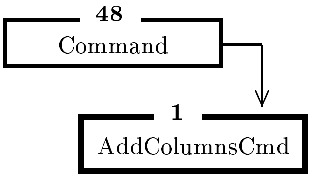
Contents

1 class **AddColumnsCmd** : public Command

add columns to table command.

Inheritance



Public Members

	AddColumnsCmd (Table *t, TableViewer *v, int numberOfColumns, int columnPosition)
virtual	**~AddColumnsCmd** ()
void	**Execute** ()
void	**UnExecute** ()
void	**ReExecute** ()
const char*	**GetName** () const

Private Members

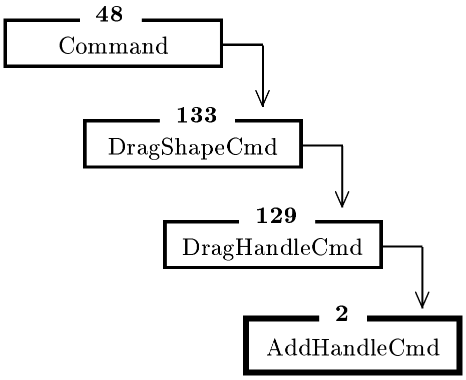
int	**columnPosition**
int	**numberOfColumns**
List <CellColumn *> *	**columns**
TableViewer*	**tviewer**

2

class **AddHandleCmd** : public DragHandleCmd

command class for adding line handles.

Inheritance



Public Members

	AddHandleCmd (Line *s, int x, int y, unsigned which)
void	**UnExecute** ()
void	**Executable** ()
void	**DrawOutLine** (const Point *delta)
const char*	**GetName** () const

Private Members

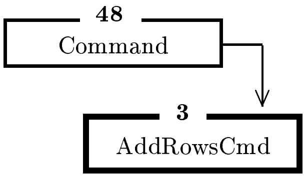
void	**CalcPoints** ()
Point	**CalcHandle** ()
Point	**newPoint**
unsigned	**addPosition**

3

class **AddRowsCmd** : public Command

add rows to table command.

Inheritance



Public Members

AddRowsCmd (Table *t, TableViewer *v, int numberOfRows, int rowPosition)

virtual **~AddRowsCmd** ()

void **Execute** ()

void **UnExecute** ()

void **ReExecute** ()

const char* **GetName** () const

Private Members

int **rowPosition**

int **numberOfRows**

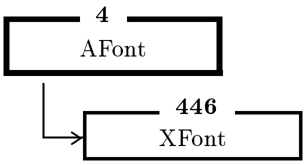
List <CellRow *> * **rows**

TableViewer* **tviewer**

4 class **AFont**

An abstract font class.

Inheritance



Public Members

static const int **HELVETICA** *Helvetica font family.*

static const int **TIMESROMAN** *Times Roman font family.*

static const int **COURIER** *Courier (fixed width) font family.*

static const int **NEWCENTURYSCHLBK** *New century schoolbook font family.*

static const int **SYMBOL** *Symbol font family.*

static const int **PLAIN** *Plain style constant. Can be mixed with other style constants.*

static const int **BOLD** *Bold style constant. Can be mixed with other style constants.*

static const int **ITALIC** *Italicized style constant. Can be mixed with other style constants.*

AFont (const string* foundry, int name, int style, int size, bool scalable)

		Creates new font with the specified name, style and point size.
int	**GetFamily** ()	*Gets the family name of the font.*
int	**GetStyle** ()	*Gets the style of the font.*
int	**GetSize** ()	*Gets the point size of the font.*
bool	**IsScalable** ()	*Is the font scalable.*
const string*	**GetFoundry** ()	
void	**SetScalable** (bool b)	
bool	**IsUnderlined** ()	*Is the font underlined.*
void	**SetUnderlined** (bool b)	*Set/Unset font underlined option*
double	**GetFactor** ()	*The factor with which a scalable font is scaled.*
void	**SetFactor** (double d)	
bool	**IsPlain** ()	*Returns true if the font is plain.*
bool	**IsBold** ()	*Returns true if the font is bold.*
bool	**IsItalic** ()	*Returns true if the font is italic.*

Private Members

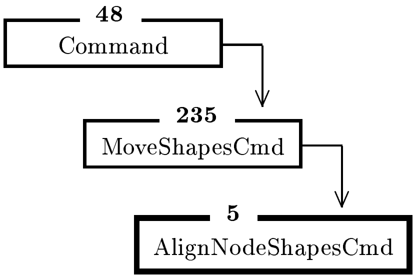
double	**factor**	*factor for scalable fonts.*
string	**foundry**	*the font foundry*
int	**family**	*The family of this font.*
int	**style**	*The style of the font (sum of constants PLAIN, BOLD, or ITALIC).*
int	**size**	*The point size of this font.*
bool	**scalable**	*The font is scalable.*
bool	**underlined**	*The font is underlined.*

5

class **AlignNodeShapesCmd** : public MoveShapesCmd

command class for aligning selected node shapes in view.

