bHYPRE_SStructParCSRMatrix_SetDoubleArray2Parameter (
 bHYPRE_SStructParCSRMatrix
 self,
 const
 char*
 name,
 struct
 sidl_double__array*
 value)

Set the double 2-D array parameter associated with name

int32_t
bHYPRE_SStructParCSRMatrix_GetIntValue (
 bHYPRE_SStructParCSRMatrix
 self, const char* name,
 int32_t* value)

Set the int parameter associated with name

int32_t
bHYPRE_SStructParCSRMatrix_GetDoubleValue (
 bHYPRE_SStructParCSRMatrix
 self,
 const char* name,
 double* value)

Get the double parameter associated with name

int32_t
bHYPRE_SStructParCSRMatrix_Setup (bHYPRE_SStructParCSRMatrix
 self, bHYPRE_Vector b,
 bHYPRE_Vector x)
(Optional) Do any preprocessing that may be necessary in order to execute
Apply

int32_t
bHYPRE_SStructParCSRMatrix_Apply (bHYPRE_SStructParCSRMatrix
 self, bHYPRE_Vector b,
 bHYPRE_Vector* x)

Apply the operator to b, returning x

int32_t
bHYPRE_SStructParCSRMatrix_ApplyAdjoint (
 bHYPRE_SStructParCSRMatrix
 self, bHYPRE_Vector b,
 bHYPRE_Vector* x)

Apply the adjoint of the operator to b, returning x

obj
Cast method for interface and class type conversions

void*
bHYPRE_SStructParCSRMatrix__cast2 (void* obj, const char* type)
String cast method for interface and class type conversions

void


```

bHYPRE_SStructParCSRMatrix__exec ( bHYPRE_SStructParCSRMatrix
                                     self,  const char* methodName,
                                     sidl_io_Deserializer inArgs,
                                     sidl_io_Serializer outArgs)

    Select and execute a method by name

char*
bHYPRE_SStructParCSRMatrix__getURL (
                                     bHYPRE_SStructParCSRMatrix
                                     self)

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

```

4.7.1

```

struct bHYPRE_SStructParCSRMatrix__object

```

Symbol "bHYPRE.SStructParCSRMatrix" (version 1.0.0)

The SStructParCSR matrix class.

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

4.7.2

```

int32_t
bHYPRE_SStructParCSRMatrix_SetCommunicator (
bHYPRE_SStructParCSRMatrix self, bHYPRE_MPICommunicator mpi_comm)

```

Set the MPI Communicator. DEPRECATED, Use Create()

4.7.3

```

int32_t
bHYPRE_SStructParCSRMatrix_Assemble (
bHYPRE_SStructParCSRMatrix self)

```

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**.

4.7.4

```
int32_t
bHYPRE_SStructParCSRMatrix_GetObject (
  bHYPRE_SStructParCSRMatrix self, sidl_BaseInterface* A)
```

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

4.7.5

```
int32_t
bHYPRE_SStructParCSRMatrix_SetGraph (
  bHYPRE_SStructParCSRMatrix self, bHYPRE_SStructGraph graph)
```

Set the matrix graph. DEPRECATED Use **Create**

4.7.6

```
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)
```

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.

4.7.7

```
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)
```

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.

4.7.8

```
int32_t
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)
```

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.

4.7.9

```
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)
```

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.

4.7.10

```
int32_t
bHYPRE_SStructParCSRMatrix_SetSymmetric (
  bHYPRE_SStructParCSRMatrix self, int32_t part, int32_t var, int32_t to_var,
  int32_t symmetric)
```

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.

4.7.11

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

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

4.8

SemiStructured ParCSR Vector

Names

4.8.1	struct bHYPRE_SStructParCSRVector__object <i>Symbol "bHYPRE"</i>	80
	void <i>Constructor function for the class</i>	
	bHYPRE_SStructParCSRVector bHYPRE_SStructParCSRVector__createRemote (const char *, sidl_BaseInterface *_ex) <i>RMI constructor function for the class</i>	
	bHYPRE_SStructParCSRVector bHYPRE_SStructParCSRVector__connect (const char *, sidl_BaseInterface *_ex) <i>RMI connector function for the class</i>	
	bHYPRE_SStructParCSRVector bHYPRE_SStructParCSRVector__Create (bHYPRE_MPICommunicator mpi_comm, bHYPRE_SStructGrid grid) <i>Method: Create[]</i>	
4.8.2	int32_t bHYPRE_SStructParCSRVector__SetCommunicator (bHYPRE_SStructParCSRVector self, bHYPRE_MPICommunicator mpi_comm) <i>Set the MPI Communicator</i>	81
	int32_t	

	bHYPRE_SStructParCSRVector_Initialize (bHYPRE_SStructParCSRVector self)	
		<i>Prepare an object for setting coefficient values, whether for the first time or subsequently</i>	
4.8.3	int32_t bHYPRE_SStructParCSRVector_Assemble (bHYPRE_SStructParCSRVector self)	
		<i>Finalize the construction of an object before using, either for the first time or on subsequent uses</i>	81
4.8.4	int32_t bHYPRE_SStructParCSRVector_GetObject (bHYPRE_SStructParCSRVector self, sidl_BaseInterface* A)	
		<i>A semi-structured matrix or vector contains a Struct or IJ matrix or vector</i>	81
	int32_t bHYPRE_SStructParCSRVector_SetGrid (bHYPRE_SStructParCSRVector self, bHYPRE_SStructGrid grid)	
		<i>Set the vector grid</i>	
4.8.5	int32_t bHYPRE_SStructParCSRVector_SetValues (bHYPRE_SStructParCSRVector self, int32_t part, int32_t* index, int32_t dim, int32_t var, double value)	
		<i>Set vector coefficients index by index</i>	82
4.8.6	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)	
		<i>Set vector coefficients a box at a time</i>	82
4.8.7	int32_t bHYPRE_SStructParCSRVector_AddToValues (bHYPRE_SStructParCSRVector self, int32_t part, int32_t* index, int32_t dim, int32_t var, double value)	
		<i>Set vector coefficients index by index</i>	82
4.8.8	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)	
		<i>Set vector coefficients a box at a time</i>		83
	int32_t	bHYPRE_SStructParCSRVector_Gather (bHYPRE_SStructParCSRVector self)	
		<i>Gather vector data before calling GetValues</i>		
4.8.9	int32_t	bHYPRE_SStructParCSRVector_GetValues (bHYPRE_SStructParCSRVector self, int32_t part, int32_t* index, int32_t dim, int32_t var, double* value)	
		<i>Get vector coefficients index by index</i>		83
4.8.10	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)	
		<i>Get vector coefficients a box at a time</i>		84
	int32_t	bHYPRE_SStructParCSRVector_SetComplex (bHYPRE_SStructParCSRVector self)	
		<i>Set the vector to be complex</i>		
4.8.11	int32_t	bHYPRE_SStructParCSRVector_Print (bHYPRE_SStructParCSRVector self, const char* filename, int32_t all)	
		<i>Print the vector to file</i>		84
	int32_t	bHYPRE_SStructParCSRVector_Clear (bHYPRE_SStructParCSRVector self)	
		<i>Set self to 0</i>		
	int32_t	bHYPRE_SStructParCSRVector_Copy (bHYPRE_SStructParCSRVector self, bHYPRE_Vector x)	
		<i>Copy data from x into self</i>		
4.8.12	int32_t			

```

bHYPRE_SStructParCSRVector_Clone ( bHYPRE_SStructParCSRVector
                                     self, bHYPRE_Vector* x)
                                     Create an x compatible with self ..... 84
int32_t
bHYPRE_SStructParCSRVector_Scale ( bHYPRE_SStructParCSRVector
                                     self, double a)
                                     Scale self by a
int32_t
bHYPRE_SStructParCSRVector_Dot ( bHYPRE_SStructParCSRVector self,
                                     bHYPRE_Vector x, double* d)
                                     Compute d, the inner-product of self and x
int32_t
bHYPRE_SStructParCSRVector_Axpy ( bHYPRE_SStructParCSRVector
                                     self, double a,
                                     bHYPRE_Vector x)
                                     Add a*x to self
obj
                                     Cast method for interface and class type conversions
void*
bHYPRE_SStructParCSRVector__cast2 ( void* obj, const char* type)
                                     String cast method for interface and class type conversions
void
bHYPRE_SStructParCSRVector__exec ( bHYPRE_SStructParCSRVector
                                     self, const char* methodName,
                                     sidl_io_Deserializer inArgs,
                                     sidl_io_Serializer outArgs)
                                     Select and execute a method by name
char*
bHYPRE_SStructParCSRVector__getURL (
                                     bHYPRE_SStructParCSRVector
                                     self)
                                     Get the URL of the Implementation of this object (for RMI)

```

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

```
int32_t
bHYPRE_SStructParCSRVector_SetCommunicator (
bHYPRE_SStructParCSRVector self, bHYPRE_MPICommunicator mpi_comm)
```

Set the MPI Communicator. DEPRECATED, Use Create()

4.8.3

```
int32_t
bHYPRE_SStructParCSRVector_Assemble ( bHYPRE_SStructParCSRVector
self)
```

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.

4.8.4

```
int32_t
bHYPRE_SStructParCSRVector_GetObject (
bHYPRE_SStructParCSRVector self, sidl_BaseInterface* A)
```

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

4.8.5

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

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_SetBoxValues (
bHYPRE_SStructParCSRVector self, int32_t part, int32_t* ilower, int32_t* iupper,
int32_t dim, int32_t var, double* values, int32_t nvalues)
```

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

```
int32_t
bHYPRE_SStructParCSRVector_AddToValues (
bHYPRE_SStructParCSRVector self, int32_t part, int32_t* index, int32_t dim,
int32_t var, double value)
```

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.

4.8.8

```
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)
```

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.

4.8.9

```
int32_t
bHYPRE_SStructParCSRVector_GetValues (
  bHYPRE_SStructParCSRVector self, int32_t part, int32_t* index, int32_t dim,
  int32_t var, double* value)
```

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.

4.8.10

```
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)
```

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.

4.8.11

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

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

4.8.12

```
int32_t
bHYPRE_SStructParCSRVector_Clone ( bHYPRE_SStructParCSRVector
  self, bHYPRE_Vector* x)
```

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.

```

5.1      struct bHYPRE_Solver__object
          Symbol "bHYPRE" ..... 86
extern C bHYPRE_Solver
bHYPRE_Solver__connect (const char *, sidl_BaseInterface *_ex)
    RMI connector function for the class

5.2      int32_t
bHYPRE_Solver_SetOperator ( bHYPRE_Solver self,
                           bHYPRE_Operator A)
    Set the operator for the linear system being solved ..... 86

5.3      int32_t
bHYPRE_Solver_SetTolerance ( bHYPRE_Solver self, double tolerance)
    (Optional) Set the convergence tolerance ..... 86

5.4      int32_t
bHYPRE_Solver_SetMaxIterations ( bHYPRE_Solver self,
                                int32_t max_iterations)
    (Optional) Set maximum number of iterations ..... 87

5.5      int32_t
bHYPRE_Solver_SetLogging ( bHYPRE_Solver self, int32_t level)
    (Optional) Set the logging level, specifying the degree of additional informa-
    tional data to be accumulated ..... 87

5.6      int32_t
bHYPRE_Solver_SetPrintLevel ( bHYPRE_Solver 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 ..... 87

int32_t
bHYPRE_Solver_GetNumIterations ( bHYPRE_Solver self,
                                int32_t* num_iterations)
    (Optional) Return the number of iterations taken

int32_t
bHYPRE_Solver_GetRelResidualNorm ( bHYPRE_Solver self,
                                   double* norm)
    (Optional) Return the norm of the relative residual

struct bHYPRE_Solver__object* bHYPRE_Solver__cast void* obj
    Cast method for interface and class type conversions

void*
bHYPRE_Solver__cast2 ( void* obj, const char* type)
    String cast method for interface and class type conversions

void

```

```
bHYPRE_Solver__exec ( bHYPRE_Solver self,  const char* methodName,
                      sidl_io_Deserializer inArgs,  sidl_io_Serializer outArgs)
    Select and execute a method by name
```

```
char*
bHYPRE_Solver__getURL ( bHYPRE_Solver self)
    Get the URL of the Implementation of this object (for RMI)
```

5.7	Identity Solver (does nothing)	87
5.8	Hybrid Solver	93

5.1

```
struct bHYPRE_Solver__object
```

Symbol "bHYPRE.Solver" (version 1.0.0)

5.2

```
int32_t
bHYPRE_Solver_SetOperator ( bHYPRE_Solver self,  bHYPRE_Operator A)
```

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

5.3

```
int32_t
bHYPRE_Solver_SetTolerance ( bHYPRE_Solver self,  double tolerance)
```

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

5.4

```
int32_t
bHYPRE_Solver_SetMaxIterations ( bHYPRE_Solver self, int32_t
max_iterations)
```

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

5.5

```
int32_t bHYPRE_Solver_SetLogging ( bHYPRE_Solver self, int32_t level)
```

(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

5.6

```
int32_t bHYPRE_Solver_SetPrintLevel ( bHYPRE_Solver 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. 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

Identity Solver (does nothing)

Names

5.7.1 struct **bHYPRE_IdentitySolver__object**

	<i>Symbol "bHYPRE"</i>	90
	void <i>Constructor function for the class</i>	
	bHYPRE_IdentitySolver bHYPRE_IdentitySolver__createRemote (const char *, sidl_BaseInterface *_ex) <i>RMI constructor function for the class</i>	
	bHYPRE_IdentitySolver bHYPRE_IdentitySolver__connect (const char *, sidl_BaseInterface *_ex) <i>RMI connector function for the class</i>	
	bHYPRE_IdentitySolver bHYPRE_IdentitySolver_Create (bHYPRE_MPICommunicator mpi_comm) <i>Method: Create[]</i>	
5.7.2	int32_t bHYPRE_IdentitySolver_SetCommunicator (bHYPRE_IdentitySolver self, bHYPRE_MPICommunicator mpi_comm) <i>Set the MPI Communicator</i>	91
	int32_t bHYPRE_IdentitySolver_SetIntParameter (bHYPRE_IdentitySolver self, const char* name, int32_t value) <i>Set the int parameter associated with name</i>	
	int32_t bHYPRE_IdentitySolver_SetDoubleParameter (bHYPRE_IdentitySolver self, const char* name, double value) <i>Set the double parameter associated with name</i>	
	int32_t bHYPRE_IdentitySolver_SetStringParameter (bHYPRE_IdentitySolver self, const char* name, const char* value) <i>Set the string parameter associated with name</i>	
	int32_t bHYPRE_IdentitySolver_SetIntArray1Parameter (bHYPRE_IdentitySolver self, const char* name, int32_t* value, int32_t nvalues) <i>Set the int 1-D array parameter associated with name</i>	
	int32_t bHYPRE_IdentitySolver_SetIntArray2Parameter (bHYPRE_IdentitySolver self, const char* name, struct sidl_int__array* value) <i>Set the int 2-D array parameter associated with name</i>	
	int32_t	

	bHYPRE_IdentitySolver_SetDoubleArray1Parameter (<div style="text-align: right;"> bHYPRE_IdentitySolver self, const char* name, double* value, int32_t nvalues) </div> <i>Set the double 1-D array parameter associated with name</i>	
	int32_t bHYPRE_IdentitySolver_SetDoubleArray2Parameter (<div style="text-align: right;"> bHYPRE_IdentitySolver self, const char* name, struct sidl_double_array* value) </div> <i>Set the double 2-D array parameter associated with name</i>	
	int32_t bHYPRE_IdentitySolver_GetInt Value (bHYPRE_IdentitySolver self, <div style="text-align: right;">const char* name, int32_t* value)</div> <i>Set the int parameter associated with name</i>	
	int32_t bHYPRE_IdentitySolver_GetDoubleValue (bHYPRE_IdentitySolver self, <div style="text-align: right;">const char* name, double* value)</div> <i>Get the double parameter associated with name</i>	
	int32_t bHYPRE_IdentitySolver_Setup (bHYPRE_IdentitySolver self, <div style="text-align: right;">bHYPRE_Vector b, bHYPRE_Vector x)</div> <i>(Optional) Do any preprocessing that may be necessary in order to execute</i> Apply	
	int32_t bHYPRE_IdentitySolver_Apply (bHYPRE_IdentitySolver self, <div style="text-align: right;">bHYPRE_Vector b, bHYPRE_Vector* x)</div> <i>Apply the operator to b, returning x</i>	
	int32_t bHYPRE_IdentitySolver_ApplyAdjoint (bHYPRE_IdentitySolver self, <div style="text-align: right;">bHYPRE_Vector b, bHYPRE_Vector* x)</div> <i>Apply the adjoint of the operator to b, returning x</i>	
5.7.3	int32_t bHYPRE_IdentitySolver_SetOperator (bHYPRE_IdentitySolver self, <div style="text-align: right;">bHYPRE_Operator A)</div> <i>Set the operator for the linear system being solved</i>	91
5.7.4	int32_t bHYPRE_IdentitySolver_SetTolerance (bHYPRE_IdentitySolver self, <div style="text-align: right;">double tolerance)</div> <i>(Optional) Set the convergence tolerance</i>	91
5.7.5	int32_t	

	bHYPRE_IdentitySolver_SetMaxIterations (bHYPRE_IdentitySolver self, int32_t max_iterations)	
	(Optional) Set maximum number of iterations	92
5.7.6	int32_t bHYPRE_IdentitySolver_SetLogging (bHYPRE_IdentitySolver self, int32_t level)	
	(Optional) Set the logging level, specifying the degree of additional informational data to be accumulated	92
5.7.7	int32_t bHYPRE_IdentitySolver_SetPrintLevel (bHYPRE_IdentitySolver 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	92
	int32_t bHYPRE_IdentitySolver_GetNumIterations (bHYPRE_IdentitySolver self, int32_t* num_iterations)	
	(Optional) Return the number of iterations taken	
	int32_t bHYPRE_IdentitySolver_GetRelResidualNorm (bHYPRE_IdentitySolver self, double* norm)	
	(Optional) Return the norm of the relative residual	
	struct bHYPRE_IdentitySolver__object* bHYPRE_IdentitySolver__cast void* obj	
	Cast method for interface and class type conversions	
	void* bHYPRE_IdentitySolver__cast2 (void* obj, const char* type)	
	String cast method for interface and class type conversions	
	void bHYPRE_IdentitySolver__exec (bHYPRE_IdentitySolver self, const char* methodName, sidl_io_Deserializer inArgs, sidl_io_Serializer outArgs)	
	Select and execute a method by name	
	char* bHYPRE_IdentitySolver__getURL (bHYPRE_IdentitySolver self)	
	Get the URL of the Implementation of this object (for RMI)	