Inheritance



Public Members

AlignNodeShapesCmd (ShapeView *view, List<GShape *> *s, NodeAlign::Type alignType)

void **Execute** ()

const char* **GetName** () const

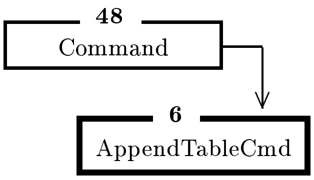
Private Members

NodeAlign::Type **alignType**

6 class **AppendTableCmd** : public Command

add rows to table command.

Inheritance



Public Members

	AppendTableCmd (Table *t, TableViewer *v)
virtual	**~AppendTableCmd** ()
void	**Execute** ()
void	**ReExecute** ()
void	**UnExecute** ()
const char*	**GetName** () const

Private Members

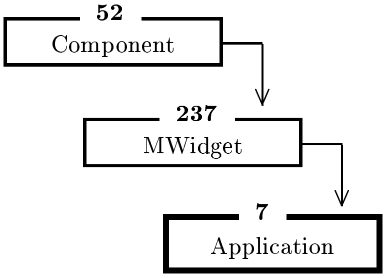
Command*	**command1**
Command*	**command2**

7

class **Application** : public MWidget

generic window-based application class. See: MWidget.

Inheritance



Public Members

	Application (const char *name)	
virtual	**~Application** ()	
void	**Manage** ()	*manage application's top-level windows*
void	**Unmanage** ()	*unmanage application's top-level windows*
void	**Iconify** ()	*iconify application's top-level windows*
void	**Quit** ()	*close all top-level windows and quit application.*
void	**CloseWindow** (MainWindow *m)	*close window and quit application when it is the last.*
Display*	**GetDisplay** ()	
XtAppContext	**GetAppContext** ()	
const char*	**GetAppClass** ()	
const char*		

GetClassName ()

bool **Initialized** ()

MessageLogDialog*
GetMessageLogDialog ()

PrinterOptions*
GetPrinterOptions ()

friend int **main** (int argc, char** argv)
Allow main and MainWindow to access protected member functions

void **CreatePrivateColormap** ()

List <Color *> *
GetColors ()

static void
XtErrorHandler (char *msg)
Show Xt and X error in error message log.

Protected Members

Config* **GetConfig** ()

virtual void
Initialize (int argc, char** argv)

virtual void
HandleEvents ()

Private Members

Display* **display** *Support commonly needed data structures as a convenience*

Config* **config** *The configuration object.*

XtAppContext
appContext *Application context, needed by Xt.*

char* **appClass** *Class name of this application*

MessageLogDialog*
messageLogDialog

PrinterOptions*

printerOptions

bool **initialized**

bool **privateCmap**

List <MainWindow*> **windows** *Top-level windows in the program*

void **RegisterWindow** (MainWindow *)
Registering toplevel windows

void **UnregisterWindow** (MainWindow *)
Unregistering toplevel windows

List <Color *> * **colors**

void **LoadColorFile** ()

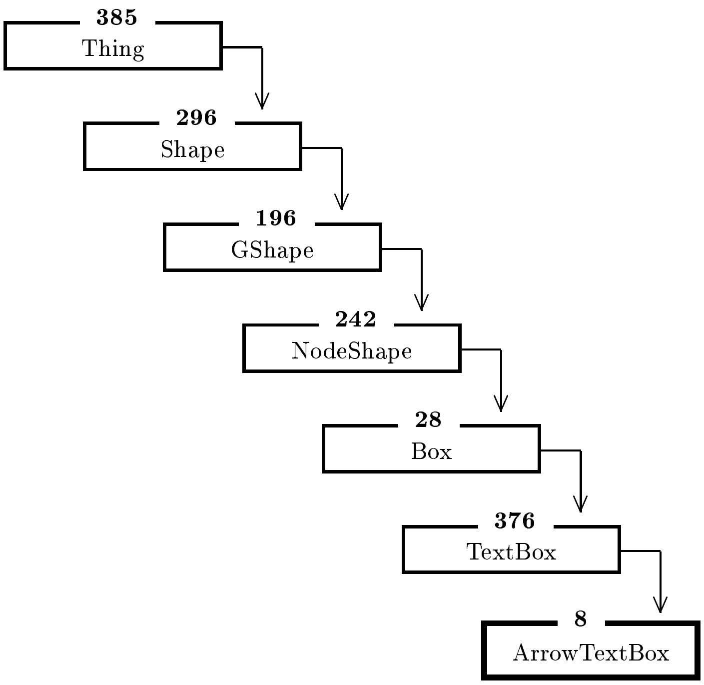
static const char **COLORFILE** []

8

class **ArrowTextBox** : public TextBox

text box class having a small downwards pointing arrow on top.

Inheritance



Public Members

	ArrowTextBox (ShapeView *v, Grafport *g, int x, int y, int w = MIN_WIDTH, int h = MIN_HEIGHT)
	ArrowTextBox (const NodeShape &c)
Shape*	**Clone** ()
int	**GetClassType** () const
int	**GetTopMost** () const

Protected Members

void	**DrawShape** ()	*draws text shape plus little arrow.*
LineEnd	**end**	
static const int	**ARROWLEN**	*length of arrow-box' arrows.*

9 class **Assoc**

binary association struct.

Public Members

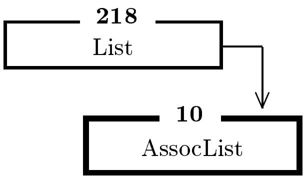
	Assoc (void* o, void* c, int t)
	Assoc (void* o)
void	**Add** (void* c, int t)
void*	**original**
void*	**copy**
int	**type**

10

class **AssocList** : public List<Assoc*>

list of Assoc class.

Inheritance



Public Members

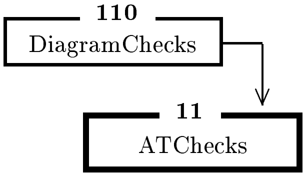
void	**Put** (Assoc* a)
Assoc*	**Get** (Assoc*)

11

class **ATChecks** : public DiagramChecks

implements generic checks for soft constraints on AT diagrams.

Inheritance



Public Members

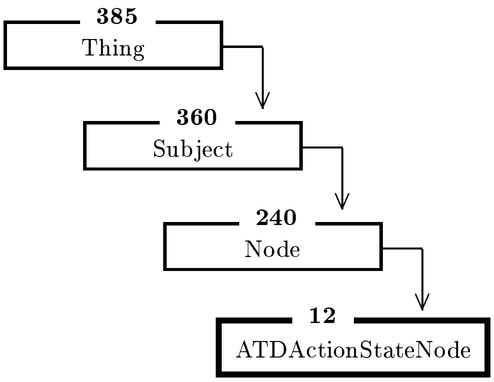
	ATChecks (Diagram *d, Graph *g)
unsigned	**CheckUnreachableStates** (string &chkbuf)
unsigned	**CheckEmptyEvents** (string &)
unsigned	**CheckEmptyActions** (string &)
unsigned	**CheckNoActions** (string &)
unsigned	**CheckDoubleEvents** (string &)

12

class **ATDActionStateNode** : public Node

action state node class.

Inheritance



Public Members

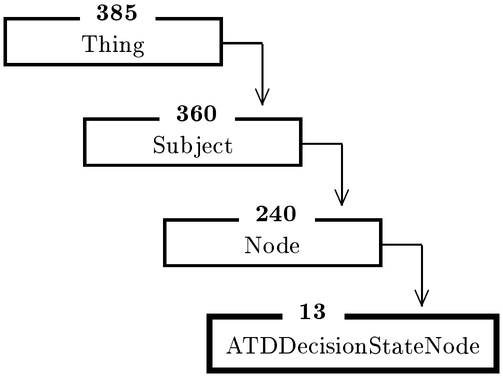
	ATDActionStateNode (ATGraph *g)
int	**GetClassType** () const
Subject*	**Clone** ()
Subject::NameErrType	**SetName** (const string *name)

13

class **ATDDecisionStateNode** : public Node

decision state (ATD) node class.

Inheritance



Public Members

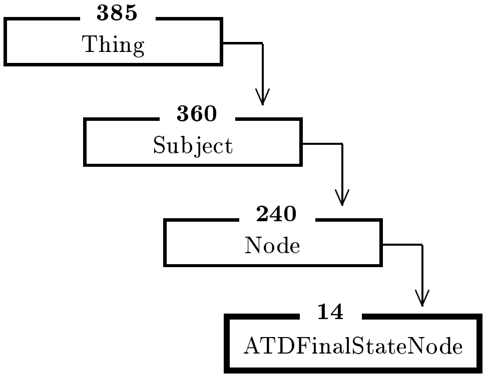
	ATDDecisionStateNode (ATGraph *g)
int	**GetClassType** () const
Subject*	**Clone** ()

14

class **ATDFinalStateNode** : public Node

final state (ATD) node class.

Inheritance



Public Members

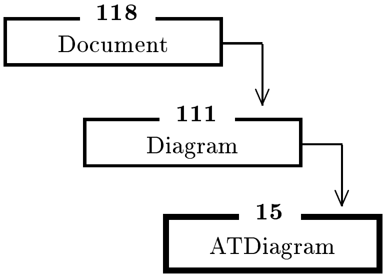
	ATDFinalStateNode (ATGraph *g)
int	**GetClassType** () const
Subject*	**Clone** ()

15

class **ATDiagram** : public Diagram

activity diagram class.