5.7.1

```
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.

5.7.2

```
int32_t  
bHYPRE_IdentitySolver_SetCommunicator ( bHYPRE_IdentitySolver self,  
bHYPRE_MPICommunicator mpi_comm)
```

Set the MPI Communicator. DEPRECATED, use Create:

5.7.3

```
int32_t  
bHYPRE_IdentitySolver_SetOperator ( bHYPRE_IdentitySolver self,  
bHYPRE_Operator A)
```

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

5.7.4

```
int32_t  
bHYPRE_IdentitySolver_SetTolerance ( bHYPRE_IdentitySolver self, double  
tolerance)
```

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

5.7.5

```
int32_t
bHYPRE_IdentitySolver_SetMaxIterations ( bHYPRE_IdentitySolver self,
int32_t max_iterations)
```

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

5.7.6

```
int32_t
bHYPRE_IdentitySolver_SetLogging ( bHYPRE_IdentitySolver self, int32_t
level)
```

(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

5.7.7

```
int32_t
bHYPRE_IdentitySolver_SetPrintLevel ( bHYPRE_IdentitySolver 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. 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

Hybrid Solver

Names

5.8.1	struct bHYPRE_Hybrid__object <i>Symbol "bHYPRE"</i> extern C struct bHYPRE_Hybrid__object* bHYPRE_Hybrid__create void <i>Constructor function for the class</i> bHYPRE_Hybrid bHYPRE_Hybrid__createRemote (const char *, sidl_BaseInterface *_ex) <i>RMI constructor function for the class</i> bHYPRE_Hybrid bHYPRE_Hybrid__connect (const char *, sidl_BaseInterface *_ex) <i>RMI connector function for the class</i> bHYPRE_Hybrid bHYPRE_Hybrid__Create (bHYPRE_MPICommunicator mpi_comm, bHYPRE_PreconditionedSolver SecondSolver, bHYPRE_Operator A) <i>Method: Create[]</i> int32_t bHYPRE_Hybrid__GetFirstSolver (bHYPRE_Hybrid self, bHYPRE_PreconditionedSolver* FirstSolver) <i>Method: GetFirstSolver[]</i> int32_t bHYPRE_Hybrid__GetSecondSolver (bHYPRE_Hybrid self, bHYPRE_PreconditionedSolver* SecondSolver) <i>Method: GetSecondSolver[]</i>	95
5.8.2	int32_t bHYPRE_Hybrid__SetCommunicator (bHYPRE_Hybrid self, bHYPRE_MPICommunicator mpi_comm) <i>Set the MPI Communicator</i> int32_t bHYPRE_Hybrid__SetIntParameter (bHYPRE_Hybrid self, const char* name, int32_t value) <i>Set the int parameter associated with name</i> int32_t bHYPRE_Hybrid__SetDoubleParameter (bHYPRE_Hybrid self, const char* name, double value) <i>Set the double parameter associated with name</i> int32_t	96

bHYPRE_Hybrid.SetStringParameter (bHYPRE_Hybrid self,
 const char* name,
 const char* value)
*Set the string parameter associated with **name***

int32_t
bHYPRE_Hybrid.SetIntArray1Parameter (bHYPRE_Hybrid self,
 const char* name,
 int32_t* value, int32_t nvalues)
*Set the int 1-D array parameter associated with **name***

int32_t
bHYPRE_Hybrid.SetIntArray2Parameter (bHYPRE_Hybrid self,
 const char* name,
 struct sidl_int_array* value)
*Set the int 2-D array parameter associated with **name***

int32_t
bHYPRE_Hybrid.SetDoubleArray1Parameter (bHYPRE_Hybrid self,
 const char* name,
 double* value,
 int32_t nvalues)
*Set the double 1-D array parameter associated with **name***

int32_t
bHYPRE_Hybrid.SetDoubleArray2Parameter (bHYPRE_Hybrid self,
 const char* name, struct
 sidl_double_array* value)
*Set the double 2-D array parameter associated with **name***

int32_t
bHYPRE_Hybrid.GetIntValue (bHYPRE_Hybrid self, const char* name,
 int32_t* value)
*Set the int parameter associated with **name***

int32_t
bHYPRE_Hybrid.GetDoubleValue (bHYPRE_Hybrid self,
 const char* name, double* value)
*Get the double parameter associated with **name***

int32_t
bHYPRE_Hybrid.Setup (bHYPRE_Hybrid self, bHYPRE_Vector b,
 bHYPRE_Vector x)
*(Optional) Do any preprocessing that may be necessary in order to execute
 Apply*

int32_t
bHYPRE_Hybrid.Apply (bHYPRE_Hybrid self, bHYPRE_Vector b,
 bHYPRE_Vector* x)
*Apply the operator to **b**, returning **x***

int32_t
bHYPRE_Hybrid.ApplyAdjoint (bHYPRE_Hybrid self,
 bHYPRE_Vector b, bHYPRE_Vector* x)
*Apply the adjoint of the operator to **b**, returning **x***

5.8.3 int32_t

	bHYPRE_Hybrid_SetOperator (bHYPRE_Hybrid self, bHYPRE_Operator A) <i>Set the operator for the linear system being solved</i>	96
5.8.4	int32_t bHYPRE_Hybrid_SetTolerance (bHYPRE_Hybrid self, double tolerance) <i>(Optional) Set the convergence tolerance</i>	96
5.8.5	int32_t bHYPRE_Hybrid_SetMaxIterations (bHYPRE_Hybrid self, int32_t max_iterations) <i>(Optional) Set maximum number of iterations</i>	97
5.8.6	int32_t bHYPRE_Hybrid_SetLogging (bHYPRE_Hybrid self, int32_t level) <i>(Optional) Set the logging level, specifying the degree of additional informational data to be accumulated</i>	97
5.8.7	int32_t bHYPRE_Hybrid_SetPrintLevel (bHYPRE_Hybrid self, int32_t level) <i>(Optional) Set the print level, specifying the degree of informational data to be printed either to the screen or to a file</i>	97
	int32_t bHYPRE_Hybrid_GetNumIterations (bHYPRE_Hybrid self, int32_t* num_iterations) <i>(Optional) Return the number of iterations taken</i>	
	int32_t bHYPRE_Hybrid_GetRelResidualNorm (bHYPRE_Hybrid self, double* norm) <i>(Optional) Return the norm of the relative residual</i>	
	struct bHYPRE_Hybrid__object* bHYPRE_Hybrid__cast void* obj <i>Cast method for interface and class type conversions</i>	
	void* bHYPRE_Hybrid__cast2 (void* obj, const char* type) <i>String cast method for interface and class type conversions</i>	
	void bHYPRE_Hybrid__exec (bHYPRE_Hybrid self, const char* methodName, sidl_io_Deserializer inArgs, sidl_io_Serializer outArgs) <i>Select and execute a method by name</i>	
	char* bHYPRE_Hybrid__getURL (bHYPRE_Hybrid self) <i>Get the URL of the Implementation of this object (for RMI)</i>	

5.8.1

```
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.

5.8.2

```
int32_t
bHYPRE_Hybrid_SetCommunicator ( bHYPRE_Hybrid self,
bHYPRE_MPICommunicator mpi_comm)
```

Set the MPI Communicator. DEPRECATED, use Create:

5.8.3

```
int32_t
bHYPRE_Hybrid_SetOperator ( bHYPRE_Hybrid self, bHYPRE_Operator A)
```

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

5.8.4

```
int32_t
bHYPRE_Hybrid_SetTolerance ( bHYPRE_Hybrid self, double tolerance)
```

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

5.8.5

```
int32_t
bHYPRE_Hybrid_SetMaxIterations ( bHYPRE_Hybrid self, int32_t
max_iterations)
```

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

5.8.6

```
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. Does nothing by default (level = 0). Other levels (if any) are implementation-specific. Must be called before **Setup** and **Apply**. DEPRECATED use SetIntParameter

5.8.7

```
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. Does nothing by default (level=0). Other levels (if any) are implementation-specific. Must be called before **Setup** and **Apply**. DEPRECATED use SetIntParameter

6

ParCSR Matrix Solvers

Names

6.1	ParCSRDiagScale Solver	98
6.2	ParCSR BoomerAMG Solver	104
6.3	ParCSR Euclid Solver	111
6.4	ParCSR Schwarz Solver	115
6.5	ParCSR ParaSails Solver	120
6.6	ParCSR Pilut Solver	125

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

6.1

ParCSRDiagScale Solver

Names

6.1.1	struct bHYPRE_ParCSRDiagScale__object <i>Symbol "bHYPRE"</i>	102
	void <i>Constructor function for the class</i>	
	bHYPRE_ParCSRDiagScale bHYPRE_ParCSRDiagScale__createRemote (const char *, sidl_BaseInterface *_ex) <i>RMI constructor function for the class</i>	
	bHYPRE_ParCSRDiagScale bHYPRE_ParCSRDiagScale__connect (const char *, sidl_BaseInterface *_ex) <i>RMI connector function for the class</i>	
	bHYPRE_ParCSRDiagScale	

```

bHYPRE_ParCSRDiagScale_Create ( bHYPRE_MPICommunicator
                                mpi_comm,
                                bHYPRE_IJParCSRMatrix A)

```

Method: Create[]

6.1.2

int32_t

```

bHYPRE_ParCSRDiagScale_SetCommunicator (
                                bHYPRE_ParCSRDiagScale
                                self,
                                bHYPRE_MPICommunicator
                                mpi_comm)

```

Set the MPI Communicator 102

int32_t

```

bHYPRE_ParCSRDiagScale_SetIntParameter (
                                bHYPRE_ParCSRDiagScale
                                self, const char* name,
                                int32_t value)

```

Set the int parameter associated with name

int32_t

```

bHYPRE_ParCSRDiagScale_SetDoubleParameter (
                                bHYPRE_ParCSRDiagScale
                                self,
                                const char* name,
                                double value)

```

Set the double parameter associated with name

int32_t

```

bHYPRE_ParCSRDiagScale_SetStringParameter (
                                bHYPRE_ParCSRDiagScale
                                self, const char* name,
                                const char* value)

```

Set the string parameter associated with name

int32_t

```

bHYPRE_ParCSRDiagScale_SetIntArray1Parameter (
                                bHYPRE_ParCSRDiagScale
                                self,
                                const char* name,
                                int32_t* value,
                                int32_t nvalues)

```

Set the int 1-D array parameter associated with name

int32_t

```

bHYPRE_ParCSRDiagScale_SetIntArray2Parameter (
                                bHYPRE_ParCSRDiagScale
                                self,
                                const char* name,
                                struct
                                sidl_int__array*
                                value)

```

Set the int 2-D array parameter associated with name

int32_t

bHYPRE_ParCSRDiagScale_SetDoubleArray1Parameter (
 bHYPRE_ParCSRDiagScale
 self, const
 char* name,
 double* value,
 int32_t nvalues)

Set the double 1-D array parameter associated with name

int32_t

bHYPRE_ParCSRDiagScale_SetDoubleArray2Parameter (
 bHYPRE_ParCSRDiagScale
 self, const
 char* name,
 struct
 sidl_double__array*
 value)

Set the double 2-D array parameter associated with name

int32_t

bHYPRE_ParCSRDiagScale_GetIntValue (bHYPRE_ParCSRDiagScale
 self, const char* name,
 int32_t* value)

Set the int parameter associated with name

int32_t

bHYPRE_ParCSRDiagScale_GetDoubleValue (
 bHYPRE_ParCSRDiagScale
 self, const char* name,
 double* value)

Get the double parameter associated with name

int32_t

bHYPRE_ParCSRDiagScale_Setup (bHYPRE_ParCSRDiagScale self,
 bHYPRE_Vector b,
 bHYPRE_Vector x)

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

int32_t

bHYPRE_ParCSRDiagScale_Apply (bHYPRE_ParCSRDiagScale self,
 bHYPRE_Vector b,
 bHYPRE_Vector* x)

Apply the operator to b, returning x

int32_t

bHYPRE_ParCSRDiagScale_ApplyAdjoint (bHYPRE_ParCSRDiagScale
 self, bHYPRE_Vector b,
 bHYPRE_Vector* x)

Apply the adjoint of the operator to b, returning x

6.1.3

int32_t

bHYPRE_ParCSRDiagScale_SetOperator (bHYPRE_ParCSRDiagScale
 self, bHYPRE_Operator A)

Set the operator for the linear system being solved

102

6.1.4

int32_t

	bHYPRE_ParCSRDiagScale_SetTolerance (bHYPRE_ParCSRDiagScale self, double tolerance)	
	<i>(Optional) Set the convergence tolerance</i>	103
6.1.5	int32_t bHYPRE_ParCSRDiagScale_SetMaxIterations (bHYPRE_ParCSRDiagScale self, int32_t max_iterations)	
	<i>(Optional) Set maximum number of iterations</i>	103
6.1.6	int32_t bHYPRE_ParCSRDiagScale_SetLogging (bHYPRE_ParCSRDiagScale self, int32_t level)	
	<i>(Optional) Set the logging level, specifying the degree of additional informa- tional data to be accumulated</i>	103
6.1.7	int32_t bHYPRE_ParCSRDiagScale_SetPrintLevel (bHYPRE_ParCSRDiagScale self, int32_t level)	
	<i>(Optional) Set the print level, specifying the degree of informational data to be printed either to the screen or to a file</i>	103
	int32_t bHYPRE_ParCSRDiagScale_GetNumIterations (bHYPRE_ParCSRDiagScale self, int32_t* num_iterations)	
	<i>(Optional) Return the number of iterations taken</i>	
	int32_t bHYPRE_ParCSRDiagScale_GetRelResidualNorm (bHYPRE_ParCSRDiagScale self, double* norm)	
	<i>(Optional) Return the norm of the relative residual</i>	
	obj <i>Cast method for interface and class type conversions</i>	
	void* bHYPRE_ParCSRDiagScale__cast2 (void* obj, const char* type)	
	<i>String cast method for interface and class type conversions</i>	
	void bHYPRE_ParCSRDiagScale__exec (bHYPRE_ParCSRDiagScale self, const char* methodName, sidl_io_Deserializer inArgs, sidl_io_Serializer outArgs)	
	<i>Select and execute a method by name</i>	
	char* bHYPRE_ParCSRDiagScale__getURL (bHYPRE_ParCSRDiagScale self)	
	<i>Get the URL of the Implementation of this object (for RMI)</i>	

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

```
int32_t  
bHYPRE_ParCSRDiagScale_SetCommunicator (  
  bHYPRE_ParCSRDiagScale self, bHYPRE_MPICommunicator mpi_comm)
```

Set the MPI Communicator. DEPRECATED, use Create:

6.1.3

```
int32_t  
bHYPRE_ParCSRDiagScale_SetOperator ( bHYPRE_ParCSRDiagScale self,  
  bHYPRE_Operator A)
```

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

6.1.4

```
int32_t
bHYPRE_ParCSRDiagScale_SetTolerance ( bHYPRE_ParCSRDiagScale self,
double tolerance)
```

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

6.1.5

```
int32_t
bHYPRE_ParCSRDiagScale_SetMaxIterations (
bHYPRE_ParCSRDiagScale self, int32_t max_iterations)
```

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

6.1.6

```
int32_t
bHYPRE_ParCSRDiagScale_SetLogging ( bHYPRE_ParCSRDiagScale self,
int32_t level)
```

(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

```
int32_t
bHYPRE_ParCSRDiagScale_SetPrintLevel ( bHYPRE_ParCSRDiagScale
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. 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

ParCSR BoomerAMG Solver

Names

6.2.1	struct bHYPRE_BoomerAMG__object <i>Symbol "bHYPRE"</i>	107
	void <i>Constructor function for the class</i> bHYPRE_BoomerAMG bHYPRE_BoomerAMG__createRemote (const char *, sidl_BaseInterface *_ex) <i>RMI constructor function for the class</i> bHYPRE_BoomerAMG bHYPRE_BoomerAMG__connect (const char *, sidl_BaseInterface *_ex) <i>RMI connector function for the class</i> bHYPRE_BoomerAMG bHYPRE_BoomerAMG_Create (bHYPRE_MPICommunicator mpi_comm, bHYPRE_IJParCSRMatrix A) <i>Method: Create[]</i> int32_t bHYPRE_BoomerAMG_SetLevelRelaxWt (bHYPRE_BoomerAMG self, double relax_wt, int32_t level) <i>Method: SetLevelRelaxWt[]</i> int32_t bHYPRE_BoomerAMG_InitGridRelaxation (bHYPRE_BoomerAMG self, struct sidl_int__array** num_grid_sweeps, struct sidl_int__array** grid_relax_type, struct sidl_int__array** grid_relax_points, int32_t coarsen_type, struct sidl_double__array** relax_weights, int32_t max_levels) <i>Method: InitGridRelaxation[]</i>	
6.2.2	int32_t	

bHYPRE_BoomerAMG_SetCommunicator (bHYPRE_BoomerAMG self, bHYPRE_MPICommunicator mpi_comm)	
<i>Set the MPI Communicator</i>	109
int32_t		
bHYPRE_BoomerAMG_SetIntParameter (bHYPRE_BoomerAMG self, const char* name, int32_t value)	
<i>Set the int parameter associated with name</i>		
int32_t		
bHYPRE_BoomerAMG_SetDoubleParameter (bHYPRE_BoomerAMG self, const char* name, double value)	
<i>Set the double parameter associated with name</i>		
int32_t		
bHYPRE_BoomerAMG_SetStringParameter (bHYPRE_BoomerAMG self, const char* name, const char* value)	
<i>Set the string parameter associated with name</i>		
int32_t		
bHYPRE_BoomerAMG_SetIntArray1Parameter (bHYPRE_BoomerAMG self, const char* name, int32_t* value, int32_t nvalues)	
<i>Set the int 1-D array parameter associated with name</i>		
int32_t		
bHYPRE_BoomerAMG_SetIntArray2Parameter (bHYPRE_BoomerAMG self, const char* name, struct sidl_int__array* value)	
<i>Set the int 2-D array parameter associated with name</i>		
int32_t		
bHYPRE_BoomerAMG_SetDoubleArray1Parameter (bHYPRE_BoomerAMG self, const char* name, double* value, int32_t nvalues)	
<i>Set the double 1-D array parameter associated with name</i>		
int32_t		
bHYPRE_BoomerAMG_SetDoubleArray2Parameter (bHYPRE_BoomerAMG self, const char* name, struct sidl_double__array* value)	
<i>Set the double 2-D array parameter associated with name</i>		
int32_t		