Inheritance



Public Members

ATDiagram (Config *, ATWindow *, ATViewer *, ATGraph *)

virtual **~ATDiagram** ()

Node* **CreateNode** ()

Edge* **CreateEdge** (Subject *n1, Subject *n2)

NodeShape* **CreateNodeShape** (Node *node, int x, int y)

Line* **CreateLine** (Edge *edge, GShape *fromShape, GShape *toShape, List<Point *> *line)

void **UpdateNodeType** (int n)

void **UpdateEdgeType** (int n)

void **CheckDocument** ()

static const int **DOT_WIDTH**

Protected Members

Thing*	**CreateThing** (int classNr)
bool	**HasIndexNode** (int code)
bool	**HasIndexShape** (int code)

Private Members

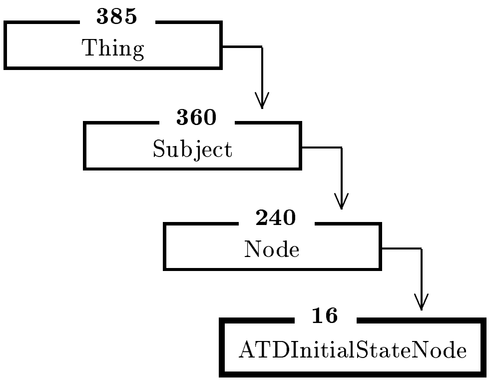
ATChecks*
atChecks

16

class **ATDInitialStateNode** : public Node

initial state (ATD) node class.

Inheritance



Public Members

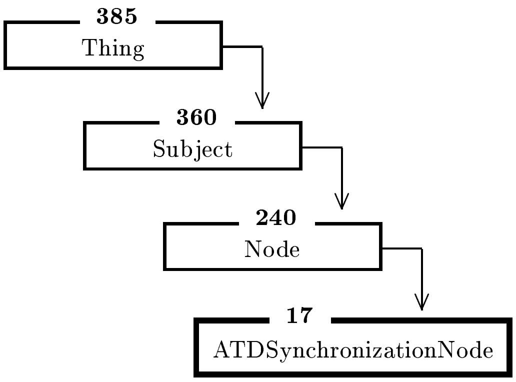
	ATDInitialStateNode (ATGraph *g)
int	**GetClassType** () const
Subject*	**Clone** ()

17

class **ATDSynchronizationNode** : public Node

synchronization (ATD) node class.

Inheritance



Public Members

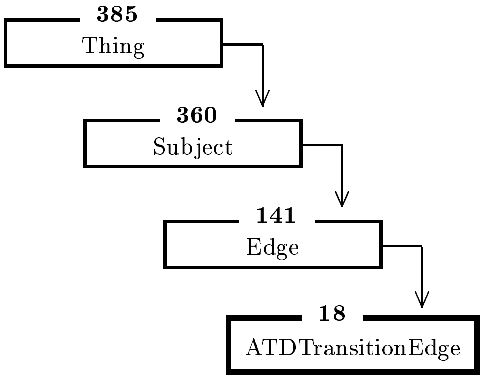
	ATDSynchronizationNode (ATGraph *g)
int	**GetClassType** () const
Subject*	**Clone** ()

18

class **ATDTransitionEdge** : public Edge

transition (ATD) edge class.

Inheritance



Public Members

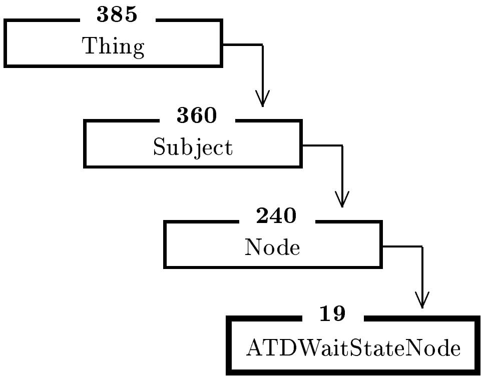
	ATDTransitionEdge (ATGraph *g, Subject *s1, Subject *s2)
	ATDTransitionEdge (const ATDTransitionEdge &t)
int	**GetClassType** () const
Subject*	**Clone** ()
Subject::NameErrType	**SetName** (const string *s)

19

class **ATDWaitStateNode** : public Node

action state (ATD) node class.

Inheritance



Public Members

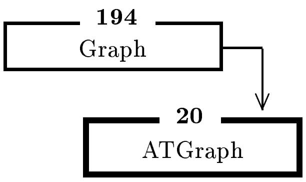
	ATDWaitStateNode (ATGraph *g)
int	**GetClassType** () const
Subject*	**Clone** ()
Subject::NameErrType	**SetName** (const string *name)

20

class **ATGraph** : public Graph

generic diagram graph class.

Inheritance



Public Members

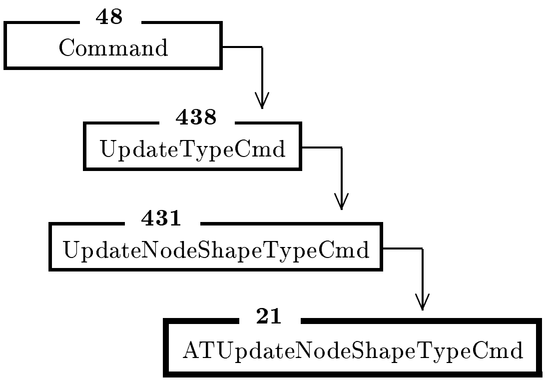
	ATGraph ()
void	**InitConnections** ()

21

class **ATUpdateNodeShapeTypeCmd** : public UpdateNodeShapeTypeCmd

command for changing node shape types in a TATD

Inheritance



Public Members

ATUpdateNodeShapeTypeCmd (ShapeView *v, List<int *> *fromTypes, int toType)

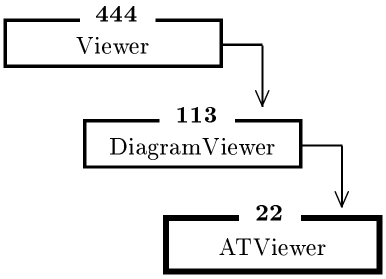
virtual Thing* **Transform** (Thing *thing)

22

class **ATViewer** : public DiagramViewer

generic diagram viewer class.

Inheritance



Public Members

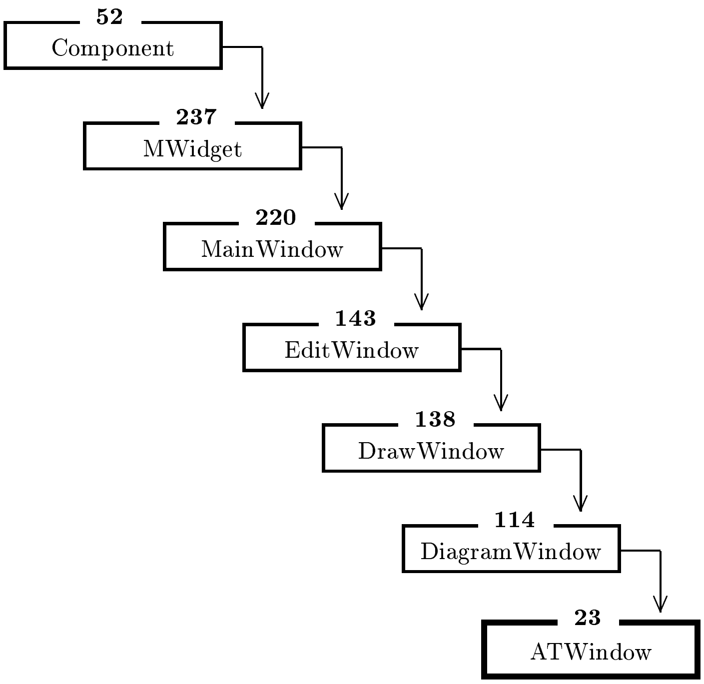
	ATViewer (Config *c, ATWindow *w)	
void	**UpdateNodeShapeType** (int)	*change node shape type of selected types.*

23

class **ATWindow** : public DiagramWindow

generic diagram edit window class.

Inheritance



Public Members

ATWindow (const char *name)

Private Members

void **InitMenuItems** ()

int **GetTool** ()

DiagramViewer* **CreateViewer** (Config *c)

Graph* **CreateGraph** ()

Diagram* **CreateDiagram** (Config *c, DiagramViewer *v, Graph *g)

void **DetermineIcons** ()

static const char* **AT_NODE_NAMES** []

static const Bitmap **AT_NODE_ICONS** []

static const char* **AT_EDGE_NAMES** []

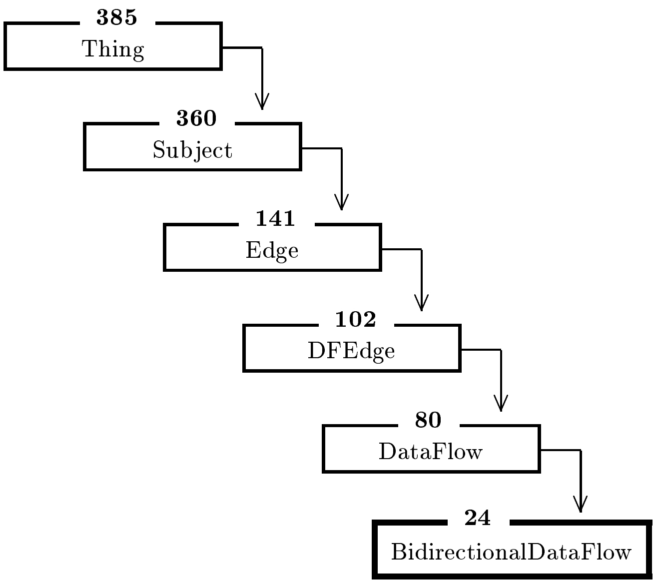
static const Bitmap **AT_EDGE_ICONS** []

24

class **BidirectionalDataFlow** : public DataFlow

bidirectional data flow edge class.