	bHYPRE_BoomerAMG_GetIntValue (bHYPRE_BoomerAMG self, const char* name, int32_t* value) <i>Set the int parameter associated with name</i>	
	int32_t bHYPRE_BoomerAMG_GetDoubleValue (bHYPRE_BoomerAMG self, const char* name, double* value) <i>Get the double parameter associated with name</i>	
	int32_t bHYPRE_BoomerAMG_Setup (bHYPRE_BoomerAMG self, bHYPRE_Vector b, bHYPRE_Vector x) <i>(Optional) Do any preprocessing that may be necessary in order to execute Apply</i>	
	int32_t bHYPRE_BoomerAMG_Apply (bHYPRE_BoomerAMG self, bHYPRE_Vector b, bHYPRE_Vector* x) <i>Apply the operator to b, returning x</i>	
	int32_t bHYPRE_BoomerAMG_ApplyAdjoint (bHYPRE_BoomerAMG self, bHYPRE_Vector b, bHYPRE_Vector* x) <i>Apply the adjoint of the operator to b, returning x</i>	
6.2.3	int32_t bHYPRE_BoomerAMG_SetOperator (bHYPRE_BoomerAMG self, bHYPRE_Operator A) <i>Set the operator for the linear system being solved</i>	109
6.2.4	int32_t bHYPRE_BoomerAMG_SetTolerance (bHYPRE_BoomerAMG self, double tolerance) <i>(Optional) Set the convergence tolerance</i>	109
6.2.5	int32_t bHYPRE_BoomerAMG_SetMaxIterations (bHYPRE_BoomerAMG self, int32_t max_iterations) <i>(Optional) Set maximum number of iterations</i>	110
6.2.6	int32_t bHYPRE_BoomerAMG_SetLogging (bHYPRE_BoomerAMG self, int32_t level) <i>(Optional) Set the logging level, specifying the degree of additional informational data to be accumulated</i>	110
6.2.7	int32_t bHYPRE_BoomerAMG_SetPrintLevel (bHYPRE_BoomerAMG self, int32_t level) <i>(Optional) Set the print level, specifying the degree of informational data to be printed either to the screen or to a file</i>	110
	int32_t bHYPRE_BoomerAMG_GetNumIterations (bHYPRE_BoomerAMG self, int32_t* num_iterations) <i>(Optional) Return the number of iterations taken</i>	
	int32_t	

```

bHYPRE_BoomerAMG_GetRelResidualNorm ( bHYPRE_BoomerAMG
                                         self, double* norm)
    (Optional) Return the norm of the relative residual

struct bHYPRE_BoomerAMG__object* bHYPRE_BoomerAMG__cast void* obj
    Cast method for interface and class type conversions

void*
bHYPRE_BoomerAMG__cast2 ( void* obj, const char* type)
    String cast method for interface and class type conversions

void
bHYPRE_BoomerAMG__exec ( bHYPRE_BoomerAMG self,
                          const char* methodName,
                          sidl_io_Deserializer inArgs,
                          sidl_io_Serializer outArgs)
    Select and execute a method by name

char*
bHYPRE_BoomerAMG__getURL ( bHYPRE_BoomerAMG self)
    Get the URL of the Implementation of this object (for RMI)

```

6.2.1

```

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. DEPRECATED: 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

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.

Cycle0RelaxType (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. DEPRECATED: 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

```
int32_t
bHYPRE_BoomerAMG_SetCommunicator ( bHYPRE_BoomerAMG self,
bHYPRE_MPICommunicator mpi_comm)
```

Set the MPI Communicator. DEPRECATED, use Create:

6.2.3

```
int32_t
bHYPRE_BoomerAMG_SetOperator ( bHYPRE_BoomerAMG self,
bHYPRE_Operator A)
```

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

6.2.4

```
int32_t
bHYPRE_BoomerAMG_SetTolerance ( bHYPRE_BoomerAMG self, double
tolerance)
```

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

6.2.5

```
int32_t
bHYPRE_BoomerAMG_SetMaxIterations ( bHYPRE_BoomerAMG self,
int32_t max_iterations)
```

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

6.2.6

```
int32_t
bHYPRE_BoomerAMG_SetLogging ( bHYPRE_BoomerAMG self, int32_t
level)
```

(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

```
int32_t
bHYPRE_BoomerAMG_SetPrintLevel ( bHYPRE_BoomerAMG 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. 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

ParCSR Euclid Solver

Names

6.3.1	struct bHYPRE_Euclid__object <i>Symbol "bHYPRE"</i> extern C struct bHYPRE_Euclid__object* bHYPRE_Euclid__create void <i>Constructor function for the class</i> bHYPRE_Euclid bHYPRE_Euclid__createRemote (const char *, sidl_BaseInterface *_ex) <i>RMI constructor function for the class</i> bHYPRE_Euclid bHYPRE_Euclid__connect (const char *, sidl_BaseInterface *_ex) <i>RMI connector function for the class</i> bHYPRE_Euclid bHYPRE_Euclid_Create (bHYPRE_MPICommunicator mpi_comm, bHYPRE_IJParCSRMatrix A) <i>Method: Create[]</i> int32_t bHYPRE_Euclid_SetParameters (bHYPRE_Euclid self, int32_t argc, char** argv) <i>Method: SetParameters[]</i>	113
6.3.2	int32_t bHYPRE_Euclid_SetCommunicator (bHYPRE_Euclid self, bHYPRE_MPICommunicator mpi_comm) <i>Set the MPI Communicator</i> int32_t bHYPRE_Euclid_SetIntParameter (bHYPRE_Euclid self, const char* name, int32_t value) <i>Set the int parameter associated with name</i> int32_t bHYPRE_Euclid_SetDoubleParameter (bHYPRE_Euclid self, const char* name, double value) <i>Set the double parameter associated with name</i> int32_t bHYPRE_Euclid_SetStringParameter (bHYPRE_Euclid self, const char* name, const char* value) <i>Set the string parameter associated with name</i> int32_t bHYPRE_Euclid_SetIntArray1Parameter (bHYPRE_Euclid self, const char* name, int32_t* value, int32_t nvalues) <i>Set the int 1-D array parameter associated with name</i> int32_t	114

	bHYPRE_Euclid_SetIntArray2Parameter (bHYPRE_Euclid self, const char* name, struct sidl_int__array* value) <i>Set the int 2-D array parameter associated with name</i>	
	int32_t bHYPRE_Euclid_SetDoubleArray1Parameter (bHYPRE_Euclid self, const char* name, double* value, int32_t nvalues) <i>Set the double 1-D array parameter associated with name</i>	
	int32_t bHYPRE_Euclid_SetDoubleArray2Parameter (bHYPRE_Euclid self, const char* name, struct sidl_double__array* value) <i>Set the double 2-D array parameter associated with name</i>	
	int32_t bHYPRE_Euclid_GetIntValue (bHYPRE_Euclid self, const char* name, int32_t* value) <i>Set the int parameter associated with name</i>	
	int32_t bHYPRE_Euclid_GetDoubleValue (bHYPRE_Euclid self, const char* name, double* value) <i>Get the double parameter associated with name</i>	
	int32_t bHYPRE_Euclid_Setup (bHYPRE_Euclid self, bHYPRE_Vector b, bHYPRE_Vector x) <i>(Optional) Do any preprocessing that may be necessary in order to execute</i> Apply	
	int32_t bHYPRE_Euclid_Apply (bHYPRE_Euclid self, bHYPRE_Vector b, bHYPRE_Vector* x) <i>Apply the operator to b, returning x</i>	
	int32_t bHYPRE_Euclid_ApplyAdjoint (bHYPRE_Euclid self, bHYPRE_Vector b, bHYPRE_Vector* x) <i>Apply the adjoint of the operator to b, returning x</i>	
6.3.3	int32_t bHYPRE_Euclid_SetOperator (bHYPRE_Euclid self, bHYPRE_Operator A) <i>Set the operator for the linear system being solved</i>	114
6.3.4	int32_t bHYPRE_Euclid_SetTolerance (bHYPRE_Euclid self, double tolerance) <i>(Optional) Set the convergence tolerance</i>	114
6.3.5	int32_t bHYPRE_Euclid_SetMaxIterations (bHYPRE_Euclid self, int32_t max.iterations) <i>(Optional) Set maximum number of iterations</i>	114
6.3.6	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	115
6.3.7	int32_t bHYPRE_Euclid_SetPrintLevel (bHYPRE_Euclid 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	115
	int32_t bHYPRE_Euclid_GetNumIterations (bHYPRE_Euclid self, int32_t* num_iterations) (Optional) Return the number of iterations taken	
	int32_t bHYPRE_Euclid_GetRelResidualNorm (bHYPRE_Euclid self, double* norm) (Optional) Return the norm of the relative residual	
	struct bHYPRE_Euclid__object* bHYPRE_Euclid__cast void* obj Cast method for interface and class type conversions	
	void* bHYPRE_Euclid__cast2 (void* obj, const char* type) String cast method for interface and class type conversions	
	void bHYPRE_Euclid__exec (bHYPRE_Euclid self, const char* methodName, sidl_io_Deserializer inArgs, sidl_io_Serializer outArgs) Select and execute a method by name	
	char* bHYPRE_Euclid__getURL (bHYPRE_Euclid self) Get the URL of the Implementation of this object (for RMI)	

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 Set*Parameter functions are available, a Euclid-type parameter-setting function is also available, SetParameters.

6.3.2

```
int32_t
bHYPRE_Euclid_SetCommunicator ( bHYPRE_Euclid self,
bHYPRE_MPICommunicator mpi_comm)
```

Set the MPI Communicator. DEPRECATED, use Create:

6.3.3

```
int32_t
bHYPRE_Euclid_SetOperator ( bHYPRE_Euclid self, bHYPRE_Operator A)
```

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

6.3.4

```
int32_t
bHYPRE_Euclid_SetTolerance ( bHYPRE_Euclid self, double tolerance)
```

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

6.3.5

```
int32_t
bHYPRE_Euclid_SetMaxIterations ( bHYPRE_Euclid self, int32_t
max_iterations)
```

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

6.3.6

```
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. 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

```
int32_t bHYPRE_Euclid_SetPrintLevel ( bHYPRE_Euclid 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. 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**ParCSR Schwarz Solver****Names**

```
6.4.1      struct bHYPRE_Schwarz__object
              Symbol "bHYPRE" ..... 118

extern C struct bHYPRE_Schwarz__object* bHYPRE_Schwarz__create void
              Constructor function for the class

bHYPRE_Schwarz
bHYPRE_Schwarz__createRemote (const char *, sidl_BaseInterface *_ex)
              RMI constructor function for the class

bHYPRE_Schwarz
bHYPRE_Schwarz__connect (const char *, sidl_BaseInterface *_ex)
              RMI connector function for the class

bHYPRE_Schwarz
```

	bHYPRE_Schwarz_Create (bHYPRE_IJParCSRMatrix A) <i>Method: Create[]</i>	
6.4.2	int32_t bHYPRE_Schwarz_SetCommunicator (bHYPRE_Schwarz self, bHYPRE_MPICommunicator mpi_comm) <i>Set the MPI Communicator</i>	118
	int32_t bHYPRE_Schwarz_SetIntParameter (bHYPRE_Schwarz self, const char* name, int32_t value) <i>Set the int parameter associated with name</i>	
	int32_t bHYPRE_Schwarz_SetDoubleParameter (bHYPRE_Schwarz self, const char* name, double value) <i>Set the double parameter associated with name</i>	
	int32_t bHYPRE_Schwarz_SetStringParameter (bHYPRE_Schwarz self, const char* name, const char* value) <i>Set the string parameter associated with name</i>	
	int32_t bHYPRE_Schwarz_SetIntArray1Parameter (bHYPRE_Schwarz self, const char* name, int32_t* value, int32_t nvalues) <i>Set the int 1-D array parameter associated with name</i>	
	int32_t bHYPRE_Schwarz_SetIntArray2Parameter (bHYPRE_Schwarz self, const char* name, struct sidl_int_array* value) <i>Set the int 2-D array parameter associated with name</i>	
	int32_t bHYPRE_Schwarz_SetDoubleArray1Parameter (bHYPRE_Schwarz self, const char* name, double* value, int32_t nvalues) <i>Set the double 1-D array parameter associated with name</i>	
	int32_t bHYPRE_Schwarz_SetDoubleArray2Parameter (bHYPRE_Schwarz self, const char* name, struct sidl_double_array* value) <i>Set the double 2-D array parameter associated with name</i>	
	int32_t bHYPRE_Schwarz_GetIntValue (bHYPRE_Schwarz self, const char* name, int32_t* value) <i>Set the int parameter associated with name</i>	
	int32_t	

	bHYPRE_Schwarz_GetDoubleValue (bHYPRE_Schwarz self, const char* name, double* value) <i>Get the double parameter associated with name</i>	
	int32_t bHYPRE_Schwarz_Setup (bHYPRE_Schwarz self, bHYPRE_Vector b, bHYPRE_Vector x) <i>(Optional) Do any preprocessing that may be necessary in order to execute Apply</i>	
	int32_t bHYPRE_Schwarz_Apply (bHYPRE_Schwarz self, bHYPRE_Vector b, bHYPRE_Vector* x) <i>Apply the operator to b, returning x</i>	
	int32_t bHYPRE_Schwarz_ApplyAdjoint (bHYPRE_Schwarz self, bHYPRE_Vector b, bHYPRE_Vector* x) <i>Apply the adjoint of the operator to b, returning x</i>	
6.4.3	int32_t bHYPRE_Schwarz_SetOperator (bHYPRE_Schwarz self, bHYPRE_Operator A) <i>Set the operator for the linear system being solved</i>	119
6.4.4	int32_t bHYPRE_Schwarz_SetTolerance (bHYPRE_Schwarz self, double tolerance) <i>(Optional) Set the convergence tolerance</i>	119
6.4.5	int32_t bHYPRE_Schwarz_SetMaxIterations (bHYPRE_Schwarz self, int32_t max_iterations) <i>(Optional) Set maximum number of iterations</i>	119
6.4.6	int32_t bHYPRE_Schwarz_SetLogging (bHYPRE_Schwarz self, int32_t level) <i>(Optional) Set the logging level, specifying the degree of additional informa- tional data to be accumulated</i>	119
6.4.7	int32_t bHYPRE_Schwarz_SetPrintLevel (bHYPRE_Schwarz self, int32_t level) <i>(Optional) Set the print level, specifying the degree of informational data to be printed either to the screen or to a file</i>	120
	int32_t bHYPRE_Schwarz_GetNumIterations (bHYPRE_Schwarz self, int32_t* num_iterations) <i>(Optional) Return the number of iterations taken</i>	
	int32_t bHYPRE_Schwarz_GetRelResidualNorm (bHYPRE_Schwarz self, double* norm) <i>(Optional) Return the norm of the relative residual</i>	
	struct bHYPRE_Schwarz__object* bHYPRE_Schwarz__cast void* obj <i>Cast method for interface and class type conversions</i>	
	void*	

```

bHYPRE_Schwarz__cast2 ( void* obj, const char* type)
    String cast method for interface and class type conversions

void
bHYPRE_Schwarz__exec ( bHYPRE_Schwarz self,  const char* methodName,
                        sidl_io_Deserializer inArgs,
                        sidl_io_Serializer outArgs)
    Select and execute a method by name

char*
bHYPRE_Schwarz__getURL ( bHYPRE_Schwarz self)
    Get the URL of the Implementation of this object (for RMI)

```

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

```

int32_t
bHYPRE_Schwarz_SetCommunicator ( bHYPRE_Schwarz self,
bHYPRE_MPICommunicator mpi_comm)

```

Set the MPI Communicator. DEPRECATED, use Create:

6.4.3

```
int32_t
bHYPRE_Schwarz_SetOperator ( bHYPRE_Schwarz self, bHYPRE_Operator
A)
```

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

6.4.4

```
int32_t
bHYPRE_Schwarz_SetTolerance ( bHYPRE_Schwarz self, double tolerance)
```

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

6.4.5

```
int32_t
bHYPRE_Schwarz_SetMaxIterations ( bHYPRE_Schwarz self, int32_t
max_iterations)
```

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

6.4.6

```
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. 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

```
int32_t
bHYPRE_Schwarz_SetPrintLevel ( bHYPRE_Schwarz 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. 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

ParCSR ParaSails Solver

Names

```
6.5.1      struct bHYPRE_ParaSails__object
              Symbol "bHYPRE" ..... 123

extern C struct bHYPRE_ParaSails__object* bHYPRE_ParaSails__create void
              Constructor function for the class

bHYPRE_ParaSails
bHYPRE_ParaSails__createRemote (const char *, sidl_BaseInterface *_ex)
              RMI constructor function for the class

bHYPRE_ParaSails
bHYPRE_ParaSails__connect (const char *, sidl_BaseInterface *_ex)
              RMI connector function for the class

bHYPRE_ParaSails
bHYPRE_ParaSails_Create ( bHYPRE_MPICommunicator mpi_comm,
                          bHYPRE_IJParCSRMatrix A)
              Method: Create[]

6.5.2      int32_t
bHYPRE_ParaSails_SetCommunicator ( bHYPRE_ParaSails self,
                                   bHYPRE_MPICommunicator
                                   mpi_comm)
              Set the MPI Communicator ..... 123

int32_t
bHYPRE_ParaSails_SetIntParameter ( bHYPRE_ParaSails self,
                                   const char* name, int32_t value)
              Set the int parameter associated with name

int32_t
```


bHYPRE_ParaSails_SetDoubleParameter (bHYPRE_ParaSails self,
const char* name, double value)

*Set the double parameter associated with **name***

int32_t

bHYPRE_ParaSails_SetStringParameter (bHYPRE_ParaSails self,
const char* name,
const char* value)

*Set the string parameter associated with **name***

int32_t

bHYPRE_ParaSails_SetIntArray1Parameter (bHYPRE_ParaSails self,
const char* name,
int32_t* value,
int32_t nvalues)

*Set the int 1-D array parameter associated with **name***

int32_t

bHYPRE_ParaSails_SetIntArray2Parameter (bHYPRE_ParaSails self,
const char* name,
struct sidl_int_array* value)

*Set the int 2-D array parameter associated with **name***

int32_t

bHYPRE_ParaSails_SetDoubleArray1Parameter (bHYPRE_ParaSails self,
const char* name,
double* value,
int32_t nvalues)

*Set the double 1-D array parameter associated with **name***

int32_t

bHYPRE_ParaSails_SetDoubleArray2Parameter (bHYPRE_ParaSails self,
const char* name,
struct sidl_double_array*
value)

*Set the double 2-D array parameter associated with **name***

int32_t

bHYPRE_ParaSails_GetIntValue (bHYPRE_ParaSails self,
const char* name, int32_t* value)

*Set the int parameter associated with **name***

int32_t

bHYPRE_ParaSails_GetDoubleValue (bHYPRE_ParaSails self,
const char* name, double* value)