Inheritance



Public Members

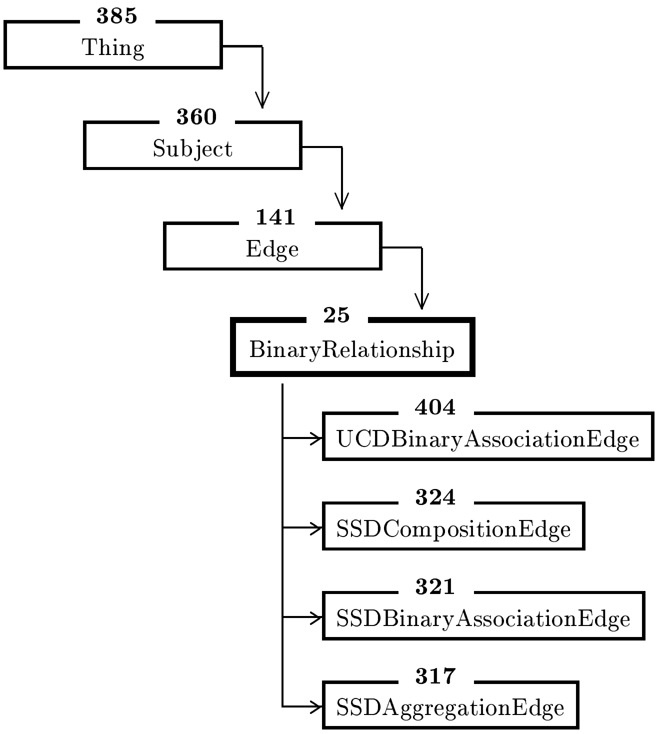
	BidirectionalDataFlow (DFGraph *g, Subject *n1, Subject *n2)
int	**GetClassType** () const
Subject*	**Clone** ()
Subject::NameErrType	**SetName** (const string *name)

25

class **BinaryRelationship** : public Edge

binary relationship (ER) edge class.

Inheritance



Public Members

	BinaryRelationship (Graph *g, Subject *n1, Subject *n2)
Subject*	**Clone** ()
int	**GetClassType** () const
Subject::NameErrType	**SetName** (const string *name)
bool	**SetConstraint1** (const string *s)

const string*
GetConstraint1 () const

bool **SetConstraint2** (const string *s)

const string*
GetConstraint2 () const

bool **SetRoleName1** (const string *s)

const string*
GetRoleName1 () const

bool **SetRoleName2** (const string *s)

const string*
GetRoleName2 () const

void **WriteMembers** (OutputFile *f)

bool **ReadMembers** (InputFile *f, double format)

Private Members

string **constraint1**

string **constraint2**

string **roleName1**

string **roleName2**

virtual bool
SetConstraint (string *cons, const string *name)

virtual bool
SetRoleName (string *cons, const string *name)

bool **ParseCardConstraint** (const string *s)

26

class **Bitmap**

datatype for storing an X bitmap.

Public Members

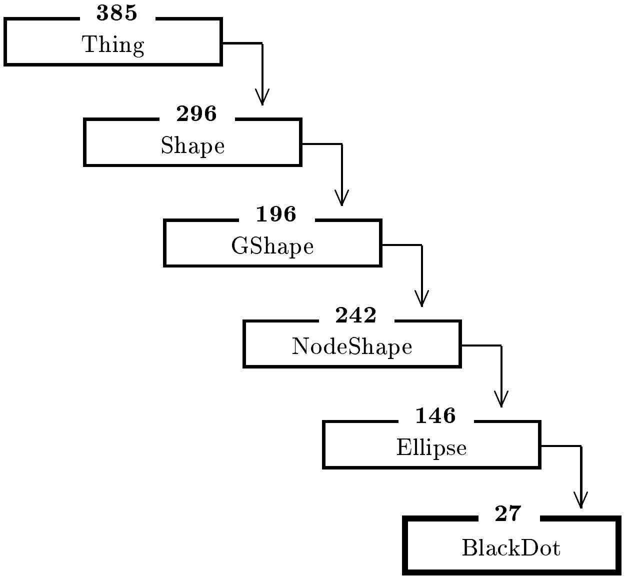
int	**xbm_width**	*width in pixels.*
int	**xbm_height**	*height in pixels.*
unsigned char*	**xbm_bits**	*bit string.*
Pixmap	**CreatePixmap** (Widget w, Display *display) const	*returns a color pixmap from the bitmap.*
Pixmap	**GetBitmap** (Widget w, Display *display) const	*returns the bitmap as pixmap.*

27

class **BlackDot** : public Ellipse

Mini ellipse class, drawn as a black dot.