*Get the double parameter associated with **name***

int32_t

bHYPRE_ParaSails_Setup (bHYPRE_ParaSails self, bHYPRE_Vector b,
bHYPRE_Vector x)

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

int32_t

bHYPRE_ParaSails_Apply (bHYPRE_ParaSails self, bHYPRE_Vector b,
bHYPRE_Vector* x)

*Apply the operator to **b**, returning **x***

int32_t

	bHYPRE_ParaSails_ApplyAdjoint (bHYPRE_ParaSails self, bHYPRE_Vector b, bHYPRE_Vector* x) <i>Apply the adjoint of the operator to b, returning x</i>	
6.5.3	int32_t bHYPRE_ParaSails_SetOperator (bHYPRE_ParaSails self, bHYPRE_Operator A) <i>Set the operator for the linear system being solved</i>	123
6.5.4	int32_t bHYPRE_ParaSails_SetTolerance (bHYPRE_ParaSails self, double tolerance) <i>(Optional) Set the convergence tolerance</i>	124
6.5.5	int32_t bHYPRE_ParaSails_SetMaxIterations (bHYPRE_ParaSails self, int32_t max_iterations) <i>(Optional) Set maximum number of iterations</i>	124
6.5.6	int32_t bHYPRE_ParaSails_SetLogging (bHYPRE_ParaSails self, int32_t level) <i>(Optional) Set the logging level, specifying the degree of additional informational data to be accumulated</i>	124
6.5.7	int32_t bHYPRE_ParaSails_SetPrintLevel (bHYPRE_ParaSails self, int32_t level) <i>(Optional) Set the print level, specifying the degree of informational data to be printed either to the screen or to a file</i>	124
	int32_t bHYPRE_ParaSails_GetNumIterations (bHYPRE_ParaSails self, int32_t* num_iterations) <i>(Optional) Return the number of iterations taken</i>	
	int32_t bHYPRE_ParaSails_GetRelResidualNorm (bHYPRE_ParaSails self, double* norm) <i>(Optional) Return the norm of the relative residual</i>	
	struct bHYPRE_ParaSails__object* bHYPRE_ParaSails__cast void* obj <i>Cast method for interface and class type conversions</i>	
	void* bHYPRE_ParaSails__cast2 (void* obj, const char* type) <i>String cast method for interface and class type conversions</i>	
	void bHYPRE_ParaSails__exec (bHYPRE_ParaSails self, const char* methodName, sidl_io_Deserializer inArgs, sidl_io_Serializer outArgs) <i>Select and execute a method by name</i>	
	char* bHYPRE_ParaSails__getURL (bHYPRE_ParaSails self) <i>Get the URL of the Implementation of this object (for RMI)</i>	

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

```
int32_t
bHYPRE_ParaSails_SetCommunicator ( bHYPRE_ParaSails self,
bHYPRE_MPICommunicator mpi_comm)
```

Set the MPI Communicator. DEPRECATED, use Create:

6.5.3

```
int32_t
bHYPRE_ParaSails_SetOperator ( bHYPRE_ParaSails self,
bHYPRE_Operator A)
```

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

6.5.4

```
int32_t
bHYPRE_ParaSails_SetTolerance ( bHYPRE_ParaSails self, double tolerance)
```

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

6.5.5

```
int32_t
bHYPRE_ParaSails_SetMaxIterations ( bHYPRE_ParaSails self, int32_t
max_iterations)
```

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

6.5.6

```
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. 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

```
int32_t
bHYPRE_ParaSails_SetPrintLevel ( bHYPRE_ParaSails 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. 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

ParCSR Pilut Solver

Names

6.6.1	struct bHYPRE_Pilut__object <i>Symbol "bHYPRE"</i> extern C struct bHYPRE_Pilut__object* bHYPRE_Pilut__create void <i>Constructor function for the class</i> bHYPRE_Pilut bHYPRE_Pilut__createRemote (const char *, sidl_BaseInterface *_ex) <i>RMI constructor function for the class</i> bHYPRE_Pilut bHYPRE_Pilut__connect (const char *, sidl_BaseInterface *_ex) <i>RMI connector function for the class</i> bHYPRE_Pilut bHYPRE_Pilut_Create (bHYPRE_MPICommunicator mpi_comm, bHYPRE_Operator A) <i>Method: Create[]</i>	127
6.6.2	int32_t bHYPRE_Pilut_SetCommunicator (bHYPRE_Pilut self, bHYPRE_MPICommunicator mpi_comm) <i>Set the MPI Communicator</i> int32_t bHYPRE_Pilut_SetIntParameter (bHYPRE_Pilut self, const char* name, int32_t value) <i>Set the int parameter associated with name</i> int32_t bHYPRE_Pilut_SetDoubleParameter (bHYPRE_Pilut self, const char* name, double value) <i>Set the double parameter associated with name</i> int32_t bHYPRE_Pilut_SetStringParameter (bHYPRE_Pilut self, const char* name, const char* value) <i>Set the string parameter associated with name</i> int32_t	128

	bHYPRE_Pilut_SetIntArray1Parameter (bHYPRE_Pilut self, const char* name, int32_t* value, int32_t nvalues) <i>Set the int 1-D array parameter associated with name</i>	
	int32_t bHYPRE_Pilut_SetIntArray2Parameter (bHYPRE_Pilut self, const char* name, struct sidl_int__array* value) <i>Set the int 2-D array parameter associated with name</i>	
	int32_t bHYPRE_Pilut_SetDoubleArray1Parameter (bHYPRE_Pilut self, const char* name, double* value, int32_t nvalues) <i>Set the double 1-D array parameter associated with name</i>	
	int32_t bHYPRE_Pilut_SetDoubleArray2Parameter (bHYPRE_Pilut self, const char* name, struct sidl_double__array* value) <i>Set the double 2-D array parameter associated with name</i>	
	int32_t bHYPRE_Pilut_GetIntValue (bHYPRE_Pilut self, const char* name, int32_t* value) <i>Set the int parameter associated with name</i>	
	int32_t bHYPRE_Pilut_GetDoubleValue (bHYPRE_Pilut self, const char* name, double* value) <i>Get the double parameter associated with name</i>	
	int32_t bHYPRE_Pilut_Setup (bHYPRE_Pilut self, bHYPRE_Vector b, bHYPRE_Vector x) <i>(Optional) Do any preprocessing that may be necessary in order to execute</i> Apply	
	int32_t bHYPRE_Pilut_Apply (bHYPRE_Pilut self, bHYPRE_Vector b, bHYPRE_Vector* x) <i>Apply the operator to b, returning x</i>	
	int32_t bHYPRE_Pilut_ApplyAdjoint (bHYPRE_Pilut self, bHYPRE_Vector b, bHYPRE_Vector* x) <i>Apply the adjoint of the operator to b, returning x</i>	
6.6.3	int32_t bHYPRE_Pilut_SetOperator (bHYPRE_Pilut self, bHYPRE_Operator A) <i>Set the operator for the linear system being solved</i>	128
6.6.4	int32_t bHYPRE_Pilut_SetTolerance (bHYPRE_Pilut self, double tolerance) <i>(Optional) Set the convergence tolerance</i>	128
6.6.5	int32_t	

	bHYPRE_Pilut_SetMaxIterations (bHYPRE_Pilut self, int32_t max_iterations)	
	<i>(Optional) Set maximum number of iterations</i>	128
6.6.6	int32_t bHYPRE_Pilut_SetLogging (bHYPRE_Pilut self, int32_t level) <i>(Optional) Set the logging level, specifying the degree of additional informational data to be accumulated</i>	129
6.6.7	int32_t bHYPRE_Pilut_SetPrintLevel (bHYPRE_Pilut self, int32_t level) <i>(Optional) Set the print level, specifying the degree of informational data to be printed either to the screen or to a file</i>	129
	int32_t bHYPRE_Pilut_GetNumIterations (bHYPRE_Pilut self, int32_t* num_iterations)	
	<i>(Optional) Return the number of iterations taken</i>	
	int32_t bHYPRE_Pilut_GetRelResidualNorm (bHYPRE_Pilut self, double* norm)	
	<i>(Optional) Return the norm of the relative residual</i>	
	struct bHYPRE_Pilut__object* bHYPRE_Pilut__cast void* obj <i>Cast method for interface and class type conversions</i>	
	void* bHYPRE_Pilut__cast2 (void* obj, const char* type) <i>String cast method for interface and class type conversions</i>	
	void bHYPRE_Pilut__exec (bHYPRE_Pilut self, const char* methodName, sidl_io_Deserializer inArgs, sidl_io_Serializer outArgs) <i>Select and execute a method by name</i>	
	char* bHYPRE_Pilut__getURL (bHYPRE_Pilut self) <i>Get the URL of the Implementation of this object (for RMI)</i>	

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

```
int32_t  
bHYPRE_Pilut_SetCommunicator ( bHYPRE_Pilut self,  
bHYPRE_MPICommunicator mpi_comm)
```

Set the MPI Communicator. DEPRECATED, use Create:

6.6.3

```
int32_t  
bHYPRE_Pilut_SetOperator ( bHYPRE_Pilut self, bHYPRE_Operator A)
```

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

6.6.4

```
int32_t bHYPRE_Pilut_SetTolerance ( bHYPRE_Pilut self, double tolerance)
```

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

6.6.5

```
int32_t  
bHYPRE_Pilut_SetMaxIterations ( bHYPRE_Pilut self, int32_t  
max_iterations)
```


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

6.6.6

```
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. 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

```
int32_t bHYPRE_Pilut_SetPrintLevel ( bHYPRE_Pilut 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. Does nothing by default (level=0). Other levels (if any) are implementation-specific. Must be called before **Setup** and **Apply**. DEPRECATED use SetIntParameter

Structured Matrix Solvers

Names

7.1	StructDiagScale Solver	130
7.2	Struct Jacobi Solver	135
7.3	Struct PFMG Solver	140
7.4	Struct SMG Solver	145

These solvers use structured matrix/vector storage schemes.

StructDiagScale Solver

Names

7.1.1	struct bHYPRE_StructDiagScale__object <i>Symbol "bHYPRE"</i>	133
	void <i>Constructor function for the class</i>	
	bHYPRE_StructDiagScale bHYPRE_StructDiagScale__createRemote (const char *, sidl_BaseInterface *_ex) <i>RMI constructor function for the class</i>	
	bHYPRE_StructDiagScale bHYPRE_StructDiagScale__connect (const char *, sidl_BaseInterface *_ex) <i>RMI connector function for the class</i>	
	bHYPRE_StructDiagScale bHYPRE_StructDiagScale_Create (bHYPRE_MPICommunicator mpi_comm, bHYPRE_StructMatrix A) <i>Method: Create[]</i>	
7.1.2	int32_t	

bHYPRE_StructDiagScale_SetCommunicator (bHYPRE_StructDiagScale
self,
bHYPRE_MPICommunicator
mpi_comm)
Set the MPI Communicator 134

int32_t
bHYPRE_StructDiagScale_SetIntParameter (bHYPRE_StructDiagScale
self, const char* name,
int32_t value)
Set the int parameter associated with name

int32_t
bHYPRE_StructDiagScale_SetDoubleParameter (
bHYPRE_StructDiagScale
self, const char* name,
double value)
Set the double parameter associated with name

int32_t
bHYPRE_StructDiagScale_SetStringParameter (
bHYPRE_StructDiagScale
self, const char* name,
const char* value)
Set the string parameter associated with name

int32_t
bHYPRE_StructDiagScale_SetIntArray1Parameter (
bHYPRE_StructDiagScale
self,
const char* name,
int32_t* value,
int32_t nvalues)
Set the int 1-D array parameter associated with name

int32_t
bHYPRE_StructDiagScale_SetIntArray2Parameter (
bHYPRE_StructDiagScale
self,
const char* name,
struct sidl_int__array*
value)
Set the int 2-D array parameter associated with name

int32_t
bHYPRE_StructDiagScale_SetDoubleArray1Parameter (
bHYPRE_StructDiagScale
self, const char*
name,
double* value,
int32_t nvalues)
Set the double 1-D array parameter associated with name

int32_t

	bHYPRE_StructDiagScale_SetDoubleArray2Parameter (<div style="text-align: right;"> bHYPRE_StructDiagScale self, const char* name, struct sidl_double__array* value) </div> <i>Set the double 2-D array parameter associated with name</i>	
	int32_t bHYPRE_StructDiagScale_GetIntValue (bHYPRE_StructDiagScale self, const char* name, int32_t* value) <i>Set the int parameter associated with name</i>	
	int32_t bHYPRE_StructDiagScale_GetDoubleValue (bHYPRE_StructDiagScale self, const char* name, double* value) <i>Get the double parameter associated with name</i>	
	int32_t bHYPRE_StructDiagScale_Setup (bHYPRE_StructDiagScale self, bHYPRE_Vector b, bHYPRE_Vector x) <i>(Optional) Do any preprocessing that may be necessary in order to execute Apply</i>	
	int32_t bHYPRE_StructDiagScale_Apply (bHYPRE_StructDiagScale self, bHYPRE_Vector b, bHYPRE_Vector* x) <i>Apply the operator to b, returning x</i>	
	int32_t bHYPRE_StructDiagScale_ApplyAdjoint (bHYPRE_StructDiagScale self, bHYPRE_Vector b, bHYPRE_Vector* x) <i>Apply the adjoint of the operator to b, returning x</i>	
7.1.3	int32_t bHYPRE_StructDiagScale_SetOperator (bHYPRE_StructDiagScale self, bHYPRE_Operator A) <i>Set the operator for the linear system being solved</i>	134
7.1.4	int32_t bHYPRE_StructDiagScale_SetTolerance (bHYPRE_StructDiagScale self, double tolerance) <i>(Optional) Set the convergence tolerance</i>	134
7.1.5	int32_t bHYPRE_StructDiagScale_SetMaxIterations (bHYPRE_StructDiagScale self, int32_t max_iterations) <i>(Optional) Set maximum number of iterations</i>	134
7.1.6	int32_t bHYPRE_StructDiagScale_SetLogging (bHYPRE_StructDiagScale self, int32_t level) <i>(Optional) Set the logging level, specifying the degree of additional informational data to be accumulated</i>	135
7.1.7	int32_t	

bHYPRE_StructDiagScale_SetPrintLevel (bHYPRE_StructDiagScale 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 135*

int32_t
bHYPRE_StructDiagScale_GetNumIterations (bHYPRE_StructDiagScale
self,
int32_t* num_iterations)
(Optional) Return the number of iterations taken

int32_t
bHYPRE_StructDiagScale_GetRelResidualNorm (
bHYPRE_StructDiagScale
self, double* norm)
(Optional) Return the norm of the relative residual

obj
Cast method for interface and class type conversions

void*
bHYPRE_StructDiagScale__cast2 (void* obj, const char* type)
String cast method for interface and class type conversions

void
bHYPRE_StructDiagScale__exec (bHYPRE_StructDiagScale self,
const char* methodName,
sidl_io_Deserializer inArgs,
sidl_io_Serializer outArgs)
Select and execute a method by name

char*
bHYPRE_StructDiagScale__getURL (bHYPRE_StructDiagScale self)
Get the URL of the Implementation of this object (for RMI)

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

```
int32_t
bHYPRE_StructDiagScale_SetCommunicator ( bHYPRE_StructDiagScale
self, bHYPRE_MPICommunicator mpi.comm)
```

Set the MPI Communicator. DEPRECATED, use Create:

7.1.3

```
int32_t
bHYPRE_StructDiagScale_SetOperator ( bHYPRE_StructDiagScale self,
bHYPRE_Operator A)
```

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

7.1.4

```
int32_t
bHYPRE_StructDiagScale_SetTolerance ( bHYPRE_StructDiagScale self,
double tolerance)
```

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

7.1.5

```
int32_t
bHYPRE_StructDiagScale_SetMaxIterations ( bHYPRE_StructDiagScale
self, int32_t max.iterations)
```

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

7.1.6

```
int32_t
bHYPRE_StructDiagScale_SetLogging ( bHYPRE_StructDiagScale self,
int32_t level)
```

(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

```
int32_t
bHYPRE_StructDiagScale_SetPrintLevel ( bHYPRE_StructDiagScale 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. 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

Struct Jacobi Solver

Names

7.2.1	struct bHYPRE_StructJacobi_object <i>Symbol "bHYPRE"</i>	138
	void <i>Constructor function for the class</i> bHYPRE_StructJacobi	

bHYPRE_StructJacobi__createRemote (const char *,
sidl_BaseInterface *_ex)
RMI constructor function for the class

bHYPRE_StructJacobi

bHYPRE_StructJacobi__connect (const char *, sidl_BaseInterface *_ex)
RMI connector function for the class

bHYPRE_StructJacobi

bHYPRE_StructJacobi_Create (bHYPRE_MPICommunicator mpi_comm,
bHYPRE_StructMatrix A)
Method: Create[]

7.2.2

int32_t

bHYPRE_StructJacobi_SetCommunicator (bHYPRE_StructJacobi self,
bHYPRE_MPICommunicator
mpi_comm)
Set the MPI Communicator 139

int32_t

bHYPRE_StructJacobi_SetIntParameter (bHYPRE_StructJacobi self,
const char* name, int32_t value)
Set the int parameter associated with name

int32_t

bHYPRE_StructJacobi_SetDoubleParameter (bHYPRE_StructJacobi self,
const char* name,
double value)
Set the double parameter associated with name

int32_t

bHYPRE_StructJacobi_SetStringParameter (bHYPRE_StructJacobi self,
const char* name,
const char* value)
Set the string parameter associated with name

int32_t

bHYPRE_StructJacobi_SetIntArray1Parameter (bHYPRE_StructJacobi
self, const char* name,
int32_t* value,
int32_t nvalues)
Set the int 1-D array parameter associated with name

int32_t

bHYPRE_StructJacobi_SetIntArray2Parameter (bHYPRE_StructJacobi
self, const char* name,
struct sidl_int_array*
value)
Set the int 2-D array parameter associated with name

int32_t

bHYPRE_StructJacobi_SetDoubleArray1Parameter (
bHYPRE_StructJacobi
self,
const char* name,
double* value,
int32_t nvalues)
Set the double 1-D array parameter associated with name

int32_t

	bHYPRE_StructJacobi_SetDoubleArray2Parameter (<div style="text-align: right;"> bHYPRE_StructJacobi self, const char* name, struct sidl_double_array* value) </div> <i>Set the double 2-D array parameter associated with name</i>	
	int32_t bHYPRE_StructJacobi_GetIntValue (bHYPRE_StructJacobi self, const char* name, int32_t* value) <i>Set the int parameter associated with name</i>	
	int32_t bHYPRE_StructJacobi_GetDoubleValue (bHYPRE_StructJacobi self, const char* name, double* value) <i>Get the double parameter associated with name</i>	
	int32_t bHYPRE_StructJacobi_Setup (bHYPRE_StructJacobi self, bHYPRE_Vector b, bHYPRE_Vector x) <i>(Optional) Do any preprocessing that may be necessary in order to execute Apply</i>	
	int32_t bHYPRE_StructJacobi_Apply (bHYPRE_StructJacobi self, bHYPRE_Vector b, bHYPRE_Vector* x) <i>Apply the operator to b, returning x</i>	
	int32_t bHYPRE_StructJacobi_ApplyAdjoint (bHYPRE_StructJacobi self, bHYPRE_Vector b, bHYPRE_Vector* x) <i>Apply the adjoint of the operator to b, returning x</i>	
7.2.3	int32_t bHYPRE_StructJacobi_SetOperator (bHYPRE_StructJacobi self, bHYPRE_Operator A) <i>Set the operator for the linear system being solved</i>	139
7.2.4	int32_t bHYPRE_StructJacobi_SetTolerance (bHYPRE_StructJacobi self, double tolerance) <i>(Optional) Set the convergence tolerance</i>	139
7.2.5	int32_t bHYPRE_StructJacobi_SetMaxIterations (bHYPRE_StructJacobi self, int32_t max_iterations) <i>(Optional) Set maximum number of iterations</i>	139
7.2.6	int32_t bHYPRE_StructJacobi_SetLogging (bHYPRE_StructJacobi self, int32_t level) <i>(Optional) Set the logging level, specifying the degree of additional informa- tional data to be accumulated</i>	140
7.2.7	int32_t	

```

bHYPRE_StructJacobi_SetPrintLevel ( bHYPRE_StructJacobi 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 ..... 140

int32_t
bHYPRE_StructJacobi_GetNumIterations ( bHYPRE_StructJacobi self,
                                     int32_t* num_iterations)
    (Optional) Return the number of iterations taken

int32_t
bHYPRE_StructJacobi_GetRelResidualNorm ( bHYPRE_StructJacobi self,
                                     double* norm)
    (Optional) Return the norm of the relative residual

struct bHYPRE_StructJacobi__object* bHYPRE_StructJacobi__cast void* obj
    Cast method for interface and class type conversions

void*
bHYPRE_StructJacobi__cast2 ( void* obj, const char* type)
    String cast method for interface and class type conversions

void
bHYPRE_StructJacobi__exec ( bHYPRE_StructJacobi self,
                             const char* methodName,
                             sidl_io_Deserializer inArgs,
                             sidl_io_Serializer outArgs)
    Select and execute a method by name

char*
bHYPRE_StructJacobi__getURL ( bHYPRE_StructJacobi self)
    Get the URL of the Implementation of this object (for RMI)

```

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

```
int32_t  
bHYPRE_StructJacobi_SetCommunicator ( bHYPRE_StructJacobi self,  
bHYPRE_MPICommunicator mpi_comm)
```

Set the MPI Communicator. DEPRECATED, use Create:

7.2.3

```
int32_t  
bHYPRE_StructJacobi_SetOperator ( bHYPRE_StructJacobi self,  
bHYPRE_Operator A)
```

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

7.2.4

```
int32_t  
bHYPRE_StructJacobi_SetTolerance ( bHYPRE_StructJacobi self, double  
tolerance)
```

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

7.2.5

```
int32_t  
bHYPRE_StructJacobi_SetMaxIterations ( bHYPRE_StructJacobi self,  
int32_t max_iterations)
```

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

7.2.6

```
int32_t
bHYPRE_StructJacobi_SetLogging ( bHYPRE_StructJacobi self, int32_t
level)
```

(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

```
int32_t
bHYPRE_StructJacobi_SetPrintLevel ( bHYPRE_StructJacobi 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. 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

Struct PFMG Solver

Names

7.3.1	struct bHYPRE_StructPFMG__object <i>Symbol "bHYPRE"</i>	143
	void <i>Constructor function for the class</i>	
	bHYPRE_StructPFMG	

bHYPRE_StructPFMG__createRemote (const char *,
sidl_BaseInterface *_ex)

RMI constructor function for the class

bHYPRE_StructPFMG

bHYPRE_StructPFMG__connect (const char *, sidl_BaseInterface *_ex)

RMI connector function for the class

bHYPRE_StructPFMG

bHYPRE_StructPFMG_Create (bHYPRE_MPICommunicator mpi_comm,
bHYPRE_StructMatrix A)

Method: Create[]

7.3.2

int32_t

bHYPRE_StructPFMG_SetCommunicator (bHYPRE_StructPFMG self,
bHYPRE_MPICommunicator
mpi_comm)

Set the MPI Communicator

144

int32_t

bHYPRE_StructPFMG_SetIntParameter (bHYPRE_StructPFMG self,
const char* name, int32_t value)

Set the int parameter associated with name

int32_t

bHYPRE_StructPFMG_SetDoubleParameter (bHYPRE_StructPFMG
self, const char* name,
double value)

Set the double parameter associated with name

int32_t

bHYPRE_StructPFMG_SetStringParameter (bHYPRE_StructPFMG self,
const char* name,
const char* value)

Set the string parameter associated with name

int32_t

bHYPRE_StructPFMG_SetIntArray1Parameter (bHYPRE_StructPFMG
self, const char* name,
int32_t* value,
int32_t nvalues)

Set the int 1-D array parameter associated with name

int32_t

bHYPRE_StructPFMG_SetIntArray2Parameter (bHYPRE_StructPFMG
self, const char* name,
struct sidl_int_array*
value)

Set the int 2-D array parameter associated with name

int32_t

bHYPRE_StructPFMG_SetDoubleArray1Parameter (
bHYPRE_StructPFMG
self,
const char* name,
double* value,
int32_t nvalues)

Set the double 1-D array parameter associated with name

int32_t

	bHYPRE_StructPFMG_SetDoubleArray2Parameter (<div style="text-align: right;"> bHYPRE_StructPFMG self, const char* name, struct sidl_double_array* value) </div> <i>Set the double 2-D array parameter associated with name</i>	
	int32_t bHYPRE_StructPFMG_GetIntValue (bHYPRE_StructPFMG self, const char* name, int32_t* value) <i>Set the int parameter associated with name</i>	
	int32_t bHYPRE_StructPFMG_GetDoubleValue (bHYPRE_StructPFMG self, const char* name, double* value) <i>Get the double parameter associated with name</i>	
	int32_t bHYPRE_StructPFMG_Setup (bHYPRE_StructPFMG self, bHYPRE_Vector b, bHYPRE_Vector x) <i>(Optional) Do any preprocessing that may be necessary in order to execute Apply</i>	
	int32_t bHYPRE_StructPFMG_Apply (bHYPRE_StructPFMG self, bHYPRE_Vector b, bHYPRE_Vector* x) <i>Apply the operator to b, returning x</i>	
	int32_t bHYPRE_StructPFMG_ApplyAdjoint (bHYPRE_StructPFMG self, bHYPRE_Vector b, bHYPRE_Vector* x) <i>Apply the adjoint of the operator to b, returning x</i>	
7.3.3	int32_t bHYPRE_StructPFMG_SetOperator (bHYPRE_StructPFMG self, bHYPRE_Operator A) <i>Set the operator for the linear system being solved</i>	144
7.3.4	int32_t bHYPRE_StructPFMG_SetTolerance (bHYPRE_StructPFMG self, double tolerance) <i>(Optional) Set the convergence tolerance</i>	144
7.3.5	int32_t bHYPRE_StructPFMG_SetMaxIterations (bHYPRE_StructPFMG self, int32_t max_iterations) <i>(Optional) Set maximum number of iterations</i>	144
7.3.6	int32_t bHYPRE_StructPFMG_SetLogging (bHYPRE_StructPFMG self, int32_t level) <i>(Optional) Set the logging level, specifying the degree of additional informational data to be accumulated</i>	145
7.3.7	int32_t	

```

bHYPRE_StructPFMG_SetPrintLevel ( bHYPRE_StructPFMG 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 ..... 145

int32_t
bHYPRE_StructPFMG_GetNumIterations ( bHYPRE_StructPFMG self,
                                       int32_t* num_iterations)
    (Optional) Return the number of iterations taken

int32_t
bHYPRE_StructPFMG_GetRelResidualNorm ( bHYPRE_StructPFMG
                                       self, double* norm)
    (Optional) Return the norm of the relative residual

struct bHYPRE_StructPFMG__object* bHYPRE_StructPFMG__cast void* obj
    Cast method for interface and class type conversions

void*
bHYPRE_StructPFMG__cast2 ( void* obj, const char* type)
    String cast method for interface and class type conversions

void
bHYPRE_StructPFMG__exec ( bHYPRE_StructPFMG self,
                           const char* methodName,
                           sidl_io_Deserializer inArgs,
                           sidl_io_Serializer outArgs)
    Select and execute a method by name

char*
bHYPRE_StructPFMG__getURL ( bHYPRE_StructPFMG self)
    Get the URL of the Implementation of this object (for RMI)

```

7.3.1

```
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

```
int32_t  
bHYPRE_StructPFMG_SetCommunicator ( bHYPRE_StructPFMG self,  
bHYPRE_MPICommunicator mpi_comm)
```

Set the MPI Communicator. DEPRECATED, use Create:

7.3.3

```
int32_t  
bHYPRE_StructPFMG_SetOperator ( bHYPRE_StructPFMG self,  
bHYPRE_Operator A)
```

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

7.3.4

```
int32_t  
bHYPRE_StructPFMG_SetTolerance ( bHYPRE_StructPFMG self, double  
tolerance)
```

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

7.3.5

```
int32_t  
bHYPRE_StructPFMG_SetMaxIterations ( bHYPRE_StructPFMG self,  
int32_t max_iterations)
```


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

7.3.6

```
int32_t
bHYPRE_StructPFMG_SetLogging ( bHYPRE_StructPFMG self, int32_t
level)
```

(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

```
int32_t
bHYPRE_StructPFMG_SetPrintLevel ( bHYPRE_StructPFMG 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. 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

Struct SMG Solver

Names

7.4.1	struct bHYPRE_StructSMG_object <i>Symbol "bHYPRE"</i>	148
	void <i>Constructor function for the class</i>	
	bHYPRE_StructSMG	

bHYPRE_StructSMG__createRemote (const char *, sidl_BaseInterface *_ex)
RMI constructor function for the class

bHYPRE_StructSMG

bHYPRE_StructSMG__connect (const char *, sidl_BaseInterface *_ex)
RMI connector function for the class

bHYPRE_StructSMG

bHYPRE_StructSMG_Create (bHYPRE_MPICommunicator mpi_comm,
bHYPRE_StructMatrix A)

Method: Create[]

7.4.2

int32_t

bHYPRE_StructSMG_SetCommunicator (bHYPRE_StructSMG self,
bHYPRE_MPICommunicator
mpi_comm)

Set the MPI Communicator 149

int32_t

bHYPRE_StructSMG_SetIntParameter (bHYPRE_StructSMG self,
const char* name, int32_t value)

Set the int parameter associated with name

int32_t

bHYPRE_StructSMG_SetDoubleParameter (bHYPRE_StructSMG self,
const char* name,
double value)

Set the double parameter associated with name

int32_t

bHYPRE_StructSMG_SetStringParameter (bHYPRE_StructSMG self,
const char* name,
const char* value)

Set the string parameter associated with name

int32_t

bHYPRE_StructSMG_SetIntArray1Parameter (bHYPRE_StructSMG self,
const char* name,
int32_t* value,
int32_t nvalues)

Set the int 1-D array parameter associated with name

int32_t

bHYPRE_StructSMG_SetIntArray2Parameter (bHYPRE_StructSMG self,
const char* name, struct
sidl_int_array* value)

Set the int 2-D array parameter associated with name

int32_t

bHYPRE_StructSMG_SetDoubleArray1Parameter (
bHYPRE_StructSMG
self,
const char* name,
double* value,
int32_t nvalues)

Set the double 1-D array parameter associated with name

int32_t

	bHYPRE_StructSMG_SetDoubleArray2Parameter (<div style="float: right; text-align: right;"> bHYPRE_StructSMG self, const char* name, struct sidl_double_array* value) </div> <i>Set the double 2-D array parameter associated with name</i>	
	int32_t bHYPRE_StructSMG_GetIntValue (bHYPRE_StructSMG self, const char* name, int32_t* value) <i>Set the int parameter associated with name</i>	
	int32_t bHYPRE_StructSMG_GetDoubleValue (bHYPRE_StructSMG self, const char* name, double* value) <i>Get the double parameter associated with name</i>	
	int32_t bHYPRE_StructSMG_Setup (bHYPRE_StructSMG self, bHYPRE_Vector b, bHYPRE_Vector x) <i>(Optional) Do any preprocessing that may be necessary in order to execute Apply</i>	
	int32_t bHYPRE_StructSMG_Apply (bHYPRE_StructSMG self, bHYPRE_Vector b, bHYPRE_Vector* x) <i>Apply the operator to b, returning x</i>	
	int32_t bHYPRE_StructSMG_ApplyAdjoint (bHYPRE_StructSMG self, bHYPRE_Vector b, bHYPRE_Vector* x) <i>Apply the adjoint of the operator to b, returning x</i>	
7.4.3	int32_t bHYPRE_StructSMG_SetOperator (bHYPRE_StructSMG self, bHYPRE_Operator A) <i>Set the operator for the linear system being solved</i>	149
7.4.4	int32_t bHYPRE_StructSMG_SetTolerance (bHYPRE_StructSMG self, double tolerance) <i>(Optional) Set the convergence tolerance</i>	149
7.4.5	int32_t bHYPRE_StructSMG_SetMaxIterations (bHYPRE_StructSMG self, int32_t max_iterations) <i>(Optional) Set maximum number of iterations</i>	149
7.4.6	int32_t bHYPRE_StructSMG_SetLogging (bHYPRE_StructSMG self, int32_t level) <i>(Optional) Set the logging level, specifying the degree of additional informa- tional data to be accumulated</i>	150
7.4.7	int32_t	

```

bHYPRE_StructSMG_SetPrintLevel ( bHYPRE_StructSMG 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 ..... 150

int32_t
bHYPRE_StructSMG_GetNumIterations ( bHYPRE_StructSMG self,
                                       int32_t* num_iterations)
    (Optional) Return the number of iterations taken

int32_t
bHYPRE_StructSMG_GetRelResidualNorm ( bHYPRE_StructSMG self,
                                         double* norm)
    (Optional) Return the norm of the relative residual

struct bHYPRE_StructSMG__object* bHYPRE_StructSMG__cast void* obj
    Cast method for interface and class type conversions

void*
bHYPRE_StructSMG__cast2 ( void* obj, const char* type)
    String cast method for interface and class type conversions

void
bHYPRE_StructSMG__exec ( bHYPRE_StructSMG self,
                          const char* methodName,
                          sidl_io_Deserializer inArgs,
                          sidl_io_Serializer outArgs)
    Select and execute a method by name

char*
bHYPRE_StructSMG__getURL ( bHYPRE_StructSMG self)
    Get the URL of the Implementation of this object (for RMI)

```

7.4.1

```
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

```
int32_t  
bHYPRE_StructSMG_SetCommunicator ( bHYPRE_StructSMG self,  
bHYPRE_MPICommunicator mpi_comm)
```

Set the MPI Communicator. DEPRECATED, use Create:

7.4.3

```
int32_t  
bHYPRE_StructSMG_SetOperator ( bHYPRE_StructSMG self,  
bHYPRE_Operator A)
```

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

7.4.4

```
int32_t  
bHYPRE_StructSMG_SetTolerance ( bHYPRE_StructSMG self, double  
tolerance)
```

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

7.4.5

```
int32_t  
bHYPRE_StructSMG_SetMaxIterations ( bHYPRE_StructSMG self, int32_t  
max_iterations)
```

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

7.4.6

```
int32_t
bHYPRE_StructSMG_SetLogging ( bHYPRE_StructSMG self, int32_t level)
```

(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

```
int32_t
bHYPRE_StructSMG_SetPrintLevel ( bHYPRE_StructSMG 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. Does nothing by default (level=0). Other levels (if any) are implementation-specific. Must be called before **Setup** and **Apply**. DEPRECATED use SetIntParameter

SemiStructured Matrix Solvers

Names

8.1	SemiStruct DiagScale Solver	151
8.2	Struct Split Solver	156

These solvers use semi-structured matrix/vector storage schemes.