Inheritance



Public Members

static const int
BLACK_DOT_WIDTH

BlackDot (ShapeView *v, Grafport *g, int x, int y, int w = BLACK_DOT_WIDTH, int h = BLACK_DOT_WIDTH)

BlackDot (const NodeShape &c)

Shape* **Clone** ()

int **GetClassType** () const

Private Members

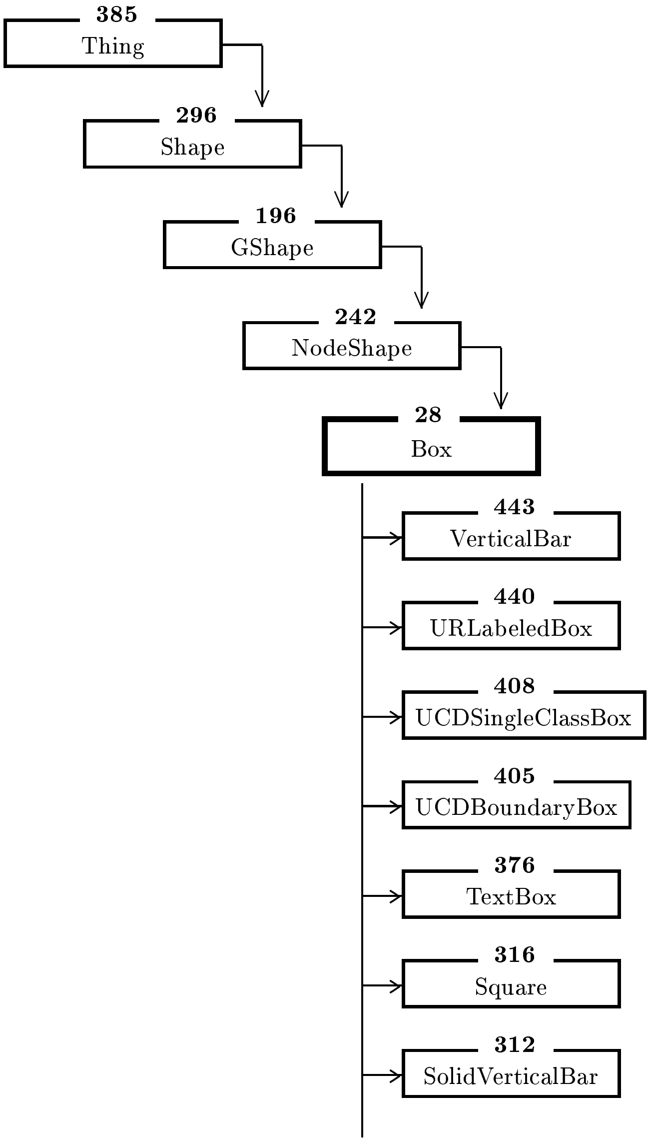
void **Init** ()

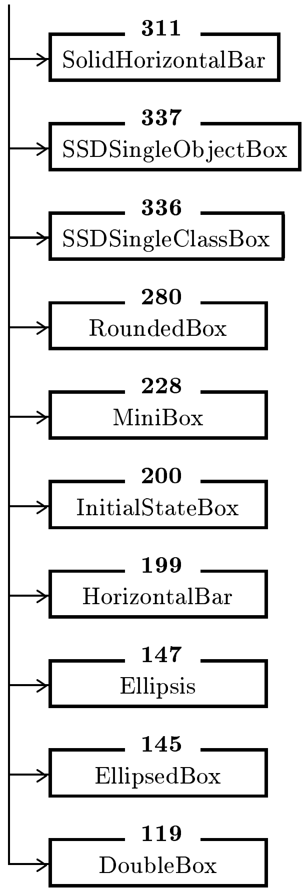
28

class **Box** : public NodeShape

Rectangular shape class.

Inheritance





Public Members

static const int
BOX_WIDTH *default width*

static const int
BOX_HEIGHT *default height*

Box (ShapeView *v, Grafport *g, int x, int y,
int w = BOX_WIDTH,
int h = BOX_HEIGHT)

Box (const NodeShape &)

Shape* **Clone** ()

void **DrawOutLine** (const Point *center)

draw box-form dotted outline snaps to the grid.

Point **GiveSnp** (Point to) const
See Shape::

Point **GiveOrthoSnp** (Point to) const
See Shape::

Point **GiveSnp** (Point p, int order, int count, int &code)
order and count are for multiple edges. code for horz or vert point.

int **GetClassType** () const

static const int
BOX_HANDLES
default nr. of handles.