SemiStruct DiagScale Solver

Names

8.1.1	struct bHYPRE_SStructDiagScale__object <i>Symbol "bHYPRE"</i>	154
	void <i>Constructor function for the class</i>	
	bHYPRE_SStructDiagScale bHYPRE_SStructDiagScale__createRemote (const char *, sidl_BaseInterface *_ex) <i>RMI constructor function for the class</i>	
	bHYPRE_SStructDiagScale bHYPRE_SStructDiagScale__connect (const char *, sidl_BaseInterface *_ex) <i>RMI connector function for the class</i>	
	bHYPRE_SStructDiagScale bHYPRE_SStructDiagScale_Create (bHYPRE_MPICommunicator mpi_comm, bHYPRE_Operator A) <i>Method: Create[]</i>	
8.1.2	int32_t bHYPRE_SStructDiagScale_SetCommunicator (bHYPRE_SStructDiagScale self, bHYPRE_MPICommunicator mpi_comm) <i>Set the MPI Communicator</i>	154
	int32_t	

bHYPRE_SStructDiagScale_SetIntParameter (bHYPRE_SStructDiagScale
self, const char* name,
int32_t value)

Set the int parameter associated with name

int32_t

bHYPRE_SStructDiagScale_SetDoubleParameter (
bHYPRE_SStructDiagScale
self, const char* name,
double value)

Set the double parameter associated with name

int32_t

bHYPRE_SStructDiagScale_SetStringParameter (
bHYPRE_SStructDiagScale
self, const char* name,
const char* value)

Set the string parameter associated with name

int32_t

bHYPRE_SStructDiagScale_SetIntArray1Parameter (
bHYPRE_SStructDiagScale
self,
const char* name,
int32_t* value,
int32_t nvalues)

Set the int 1-D array parameter associated with name

int32_t

bHYPRE_SStructDiagScale_SetIntArray2Parameter (
bHYPRE_SStructDiagScale
self,
const char* name,
struct
sidl_int__array*
value)

Set the int 2-D array parameter associated with name

int32_t

bHYPRE_SStructDiagScale_SetDoubleArray1Parameter (
bHYPRE_SStructDiagScale
self, const
char* name,
double* value,
int32_t nvalues)

Set the double 1-D array parameter associated with name

int32_t

bHYPRE_SStructDiagScale_SetDoubleArray2Parameter (
bHYPRE_SStructDiagScale
self, const
char* name,
struct
sidl_double__array*
value)

Set the double 2-D array parameter associated with name

int32_t

	bHYPRE_SStructDiagScale_GetInt Value (bHYPRE_SStructDiagScale self, const char* name, int32_t* value) <i>Set the int parameter associated with name</i>	
	int32_t bHYPRE_SStructDiagScale_GetDoubleValue (bHYPRE_SStructDiagScale self, const char* name, double* value) <i>Get the double parameter associated with name</i>	
	int32_t bHYPRE_SStructDiagScale_Setup (bHYPRE_SStructDiagScale self, bHYPRE_Vector b, bHYPRE_Vector x) <i>(Optional) Do any preprocessing that may be necessary in order to execute Apply</i>	
	int32_t bHYPRE_SStructDiagScale_Apply (bHYPRE_SStructDiagScale self, bHYPRE_Vector b, bHYPRE_Vector* x) <i>Apply the operator to b, returning x</i>	
	int32_t bHYPRE_SStructDiagScale_ApplyAdjoint (bHYPRE_SStructDiagScale self, bHYPRE_Vector b, bHYPRE_Vector* x) <i>Apply the adjoint of the operator to b, returning x</i>	
8.1.3	int32_t bHYPRE_SStructDiagScale_SetOperator (bHYPRE_SStructDiagScale self, bHYPRE_Operator A) <i>Set the operator for the linear system being solved</i>	155
8.1.4	int32_t bHYPRE_SStructDiagScale_SetTolerance (bHYPRE_SStructDiagScale self, double tolerance) <i>(Optional) Set the convergence tolerance</i>	155
8.1.5	int32_t bHYPRE_SStructDiagScale_SetMaxIterations (bHYPRE_SStructDiagScale self, int32_t max_iterations) <i>(Optional) Set maximum number of iterations</i>	155
8.1.6	int32_t bHYPRE_SStructDiagScale_SetLogging (bHYPRE_SStructDiagScale self, int32_t level) <i>(Optional) Set the logging level, specifying the degree of additional informa- tional data to be accumulated</i>	155
8.1.7	int32_t bHYPRE_SStructDiagScale_SetPrintLevel (bHYPRE_SStructDiagScale self, int32_t level) <i>(Optional) Set the print level, specifying the degree of informational data to be printed either to the screen or to a file</i>	156
	int32_t	

```

bHYPRE_SStructDiagScale_GetNumIterations (
    bHYPRE_SStructDiagScale
    self,
    int32_t* num_iterations)
    (Optional) Return the number of iterations taken

int32_t
bHYPRE_SStructDiagScale_GetRelResidualNorm (
    bHYPRE_SStructDiagScale
    self, double* norm)
    (Optional) Return the norm of the relative residual

obj
    Cast method for interface and class type conversions

void*
bHYPRE_SStructDiagScale__cast2 ( void* obj, const char* type)
    String cast method for interface and class type conversions

void
bHYPRE_SStructDiagScale__exec ( bHYPRE_SStructDiagScale self,
    const char* methodName,
    sidl_io.Deserializer inArgs,
    sidl_io.Serializer outArgs)
    Select and execute a method by name

char*
bHYPRE_SStructDiagScale__getURL ( bHYPRE_SStructDiagScale self)
    Get the URL of the Implementation of this object (for RMI)

```

8.1.1

```
struct bHYPRE_SStructDiagScale__object
```

Symbol "bHYPRE.SStructDiagScale" (version 1.0.0)

8.1.2

```

int32_t
bHYPRE_SStructDiagScale_SetCommunicator ( bHYPRE_SStructDiagScale
self, bHYPRE_MPICommunicator mpi_comm)

```

Set the MPI Communicator. DEPRECATED, use Create:

8.1.3

```
int32_t
bHYPRE_SStructDiagScale_SetOperator ( bHYPRE_SStructDiagScale self,
bHYPRE_Operator A)
```

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

8.1.4

```
int32_t
bHYPRE_SStructDiagScale_SetTolerance ( bHYPRE_SStructDiagScale self,
double tolerance)
```

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

8.1.5

```
int32_t
bHYPRE_SStructDiagScale_SetMaxIterations ( bHYPRE_SStructDiagScale
self, int32_t max_iterations)
```

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

8.1.6

```
int32_t
bHYPRE_SStructDiagScale_SetLogging ( bHYPRE_SStructDiagScale self,
int32_t level)
```

```
int32_t
bHYPRE_SStructDiagScale_SetPrintLevel ( bHYPRE_SStructDiagScale self,
int32_t level)
```

8.2 Struct Split Solver

```

8.2.1      struct  bHYPRE_SStructSplit__object
              Symbol "bHYPRE_SStructSplit__object" ..... 159

void
    Constructor function for the class

bHYPRE_SStructSplit
bHYPRE_SStructSplit__createRemote (const char *,
                                     sidl_BaseInterface *_ex)
    RMI constructor function for the class

bHYPRE_SStructSplit
bHYPRE_SStructSplit__connect (const char *, sidl_BaseInterface *_ex)
    RMI connector function for the class

bHYPRE_SStructSplit
bHYPRE_SStructSplit__Create ( bHYPRE_MPICommunicator mpi_comm,
                               bHYPRE_Operator A)
    Method: Create[]
8.2.2      int32_t

```

bHYPRE_SStructSplit_SetCommunicator (bHYPRE_SStructSplit self, bHYPRE_MPICommunicator mpi.comm)	
<i>Set the MPI Communicator</i>	159
int32_t bHYPRE_SStructSplit_SetIntParameter (bHYPRE_SStructSplit self, const char* name, int32_t value)	
<i>Set the int parameter associated with name</i>	
int32_t bHYPRE_SStructSplit_SetDoubleParameter (bHYPRE_SStructSplit self, const char* name, double value)	
<i>Set the double parameter associated with name</i>	
int32_t bHYPRE_SStructSplit_SetStringParameter (bHYPRE_SStructSplit self, const char* name, const char* value)	
<i>Set the string parameter associated with name</i>	
int32_t bHYPRE_SStructSplit_SetIntArray1Parameter (bHYPRE_SStructSplit self, const char* name, int32_t* value, int32_t nvalues)	
<i>Set the int 1-D array parameter associated with name</i>	
int32_t bHYPRE_SStructSplit_SetIntArray2Parameter (bHYPRE_SStructSplit self, const char* name, struct sidl_int__array* value)	
<i>Set the int 2-D array parameter associated with name</i>	
int32_t bHYPRE_SStructSplit_SetDoubleArray1Parameter (bHYPRE_SStructSplit self, const char* name, double* value, int32_t nvalues)	
<i>Set the double 1-D array parameter associated with name</i>	
int32_t bHYPRE_SStructSplit_SetDoubleArray2Parameter (bHYPRE_SStructSplit self, const char* name, struct sidl_double__array* value)	
<i>Set the double 2-D array parameter associated with name</i>	
int32_t	

	bHYPRE_SStructSplit_GetIntValue (bHYPRE_SStructSplit self, const char* name, int32_t* value) <i>Set the int parameter associated with name</i>	
	int32_t bHYPRE_SStructSplit_GetDoubleValue (bHYPRE_SStructSplit self, const char* name, double* value) <i>Get the double parameter associated with name</i>	
	int32_t bHYPRE_SStructSplit_Setup (bHYPRE_SStructSplit self, bHYPRE_Vector b, bHYPRE_Vector x) <i>(Optional) Do any preprocessing that may be necessary in order to execute Apply</i>	
	int32_t bHYPRE_SStructSplit_Apply (bHYPRE_SStructSplit self, bHYPRE_Vector b, bHYPRE_Vector* x) <i>Apply the operator to b, returning x</i>	
	int32_t bHYPRE_SStructSplit_ApplyAdjoint (bHYPRE_SStructSplit self, bHYPRE_Vector b, bHYPRE_Vector* x) <i>Apply the adjoint of the operator to b, returning x</i>	
8.2.3	int32_t bHYPRE_SStructSplit_SetOperator (bHYPRE_SStructSplit self, bHYPRE_Operator A) <i>Set the operator for the linear system being solved</i>	160
8.2.4	int32_t bHYPRE_SStructSplit_SetTolerance (bHYPRE_SStructSplit self, double tolerance) <i>(Optional) Set the convergence tolerance</i>	160
8.2.5	int32_t bHYPRE_SStructSplit_SetMaxIterations (bHYPRE_SStructSplit self, int32_t max_iterations) <i>(Optional) Set maximum number of iterations</i>	160
8.2.6	int32_t bHYPRE_SStructSplit_SetLogging (bHYPRE_SStructSplit self, int32_t level) <i>(Optional) Set the logging level, specifying the degree of additional informational data to be accumulated</i>	160
8.2.7	int32_t bHYPRE_SStructSplit_SetPrintLevel (bHYPRE_SStructSplit self, int32_t level) <i>(Optional) Set the print level, specifying the degree of informational data to be printed either to the screen or to a file</i>	161
	int32_t bHYPRE_SStructSplit_GetNumIterations (bHYPRE_SStructSplit self, int32_t* num_iterations) <i>(Optional) Return the number of iterations taken</i>	
	int32_t	

```

bHYPRE_SStructSplit_GetRelResidualNorm ( bHYPRE_SStructSplit self,
                                           double* norm)
    (Optional) Return the norm of the relative residual

    struct bHYPRE_SStructSplit__object* bHYPRE_SStructSplit__cast void* obj
    Cast method for interface and class type conversions

    void*
    bHYPRE_SStructSplit__cast2 ( void* obj, const char* type)
    String cast method for interface and class type conversions

    void
    bHYPRE_SStructSplit__exec ( bHYPRE_SStructSplit self,
                                const char* methodName,
                                sidl_io_Deserializer inArgs,
                                sidl_io_Serializer outArgs)
    Select and execute a method by name

    char*
    bHYPRE_SStructSplit__getURL ( bHYPRE_SStructSplit self)
    Get the URL of the Implementation of this object (for RMI)

```

8.2.1

```

struct bHYPRE_SStructSplit__object

```

Symbol "bHYPRE.SStructSplit" (version 1.0.0)

Documentation goes here.

The SStructSplit solver requires a SStruct matrix.

8.2.2

```

int32_t
bHYPRE_SStructSplit_SetCommunicator ( bHYPRE_SStructSplit self,
    bHYPRE_MPICommunicator mpi_comm)

```

Set the MPI Communicator. DEPRECATED, use Create:

8.2.3

```
int32_t
bHYPRE_SStructSplit_SetOperator ( bHYPRE_SStructSplit self,
bHYPRE_Operator A)
```

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

8.2.4

```
int32_t
bHYPRE_SStructSplit_SetTolerance ( bHYPRE_SStructSplit self, double
tolerance)
```

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

8.2.5

```
int32_t
bHYPRE_SStructSplit_SetMaxIterations ( bHYPRE_SStructSplit self,
int32_t max_iterations)
```

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

8.2.6

```
int32_t
bHYPRE_SStructSplit_SetLogging ( bHYPRE_SStructSplit self, int32_t level)
```


(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

```
int32_t
bHYPRE_SStructSplit_SetPrintLevel ( bHYPRE_SStructSplit 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. Does nothing by default (level=0). Other levels (if any) are implementation-specific. Must be called before **Setup** and **Apply**. DEPRECATED use `SetIntParameter`

PreconditionedSolver Interface

Names

9.1	<pre> struct bHYPRE_PreconditionedSolver__object <i>Symbol "bHYPRE"</i> extern C bHYPRE_PreconditionedSolver bHYPRE_PreconditionedSolver__connect (const char *, sidl_BaseInterface *_ex) <i>RMI connector function for the class</i> int32_t bHYPRE_PreconditionedSolver_SetPreconditioner (bHYPRE_PreconditionedSolver self, bHYPRE_Solver s) <i>Set the preconditioner</i> int32_t bHYPRE_PreconditionedSolver_GetPreconditioner (bHYPRE_PreconditionedSolver self, bHYPRE_Solver* s) <i>Method: GetPreconditioner[]</i> int32_t bHYPRE_PreconditionedSolver_Clone (bHYPRE_PreconditionedSolver self, bHYPRE_PreconditionedSolver* x) <i>Method: Clone[]</i> obj <i>Cast method for interface and class type conversions</i> void* bHYPRE_PreconditionedSolver__cast2 (void* obj, const char* type) <i>String cast method for interface and class type conversions</i> void bHYPRE_PreconditionedSolver__exec (bHYPRE_PreconditionedSolver self, const char* methodName, sidl_io_Deserializer inArgs, sidl_io_Serializer outArgs) <i>Select and execute a method by name</i> char* bHYPRE_PreconditionedSolver__getURL (bHYPRE_PreconditionedSolver self) <i>Get the URL of the Implementation of this object (for RMI)</i> </pre>	163
-----	---	-----

9.1

```
struct bHYPRE_PreconditionedSolver__object
```

Symbol "bHYPRE.PreconditionedSolver" (version 1.0.0)

10

Preconditioned Solvers

Names

10.1	PCG Preconditioned Solver	
	164
10.2	GMRES Preconditioned Solver	
	169
10.3	BiCGSTAB Preconditioned Solver	
	174
10.4	CGNR Preconditioned Solver	
	179

10.1

PCG Preconditioned Solver

Names

10.1.1	struct bHYPRE_PCG__object <i>Symbol "bHYPRE"</i>	167
	extern C struct bHYPRE_PCG__object* bHYPRE_PCG__create void <i>Constructor function for the class</i>	
	bHYPRE_PCG bHYPRE_PCG__createRemote (const char *, sidl_BaseInterface *_ex) <i>RMI constructor function for the class</i>	
	bHYPRE_PCG bHYPRE_PCG__connect (const char *, sidl_BaseInterface *_ex) <i>RMI connector function for the class</i>	
	bHYPRE_PCG bHYPRE_PCG_Create (bHYPRE_MPICommunicator mpi_comm, bHYPRE_Operator A) <i>Method: Create[]</i>	
10.1.2	int32_t bHYPRE_PCG_SetCommunicator (bHYPRE_PCG self, bHYPRE_MPICommunicator mpi_comm) <i>Set the MPI Communicator</i>	167
	int32_t	

```

bHYPRE_PCG_SetIntParameter ( bHYPRE_PCG self,  const char* name,
                               int32_t value)
    Set the int parameter associated with name

int32_t
bHYPRE_PCG_SetDoubleParameter ( bHYPRE_PCG self,
                                   const char* name,  double value)
    Set the double parameter associated with name

int32_t
bHYPRE_PCG_SetStringParameter ( bHYPRE_PCG self,
                                   const char* name,  const char* value)
    Set the string parameter associated with name

int32_t
bHYPRE_PCG_SetIntArray1Parameter ( bHYPRE_PCG self,
                                      const char* name,  int32_t* value,
                                      int32_t nvalues)
    Set the int 1-D array parameter associated with name

int32_t
bHYPRE_PCG_SetIntArray2Parameter ( bHYPRE_PCG self,
                                      const char* name,
                                      struct sidl_int_array* value)
    Set the int 2-D array parameter associated with name

int32_t
bHYPRE_PCG_SetDoubleArray1Parameter ( bHYPRE_PCG self,
                                         const char* name,
                                         double* value,
                                         int32_t nvalues)
    Set the double 1-D array parameter associated with name

int32_t
bHYPRE_PCG_SetDoubleArray2Parameter ( bHYPRE_PCG self,
                                         const char* name,  struct
                                         sidl_double_array* value)
    Set the double 2-D array parameter associated with name

int32_t
bHYPRE_PCG_GetIntValue ( bHYPRE_PCG self,  const char* name,
                           int32_t* value)
    Set the int parameter associated with name

int32_t
bHYPRE_PCG_GetDoubleValue ( bHYPRE_PCG self,  const char* name,
                              double* value)
    Get the double parameter associated with name

int32_t
bHYPRE_PCG_Setup ( bHYPRE_PCG self,  bHYPRE_Vector b,
                    bHYPRE_Vector x)
    (Optional) Do any preprocessing that may be necessary in order to execute
    Apply
int32_t

```

	bHYPRE_PCG_Apply (bHYPRE_PCG self, bHYPRE_Vector b, bHYPRE_Vector* x) <i>Apply the operator to b, returning x</i>	
	int32_t bHYPRE_PCG_ApplyAdjoint (bHYPRE_PCG self, bHYPRE_Vector b, bHYPRE_Vector* x) <i>Apply the adjoint of the operator to b, returning x</i>	
10.1.3	int32_t bHYPRE_PCG_SetOperator (bHYPRE_PCG self, bHYPRE_Operator A) <i>Set the operator for the linear system being solved</i>	167
10.1.4	int32_t bHYPRE_PCG_SetTolerance (bHYPRE_PCG self, double tolerance) <i>(Optional) Set the convergence tolerance</i>	168
10.1.5	int32_t bHYPRE_PCG_SetMaxIterations (bHYPRE_PCG self, int32_t max_iterations) <i>(Optional) Set maximum number of iterations</i>	168
10.1.6	int32_t bHYPRE_PCG_SetLogging (bHYPRE_PCG self, int32_t level) <i>(Optional) Set the logging level, specifying the degree of additional informa- tional data to be accumulated</i>	168
10.1.7	int32_t bHYPRE_PCG_SetPrintLevel (bHYPRE_PCG self, int32_t level) <i>(Optional) Set the print level, specifying the degree of informational data to be printed either to the screen or to a file</i>	168
	int32_t bHYPRE_PCG_GetNumIterations (bHYPRE_PCG self, int32_t* num_iterations) <i>(Optional) Return the number of iterations taken</i>	
	int32_t bHYPRE_PCG_GetRelResidualNorm (bHYPRE_PCG self, double* norm) <i>(Optional) Return the norm of the relative residual</i>	
	int32_t bHYPRE_PCG_SetPreconditioner (bHYPRE_PCG self, bHYPRE_Solver s) <i>Set the preconditioner</i>	
	int32_t bHYPRE_PCG_GetPreconditioner (bHYPRE_PCG self, bHYPRE_Solver* s) <i>Method: GetPreconditioner[]</i>	
	int32_t bHYPRE_PCG_Clone (bHYPRE_PCG self, bHYPRE_PreconditionedSolver* x) <i>Method: Clone[]</i>	
	struct bHYPRE_PCG__object* bHYPRE_PCG__cast void* obj <i>Cast method for interface and class type conversions</i>	
	void*	

```

bHYPRE_PCG__cast2 ( void* obj, const char* type)
    String cast method for interface and class type conversions

void
bHYPRE_PCG__exec ( bHYPRE_PCG self, const char* methodName,
                    sidl_io_Deserializer inArgs, sidl_io_Serializer outArgs)
    Select and execute a method by name

char*
bHYPRE_PCG__getURL ( bHYPRE_PCG self)
    Get the URL of the Implementation of this object (for RMI)

```

10.1.1

```

struct bHYPRE_PCG__object

```

Symbol "bHYPRE.PCG" (version 1.0.0)

10.1.2

```

int32_t
bHYPRE_PCG_SetCommunicator ( bHYPRE_PCG self,
bHYPRE_MPICommunicator mpi_comm)

```

Set the MPI Communicator. DEPRECATED, use Create:

10.1.3

```

int32_t
bHYPRE_PCG_SetOperator ( bHYPRE_PCG self, bHYPRE_Operator A)

```

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

10.1.4

```
int32_t bHYPRE_PCG_SetTolerance ( bHYPRE_PCG self, double tolerance)
```

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

10.1.5

```
int32_t
bHYPRE_PCG_SetMaxIterations ( bHYPRE_PCG self, int32_t
max_iterations)
```

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

10.1.6

```
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. 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

```
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. 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

GMRES Preconditioned Solver

Names

10.2.1	struct bHYPRE_GMRES__object <i>Symbol "bHYPRE"</i> extern C struct bHYPRE_GMRES__object* bHYPRE_GMRES__create void <i>Constructor function for the class</i> bHYPRE_GMRES bHYPRE_GMRES__createRemote (const char *, sidl_BaseInterface *_ex) <i>RMI constructor function for the class</i> bHYPRE_GMRES bHYPRE_GMRES__connect (const char *, sidl_BaseInterface *_ex) <i>RMI connector function for the class</i> bHYPRE_GMRES bHYPRE_GMRES_Create (bHYPRE_MPICommunicator mpi_comm, bHYPRE_Operator A) <i>Method: Create[]</i>	172
10.2.2	int32_t bHYPRE_GMRES_SetCommunicator (bHYPRE_GMRES self, bHYPRE_MPICommunicator mpi_comm) <i>Set the MPI Communicator</i> int32_t bHYPRE_GMRES_SetIntParameter (bHYPRE_GMRES self, const char* name, int32_t value) <i>Set the int parameter associated with name</i> int32_t bHYPRE_GMRES_SetDoubleParameter (bHYPRE_GMRES self, const char* name, double value) <i>Set the double parameter associated with name</i> int32_t bHYPRE_GMRES_SetStringParameter (bHYPRE_GMRES self, const char* name, const char* value) <i>Set the string parameter associated with name</i> int32_t bHYPRE_GMRES_SetIntArray1Parameter (bHYPRE_GMRES self, const char* name, int32_t* value, int32_t nvalues) <i>Set the int 1-D array parameter associated with name</i> int32_t	172

	bHYPRE_GMRES_SetIntArray2Parameter (bHYPRE_GMRES self, const char* name, struct sidl_int_array* value) <i>Set the int 2-D array parameter associated with name</i>	
	int32_t bHYPRE_GMRES_SetDoubleArray1Parameter (bHYPRE_GMRES self, const char* name, double* value, int32_t nvalues) <i>Set the double 1-D array parameter associated with name</i>	
	int32_t bHYPRE_GMRES_SetDoubleArray2Parameter (bHYPRE_GMRES self, const char* name, struct sidl_double_array* value) <i>Set the double 2-D array parameter associated with name</i>	
	int32_t bHYPRE_GMRES_GetIntValue (bHYPRE_GMRES self, const char* name, int32_t* value) <i>Set the int parameter associated with name</i>	
	int32_t bHYPRE_GMRES_GetDoubleValue (bHYPRE_GMRES self, const char* name, double* value) <i>Get the double parameter associated with name</i>	
	int32_t bHYPRE_GMRES_Setup (bHYPRE_GMRES self, bHYPRE_Vector b, bHYPRE_Vector x) <i>(Optional) Do any preprocessing that may be necessary in order to execute Apply</i>	
	int32_t bHYPRE_GMRES_Apply (bHYPRE_GMRES self, bHYPRE_Vector b, bHYPRE_Vector* x) <i>Apply the operator to b, returning x</i>	
	int32_t bHYPRE_GMRES_ApplyAdjoint (bHYPRE_GMRES self, bHYPRE_Vector b, bHYPRE_Vector* x) <i>Apply the adjoint of the operator to b, returning x</i>	
10.2.3	int32_t bHYPRE_GMRES_SetOperator (bHYPRE_GMRES self, bHYPRE_Operator A) <i>Set the operator for the linear system being solved</i>	172
10.2.4	int32_t bHYPRE_GMRES_SetTolerance (bHYPRE_GMRES self, double tolerance) <i>(Optional) Set the convergence tolerance</i>	173
10.2.5	int32_t bHYPRE_GMRES_SetMaxIterations (bHYPRE_GMRES self, int32_t max_iterations) <i>(Optional) Set maximum number of iterations</i>	173
10.2.6	int32_t	

	bHYPRE_GMRES_SetLogging (bHYPRE_GMRES self, int32_t level) (Optional) Set the logging level, specifying the degree of additional informational data to be accumulated	173
10.2.7	int32_t bHYPRE_GMRES_SetPrintLevel (bHYPRE_GMRES 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	173
	int32_t bHYPRE_GMRES_GetNumIterations (bHYPRE_GMRES self, int32_t* num_iterations) (Optional) Return the number of iterations taken	
	int32_t bHYPRE_GMRES_GetRelResidualNorm (bHYPRE_GMRES self, double* norm) (Optional) Return the norm of the relative residual	
	int32_t bHYPRE_GMRES_SetPreconditioner (bHYPRE_GMRES self, bHYPRE_Solver s) Set the preconditioner	
	int32_t bHYPRE_GMRES_GetPreconditioner (bHYPRE_GMRES self, bHYPRE_Solver* s) Method: GetPreconditioner[]	
	int32_t bHYPRE_GMRES_Clone (bHYPRE_GMRES self, bHYPRE_PreconditionedSolver* x) Method: Clone[]	
	struct bHYPRE_GMRES__object* bHYPRE_GMRES__cast void* obj Cast method for interface and class type conversions	
	void* bHYPRE_GMRES__cast2 (void* obj, const char* type) String cast method for interface and class type conversions	
	void bHYPRE_GMRES__exec (bHYPRE_GMRES self, const char* methodName, sidl_io_Deserializer inArgs, sidl_io_Serializer outArgs) Select and execute a method by name	
	char* bHYPRE_GMRES__getURL (bHYPRE_GMRES self) Get the URL of the Implementation of this object (for RMI)	

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

10.2.2

```
int32_t
bHYPRE_GMRES_SetCommunicator ( bHYPRE_GMRES self,
bHYPRE_MPICommunicator mpi_comm)
```

Set the MPI Communicator. DEPRECATED, use Create:

10.2.3

```
int32_t
bHYPRE_GMRES_SetOperator ( bHYPRE_GMRES self, bHYPRE_Operator
A)
```

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

10.2.4

```
int32_t
bHYPRE_GMRES_SetTolerance ( bHYPRE_GMRES self, double tolerance)
```

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

10.2.5

```
int32_t
bHYPRE_GMRES_SetMaxIterations ( bHYPRE_GMRES self, int32_t
max_iterations)
```

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

10.2.6

```
int32_t bHYPRE_GMRES_SetLogging ( bHYPRE_GMRES self, int32_t level)
```

(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

```
int32_t
bHYPRE_GMRES_SetPrintLevel ( bHYPRE_GMRES 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. 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

BiCGSTAB Preconditioned Solver

Names

10.3.1	<pre> struct bHYPRE_BiCGSTAB__object <i>Symbol "bHYPRE"</i> </pre>	177
	<pre> void <i>Constructor function for the class</i> </pre>	
	<pre> bHYPRE_BiCGSTAB bHYPRE_BiCGSTAB__createRemote (const char *, sidl_BaseInterface *_ex) <i>RMI constructor function for the class</i> </pre>	
	<pre> bHYPRE_BiCGSTAB bHYPRE_BiCGSTAB__connect (const char *, sidl_BaseInterface *_ex) <i>RMI connector function for the class</i> </pre>	
	<pre> bHYPRE_BiCGSTAB bHYPRE_BiCGSTAB_Create (bHYPRE_MPICommunicator mpi_comm, bHYPRE_Operator A) <i>Method: Create[]</i> </pre>	
10.3.2	<pre> int32_t bHYPRE_BiCGSTAB_SetCommunicator (bHYPRE_BiCGSTAB self, bHYPRE_MPICommunicator mpi_comm) <i>Set the MPI Communicator</i> </pre>	177
	<pre> int32_t bHYPRE_BiCGSTAB_SetIntParameter (bHYPRE_BiCGSTAB self, const char* name, int32_t value) <i>Set the int parameter associated with name</i> </pre>	
	<pre> int32_t bHYPRE_BiCGSTAB_SetDoubleParameter (bHYPRE_BiCGSTAB self, const char* name, double value) <i>Set the double parameter associated with name</i> </pre>	
	<pre> int32_t bHYPRE_BiCGSTAB_SetStringParameter (bHYPRE_BiCGSTAB self, const char* name, const char* value) <i>Set the string parameter associated with name</i> </pre>	
	<pre> int32_t </pre>	

```

bHYPRE_BiCGSTAB_SetIntArray1Parameter ( bHYPRE_BiCGSTAB
                                         self, const char* name,
                                         int32_t* value,
                                         int32_t nvalues)

    Set the int 1-D array parameter associated with name

int32_t
bHYPRE_BiCGSTAB_SetIntArray2Parameter ( bHYPRE_BiCGSTAB
                                         self, const char* name,
                                         struct sidl_int__array*
                                         value)

    Set the int 2-D array parameter associated with name

int32_t
bHYPRE_BiCGSTAB_SetDoubleArray1Parameter (
                                         bHYPRE_BiCGSTAB
                                         self,
                                         const char* name,
                                         double* value,
                                         int32_t nvalues)

    Set the double 1-D array parameter associated with name

int32_t
bHYPRE_BiCGSTAB_SetDoubleArray2Parameter (
                                         bHYPRE_BiCGSTAB
                                         self,
                                         const char* name,
                                         struct
                                         sidl_double__array*
                                         value)

    Set the double 2-D array parameter associated with name

int32_t
bHYPRE_BiCGSTAB_GetIntValue ( bHYPRE_BiCGSTAB self,
                                const char* name, int32_t* value)

    Set the int parameter associated with name

int32_t
bHYPRE_BiCGSTAB_GetDoubleValue ( bHYPRE_BiCGSTAB self,
                                    const char* name, double* value)

    Get the double parameter associated with name

int32_t
bHYPRE_BiCGSTAB_Setup ( bHYPRE_BiCGSTAB self,
                          bHYPRE_Vector b, bHYPRE_Vector x)

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

int32_t
bHYPRE_BiCGSTAB_Apply ( bHYPRE_BiCGSTAB self,
                          bHYPRE_Vector b, bHYPRE_Vector* x)

    Apply the operator to b, returning x

int32_t

```

	bHYPRE_BiCGSTAB_ApplyAdjoint (bHYPRE_BiCGSTAB self, bHYPRE_Vector b, bHYPRE_Vector* x) <i>Apply the adjoint of the operator to b, returning x</i>	
10.3.3	int32_t bHYPRE_BiCGSTAB_SetOperator (bHYPRE_BiCGSTAB self, bHYPRE_Operator A) <i>Set the operator for the linear system being solved</i>	178
10.3.4	int32_t bHYPRE_BiCGSTAB_SetTolerance (bHYPRE_BiCGSTAB self, double tolerance) <i>(Optional) Set the convergence tolerance</i>	178
10.3.5	int32_t bHYPRE_BiCGSTAB_SetMaxIterations (bHYPRE_BiCGSTAB self, int32_t max_iterations) <i>(Optional) Set maximum number of iterations</i>	178
10.3.6	int32_t bHYPRE_BiCGSTAB_SetLogging (bHYPRE_BiCGSTAB self, int32_t level) <i>(Optional) Set the logging level, specifying the degree of additional informational data to be accumulated</i>	178
10.3.7	int32_t bHYPRE_BiCGSTAB_SetPrintLevel (bHYPRE_BiCGSTAB self, int32_t level) <i>(Optional) Set the print level, specifying the degree of informational data to be printed either to the screen or to a file</i>	179
	int32_t bHYPRE_BiCGSTAB_GetNumIterations (bHYPRE_BiCGSTAB self, int32_t* num_iterations) <i>(Optional) Return the number of iterations taken</i>	
	int32_t bHYPRE_BiCGSTAB_GetRelResidualNorm (bHYPRE_BiCGSTAB self, double* norm) <i>(Optional) Return the norm of the relative residual</i>	
	int32_t bHYPRE_BiCGSTAB_SetPreconditioner (bHYPRE_BiCGSTAB self, bHYPRE_Solver s) <i>Set the preconditioner</i>	
	int32_t bHYPRE_BiCGSTAB_GetPreconditioner (bHYPRE_BiCGSTAB self, bHYPRE_Solver* s) <i>Method: GetPreconditioner[]</i>	
	int32_t bHYPRE_BiCGSTAB_Clone (bHYPRE_BiCGSTAB self, bHYPRE_PreconditionedSolver* x) <i>Method: Clone[]</i>	
	struct bHYPRE_BiCGSTAB__object* bHYPRE_BiCGSTAB__cast void* obj	

Cast method for interface and class type conversions

void*

bHYPRE_BiCGSTAB__cast2 (void* obj, const char* type)

String cast method for interface and class type conversions

void

bHYPRE_BiCGSTAB__exec (bHYPRE_BiCGSTAB self,
const char* methodName,
sidl_io_Deserializer inArgs,
sidl_io_Serializer outArgs)

Select and execute a method by name

char*

bHYPRE_BiCGSTAB__getURL (bHYPRE_BiCGSTAB self)

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

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

int32_t

bHYPRE_BiCGSTAB_SetCommunicator (bHYPRE_BiCGSTAB self,
bHYPRE_MPICommunicator mpi_comm)

Set the MPI Communicator. DEPRECATED, use Create:

10.3.3

```
int32_t  
bHYPRE_BiCGSTAB_SetOperator ( bHYPRE_BiCGSTAB self,  
bHYPRE_Operator A)
```

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

10.3.4

```
int32_t  
bHYPRE_BiCGSTAB_SetTolerance ( bHYPRE_BiCGSTAB self, double  
tolerance)
```

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

10.3.5

```
int32_t  
bHYPRE_BiCGSTAB_SetMaxIterations ( bHYPRE_BiCGSTAB self, int32_t  
max_iterations)
```

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

10.3.6

```
int32_t  
bHYPRE_BiCGSTAB_SetLogging ( bHYPRE_BiCGSTAB self, int32_t level)
```

(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

```
int32_t
bHYPRE_BiCGSTAB_SetPrintLevel ( bHYPRE_BiCGSTAB 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. 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

CGNR Preconditioned Solver

Names

10.4.1	struct bHYPRE_CGNR__object <i>Symbol "bHYPRE"</i>	182
	extern C struct bHYPRE_CGNR__object* bHYPRE_CGNR__create void <i>Constructor function for the class</i>	
	bHYPRE_CGNR bHYPRE_CGNR__createRemote (const char *, sidl_BaseInterface *_ex) <i>RMI constructor function for the class</i>	
	bHYPRE_CGNR bHYPRE_CGNR__connect (const char *, sidl_BaseInterface *_ex) <i>RMI connector function for the class</i>	
	bHYPRE_CGNR bHYPRE_CGNR_Create (bHYPRE_MPICommunicator mpi_comm, bHYPRE_Operator A) <i>Method: Create[]</i>	
10.4.2	int32_t bHYPRE_CGNR_SetCommunicator (bHYPRE_CGNR self, bHYPRE_MPICommunicator mpi_comm) <i>Set the MPI Communicator</i>	182
	int32_t	

```

bHYPRE_CGNR_SetIntParameter ( bHYPRE_CGNR self,
                                const char* name,  int32_t value)
    Set the int parameter associated with name

int32_t
bHYPRE_CGNR_SetDoubleParameter ( bHYPRE_CGNR self,
                                    const char* name,  double value)
    Set the double parameter associated with name

int32_t
bHYPRE_CGNR_SetStringParameter ( bHYPRE_CGNR self,
                                    const char* name,
                                    const char* value)
    Set the string parameter associated with name

int32_t
bHYPRE_CGNR_SetIntArray1Parameter ( bHYPRE_CGNR self,
                                        const char* name,
                                        int32_t* value,  int32_t nvalues)
    Set the int 1-D array parameter associated with name

int32_t
bHYPRE_CGNR_SetIntArray2Parameter ( bHYPRE_CGNR self,
                                        const char* name,
                                        struct sidl_int_array* value)
    Set the int 2-D array parameter associated with name

int32_t
bHYPRE_CGNR_SetDoubleArray1Parameter ( bHYPRE_CGNR self,
                                            const char* name,
                                            double* value,
                                            int32_t nvalues)
    Set the double 1-D array parameter associated with name

int32_t
bHYPRE_CGNR_SetDoubleArray2Parameter ( bHYPRE_CGNR self,
                                            const char* name,  struct
                                            sidl_double_array* value)
    Set the double 2-D array parameter associated with name

int32_t
bHYPRE_CGNR_GetIntValue ( bHYPRE_CGNR self,  const char* name,
                            int32_t* value)
    Set the int parameter associated with name

int32_t
bHYPRE_CGNR_GetDoubleValue ( bHYPRE_CGNR self,
                                const char* name,  double* value)
    Get the double parameter associated with name

int32_t
bHYPRE_CGNR_Setup ( bHYPRE_CGNR self,  bHYPRE_Vector b,
                    bHYPRE_Vector x)
    (Optional) Do any preprocessing that may be necessary in order to execute
    Apply

int32_t

```

	bHYPRE_CGNR_Apply (bHYPRE_CGNR self, bHYPRE_Vector b, bHYPRE_Vector* x) <i>Apply the operator to b, returning x</i>	
	int32_t bHYPRE_CGNR_ApplyAdjoint (bHYPRE_CGNR self, bHYPRE_Vector b, bHYPRE_Vector* x) <i>Apply the adjoint of the operator to b, returning x</i>	
10.4.3	int32_t bHYPRE_CGNR_SetOperator (bHYPRE_CGNR self, bHYPRE_Operator A) <i>Set the operator for the linear system being solved</i>	183
10.4.4	int32_t bHYPRE_CGNR_SetTolerance (bHYPRE_CGNR self, double tolerance) <i>(Optional) Set the convergence tolerance</i>	183
10.4.5	int32_t bHYPRE_CGNR_SetMaxIterations (bHYPRE_CGNR self, int32_t max_iterations) <i>(Optional) Set maximum number of iterations</i>	183
10.4.6	int32_t bHYPRE_CGNR_SetLogging (bHYPRE_CGNR self, int32_t level) <i>(Optional) Set the logging level, specifying the degree of additional informa- tional data to be accumulated</i>	183
10.4.7	int32_t bHYPRE_CGNR_SetPrintLevel (bHYPRE_CGNR self, int32_t level) <i>(Optional) Set the print level, specifying the degree of informational data to be printed either to the screen or to a file</i>	184
	int32_t bHYPRE_CGNR_GetNumIterations (bHYPRE_CGNR self, int32_t* num_iterations) <i>(Optional) Return the number of iterations taken</i>	
	int32_t bHYPRE_CGNR_GetRelResidualNorm (bHYPRE_CGNR self, double* norm) <i>(Optional) Return the norm of the relative residual</i>	
	int32_t bHYPRE_CGNR_SetPreconditioner (bHYPRE_CGNR self, bHYPRE_Solver s) <i>Set the preconditioner</i>	
	int32_t bHYPRE_CGNR_GetPreconditioner (bHYPRE_CGNR self, bHYPRE_Solver* s) <i>Method: GetPreconditioner[]</i>	
	int32_t bHYPRE_CGNR_Clone (bHYPRE_CGNR self, bHYPRE_PreconditionedSolver* x) <i>Method: Clone[]</i>	
	struct bHYPRE_CGNR__object* bHYPRE_CGNR__cast void* obj	

Cast method for interface and class type conversions

void*

bHYPRE_CGNR__cast2 (void* obj, const char* type)

String cast method for interface and class type conversions

void

bHYPRE_CGNR__exec (bHYPRE_CGNR self, const char* methodName,
sidl_io_Deserializer inArgs, sidl_io_Serializer outArgs)

Select and execute a method by name

char*

bHYPRE_CGNR__getURL (bHYPRE_CGNR self)

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

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

int32_t

bHYPRE_CGNR_SetCommunicator (bHYPRE_CGNR self,
bHYPRE_MPICommunicator mpi_comm)

Set the MPI Communicator. DEPRECATED, use Create:

10.4.3

```
int32_t
bHYPRE_CGNR_SetOperator ( bHYPRE_CGNR self, bHYPRE_Operator A)
```

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

10.4.4

```
int32_t
bHYPRE_CGNR_SetTolerance ( bHYPRE_CGNR self, double tolerance)
```

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

10.4.5

```
int32_t
bHYPRE_CGNR_SetMaxIterations ( bHYPRE_CGNR self, int32_t
max_iterations)
```

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

10.4.6

```
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. 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

```
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. Does nothing by default (level=0). Other levels (if any) are implementation-specific. Must be called before **Setup** and **Apply**. DEPRECATED use **SetIntParameter**

11

Other

Names

11.1	MPI Communicator	185
------	-------------------------	-----

11.1

MPI Communicator

Names

11.1.1	struct bHYPRE_MPICommunicator__object <i>Symbol "bHYPRE"</i>	186
	void <i>Constructor function for the class</i>	
	bHYPRE_MPICommunicator bHYPRE_MPICommunicator__createRemote (const char *, sidl_BaseInterface *_ex) <i>RMI constructor function for the class</i>	
	bHYPRE_MPICommunicator bHYPRE_MPICommunicator__connect (const char *, sidl_BaseInterface *_ex) <i>RMI connector function for the class</i>	
	bHYPRE_MPICommunicator bHYPRE_MPICommunicator_CreateC (void* mpi_comm) <i>Method: CreateC[]</i>	
	bHYPRE_MPICommunicator bHYPRE_MPICommunicator_CreateF (void* mpi_comm) <i>Method: CreateF[]</i>	
	obj <i>Cast method for interface and class type conversions</i>	
	void* bHYPRE_MPICommunicator__cast2 (void* obj, const char* type) <i>String cast method for interface and class type conversions</i>	
	void	

```
bHYPRE_MPICommunicator__exec ( bHYPRE_MPICommunicator self,
                                const char* methodName,
                                sidl_io_Deserializer inArgs,
                                sidl_io_Serializer outArgs)
```

Select and execute a method by name

char*

```
bHYPRE_MPICommunicator__getURL ( bHYPRE_MPICommunicator
                                   self)
```

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

11.1.1

```
struct bHYPRE_MPICommunicator__object
```

Symbol "bHYPRE.MPICommunicator" (version 1.0.0)

MPICommunicator class two Create functions: use CreateC if called from C code, CreateF if called from Fortran code

12

Struct Grid, etc.

Names

12.1	Struct Grid	
	187
12.2	Struct Stencil	
	189

12.1

Struct Grid

Names

12.1.1	struct bHYPRE_StructGrid__object	
	<i>Symbol "bHYPRE"</i>	188
	void	
	<i>Constructor function for the class</i>	
	bHYPRE_StructGrid	
	bHYPRE_StructGrid__createRemote (const char *, sidl_BaseInterface *_ex)	
	<i>RMI constructor function for the class</i>	
	bHYPRE_StructGrid	
	bHYPRE_StructGrid__connect (const char *, sidl_BaseInterface *_ex)	
	<i>RMI connector function for the class</i>	
	bHYPRE_StructGrid	
	bHYPRE_StructGrid_Create (bHYPRE_MPICommunicator mpi_comm,	
	int32_t dim)	
	<i>Method: Create[]</i>	
12.1.2	int32_t	
	bHYPRE_StructGrid_SetCommunicator (bHYPRE_StructGrid self,	
	bHYPRE_MPICommunicator	
	mpi_comm)	
	<i>Set the MPI Communicator</i>	189
	int32_t	
	bHYPRE_StructGrid_SetDimension (bHYPRE_StructGrid self,	
	int32_t dim)	
	<i>Method: SetDimension[]</i>	
	int32_t	

```

bHYPRE_StructGrid_SetExtents ( bHYPRE_StructGrid self,
                                int32_t* ilower, int32_t* iupper,
                                int32_t dim)
    Method: SetExtents[]

int32_t
bHYPRE_StructGrid_SetPeriodic ( bHYPRE_StructGrid self,
                                int32_t* periodic, int32_t dim)
    Method: SetPeriodic[]

int32_t
bHYPRE_StructGrid_SetNumGhost ( bHYPRE_StructGrid self,
                                int32_t* num_ghost, int32_t dim2)
    Method: SetNumGhost[]

int32_t
bHYPRE_StructGrid_Assemble ( bHYPRE_StructGrid self)
    Method: Assemble[]

struct bHYPRE_StructGrid__object* bHYPRE_StructGrid__cast void* obj
    Cast method for interface and class type conversions

void*
bHYPRE_StructGrid__cast2 ( void* obj, const char* type)
    String cast method for interface and class type conversions

void
bHYPRE_StructGrid__exec ( bHYPRE_StructGrid self,
                           const char* methodName,
                           sidl_io_Deserializer inArgs,
                           sidl_io_Serializer outArgs)
    Select and execute a method by name

char*
bHYPRE_StructGrid__getURL ( bHYPRE_StructGrid self)
    Get the URL of the Implementation of this object (for RMI)

```

12.1.1

```
struct bHYPRE_StructGrid__object
```

Symbol "bHYPRE.StructGrid" (version 1.0.0)

Define a structured grid class.

12.1.2

```
int32_t
bHYPRE_StructGrid_SetCommunicator ( bHYPRE_StructGrid self,
bHYPRE_MPICommunicator mpi_comm)
```

Set the MPI Communicator. DEPRECATED, use Create:

12.2

Struct Stencil

Names

```
12.2.1      struct bHYPRE_StructStencil__object
              Symbol "bHYPRE" ..... 190

void
    Constructor function for the class

bHYPRE_StructStencil
bHYPRE_StructStencil__createRemote (const char *,
                                     sidl_BaseInterface *_ex)
    RMI constructor function for the class

bHYPRE_StructStencil
bHYPRE_StructStencil__connect (const char *, sidl_BaseInterface *_ex)
    RMI connector function for the class

bHYPRE_StructStencil
bHYPRE_StructStencil_Create ( int32_t ndim,  int32_t size)
    Method: Create[]

int32_t
bHYPRE_StructStencil_SetDimension ( bHYPRE_StructStencil self,
                                     int32_t dim)
    Method: SetDimension[]

int32_t
bHYPRE_StructStencil_SetSize ( bHYPRE_StructStencil self,  int32_t size)
    Method: SetSize[]

int32_t
bHYPRE_StructStencil_SetElement ( bHYPRE_StructStencil self,
                                     int32_t index,  int32_t* offset,
                                     int32_t dim)
    Method: SetElement[]

struct bHYPRE_StructStencil__object* bHYPRE_StructStencil__cast void* obj
```

Cast method for interface and class type conversions

void*

bHYPRE_StructStencil__cast2 (void* obj, const char* type)

String cast method for interface and class type conversions

void

bHYPRE_StructStencil__exec (bHYPRE_StructStencil self,
const char* methodName,
sidl_io_Deserializer inArgs,
sidl_io_Serializer outArgs)

Select and execute a method by name

char*

bHYPRE_StructStencil__getURL (bHYPRE_StructStencil self)

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

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.

Semi-Structured Grid, etc.

Names

13.1	Semi-Structured Graph	191
13.2	Semi-Structured Grid	194
13.3	Semi-Structured Stencil	198
	Semi-Structured Variable	

Semi-Structured Graph

Names

13.1.1	struct bHYPRE_SStructGraph__object <i>Symbol "bHYPRE"</i>	193
	void <i>Constructor function for the class</i>	
	bHYPRE_SStructGraph bHYPRE_SStructGraph__createRemote (const char *, sidl_BaseInterface *_ex) <i>RMI constructor function for the class</i>	
	bHYPRE_SStructGraph bHYPRE_SStructGraph__connect (const char *, sidl_BaseInterface *_ex) <i>RMI connector function for the class</i>	
	bHYPRE_SStructGraph bHYPRE_SStructGraph_Create (bHYPRE_MPICommunicator mpi_comm, bHYPRE_SStructGrid grid) <i>Method: Create[]</i>	
13.1.2	int32_t bHYPRE_SStructGraph_SetCommGrid (bHYPRE_SStructGraph self, bHYPRE_MPICommunicator mpi_comm, bHYPRE_SStructGrid grid) <i>Set the grid and communicator</i>	193
	int32_t	

	bHYPRE_SStructGraph_SetStencil (bHYPRE_SStructGraph self, int32_t part, int32_t var, bHYPRE_SStructStencil stencil) <i>Set the stencil for a variable on a structured part of the grid</i>	
13.1.3	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) <i>Add a non-stencil graph entry at a particular index</i>	193
	int32_t bHYPRE_SStructGraph_SetObjectType (bHYPRE_SStructGraph self, int32_t type) <i>Method: SetObjectType[]</i>	
13.1.4	int32_t bHYPRE_SStructGraph_SetCommunicator (bHYPRE_SStructGraph self, bHYPRE_MPICommunicator mpi_comm) <i>Set the MPI Communicator</i>	194
	int32_t bHYPRE_SStructGraph_Initialize (bHYPRE_SStructGraph self) <i>Prepare an object for setting coefficient values, whether for the first time or subsequently</i>	
13.1.5	int32_t bHYPRE_SStructGraph_Assemble (bHYPRE_SStructGraph self) <i>Finalize the construction of an object before using, either for the first time or on subsequent uses</i>	194
	struct bHYPRE_SStructGraph__object* bHYPRE_SStructGraph__cast void* obj <i>Cast method for interface and class type conversions</i>	
	void* bHYPRE_SStructGraph__cast2 (void* obj, const char* type) <i>String cast method for interface and class type conversions</i>	
	void bHYPRE_SStructGraph__exec (bHYPRE_SStructGraph self, const char* methodName, sidl_io_Deserializer inArgs, sidl_io_Serializer outArgs) <i>Select and execute a method by name</i>	
	char* bHYPRE_SStructGraph__getURL (bHYPRE_SStructGraph self) <i>Get the URL of the Implementation of this object (for RMI)</i>	

13.1.1

```
struct bHYPRE_SStructGraph__object
```

Symbol "bHYPRE.SStructGraph" (version 1.0.0)

The semi-structured grid graph class.

13.1.2

```
int32_t
bHYPRE_SStructGraph_SetCommGrid ( bHYPRE_SStructGraph self,
bHYPRE_MPICommunicator mpi_comm, bHYPRE_SStructGrid grid)
```

Set the grid and communicator. DEPRECATED, use Create:

13.1.3

```
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)
```

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.

13.1.4

```
int32_t
bHYPRE_SStructGraph_SetCommunicator ( bHYPRE_SStructGraph self,
bHYPRE_MPICommunicator mpi_comm)
```

Set the MPI Communicator. DEPRECATED, Use Create()

13.1.5

```
int32_t bHYPRE_SStructGraph_Assemble ( bHYPRE_SStructGraph self)
```

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.

13.2**Semi-Structured Grid****Names**

13.2.1	struct bHYPRE_SStructGrid__object <i>Symbol "bHYPRE"</i>	196
	void <i>Constructor function for the class</i>	
	bHYPRE_SStructGrid bHYPRE_SStructGrid__createRemote (const char *, sidl_BaseInterface *_ex) <i>RMI constructor function for the class</i>	
	bHYPRE_SStructGrid bHYPRE_SStructGrid__connect (const char *, sidl_BaseInterface *_ex) <i>RMI connector function for the class</i>	
	bHYPRE_SStructGrid	

	bHYPRE_SStructGrid_Create (bHYPRE_MPICommunicator mpi_comm, int32_t ndim, int32_t nparts) <i>Set the number of dimensions ndim and the number of structured parts nparts</i>	
	int32_t bHYPRE_SStructGrid_SetNumDimParts (bHYPRE_SStructGrid self, int32_t ndim, int32_t nparts) <i>Method: SetNumDimParts[]</i>	
	int32_t bHYPRE_SStructGrid_SetCommunicator (bHYPRE_SStructGrid self, bHYPRE_MPICommunicator mpi_comm) <i>Method: SetCommunicator[]</i>	
	int32_t bHYPRE_SStructGrid_SetExtents (bHYPRE_SStructGrid self, int32_t part, int32_t* ilower, int32_t* iupper, int32_t dim) <i>Set the extents for a box on a structured part of the grid</i>	
13.2.2	int32_t bHYPRE_SStructGrid_SetVariable (bHYPRE_SStructGrid self, int32_t part, int32_t var, int32_t nvars, enum bHYPRE_SStructVariable__enum vartype) <i>Describe the variables that live on a structured part of the grid</i>	196
13.2.3	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) <i>Describe additional variables that live at a particular index</i>	197
13.2.4	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) <i>Describe how regions just outside of a part relate to other parts</i>	197
13.2.5	int32_t bHYPRE_SStructGrid_AddUnstructuredPart (bHYPRE_SStructGrid self, int32_t ilower, int32_t iupper) <i>Add an unstructured part to the grid</i>	198
	int32_t	

```

bHYPRE_SStructGrid_SetPeriodic ( bHYPRE_SStructGrid self,
                                   int32_t part,  int32_t* periodic,
                                   int32_t dim)
    (Optional) Set periodic for a particular part

int32_t
bHYPRE_SStructGrid_SetNumGhost ( bHYPRE_SStructGrid self,
                                   int32_t* num_ghost,  int32_t dim2)
    Setting ghost in the sgrids

int32_t
bHYPRE_SStructGrid_Assemble ( bHYPRE_SStructGrid self)
    Method: Assemble[]

struct bHYPRE_SStructGrid__object* bHYPRE_SStructGrid__cast void* obj
    Cast method for interface and class type conversions

void*
bHYPRE_SStructGrid__cast2 ( void* obj,  const char* type)
    String cast method for interface and class type conversions

void
bHYPRE_SStructGrid__exec ( bHYPRE_SStructGrid self,
                            const char* methodName,
                            sidl_io_Deserializer inArgs,
                            sidl_io_Serializer outArgs)
    Select and execute a method by name

char*
bHYPRE_SStructGrid__getURL ( bHYPRE_SStructGrid self)
    Get the URL of the Implementation of this object (for RMI)

```

13.2.1

```
struct bHYPRE_SStructGrid__object
```

Symbol "bHYPRE.SStructGrid" (version 1.0.0)

The semi-structured grid class.

13.2.2

```

int32_t
bHYPRE_SStructGrid_SetVariable ( bHYPRE_SStructGrid self,  int32_t part,
int32_t var,  int32_t nvars,  enum bHYPRE_SStructVariable__enum vartype)

```

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.

13.2.3

```
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)
```

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.

13.2.4

```
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)
```

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.

```
int32_t
bHYPRE_SStructGrid_AddUnstructuredPart ( bHYPRE_SStructGrid self,
int32_t ilower, int32_t iupper)
```

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.

13.3

Semi-Structured Stencil

Names

```

13.3.1 struct bHYPRE_SStructStencil__object
        Symbol "bHYPRE_SStructStencil__object" ..... 199

void
        Constructor function for the class

bHYPRE_SStructStencil
bHYPRE_SStructStencil__createRemote (const char *,
        sidl_BaseInterface *_ex)
        RMI constructor function for the class

bHYPRE_SStructStencil
bHYPRE_SStructStencil__connect (const char *, sidl_BaseInterface *_ex)
        RMI connector function for the class

bHYPRE_SStructStencil
bHYPRE_SStructStencil__Create ( int32_t ndim,  int32_t size)
        Method: Create[]

13.3.2 int32_t
bHYPRE_SStructStencil_SetNumDimSize ( bHYPRE_SStructStencil self,
        int32_t ndim,  int32_t size)
        Set the number of spatial dimensions and stencil entries ..... 199

int32_t
bHYPRE_SStructStencil_SetEntry ( bHYPRE_SStructStencil self,
        int32_t entry,  int32_t* offset,
        int32_t dim,  int32_t var)
        Set a stencil entry

obj

```

Cast method for interface and class type conversions

void*

bHYPRE_SStructStencil__cast2 (void* obj, const char* type)

String cast method for interface and class type conversions

void

bHYPRE_SStructStencil__exec (bHYPRE_SStructStencil self,
const char* methodName,
sidl_io_Deserializer inArgs,
sidl_io_Serializer outArgs)

Select and execute a method by name

char*

bHYPRE_SStructStencil__getURL (bHYPRE_SStructStencil self)

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

13.3.1

```
struct bHYPRE_SStructStencil_object
```

Symbol "bHYPRE.SStructStencil" (version 1.0.0)

The semi-structured grid stencil class.

13.3.2

int32_t

bHYPRE_SStructStencil_SetNumDimSize (bHYPRE_SStructStencil self,
int32_t ndim, int32_t size)

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

Class Graph