Protected Members

void **DrawShape** () *draws a rectangle*

29

class **Buffer**

shape and subject cut-paste buffer class.

Public Members

	Buffer ()	
	~Buffer ()	
void	**Put** (Subjects *, Shapes *)	*Copy lists of subjects and shapes and hold them in the buffer.*
Subjects*	**GetSubjectsCopy** ()	*Copy subjects in the buffer and return them in a list.*
Shapes*	**GetShapesCopy** ()	*Copy shapes in the buffer and return them in a list.*
Subjects*	**GetSubjects** () const	
Shapes*	**GetShapes** () const	
bool	**IsEmpty** () const	*Returns if buffer is empty.*
void	**SetUser** (Command *c)	*Which command uses this buffer.*
Command*	**GetUser** ()	

Private Members

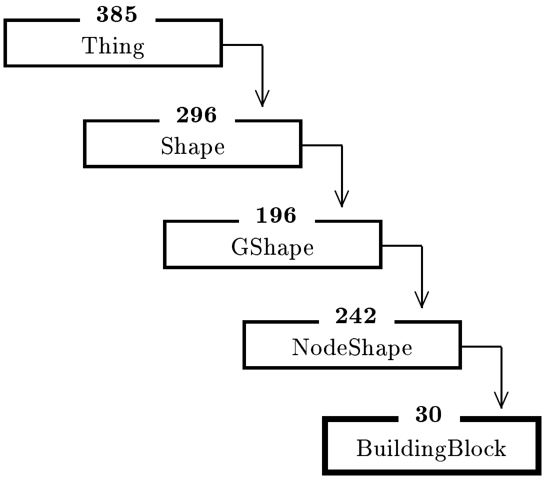
Subjects*	**subjects**
Shapes*	**shapes**
Command*	**user**

30

class **BuildingBlock** : public NodeShape

building block shape class.

Inheritance



Public Members

static const int
BUILDING_BLOCK_WIDTH

static const int
BUILDING_BLOCK_HEIGHT

BuildingBlock (ShapeView *v, Grafport *g, int x, int y, int w = BUILDING_BLOCK_WIDTH, int h = BUILDING_BLOCK_HEIGHT)

BuildingBlock (const BuildingBlock &)

BuildingBlock (const NodeShape &)

virtual **~BuildingBlock** ()

Shape* **Clone** ()

void **DrawOutLine** (const Point *center)

draws dotted building block

Point **GiveSnp** (Point to) const

Point **GiveSnp** (Point p, int order, int count, int &code)

bool **ContainsPt** (int x, int y)

int **GetClassType** () const

static const int
BUILDING_BLOCK_HANDLES

Protected Members

void **DrawShape** () *draws a building block*

void **CalcPoints** (int x, int y)

const Point*
GetPoints () const

Private Members

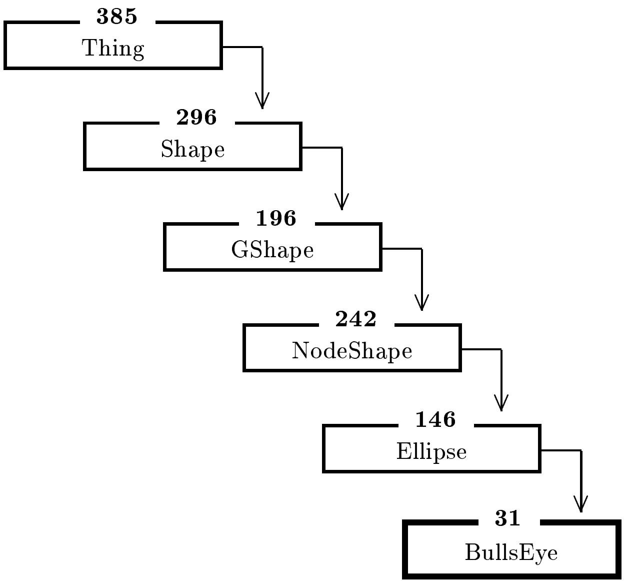
Point* **points**

31

class **BullsEye** : public Ellipse

Mini ellipse class, drawn as an encircled black dot.

Inheritance



Public Members

static const int
DOT_WIDTH

static const int
DOT_HEIGHT

static const int
CIRCLE_WIDTH

static const int
CIRCLE_HEIGHT

BullsEye (ShapeView *v, Grafport *g, int x, int y,
int w = CIRCLE_WIDTH,
int h = CIRCLE_HEIGHT)

BullsEye (const NodeShape &c)

Shape*	**Clone** ()	
int	**GetClassType** () const	

Protected Members

void	**DrawShape** ()	*draws black dot inside the ellipse.*

Private Members

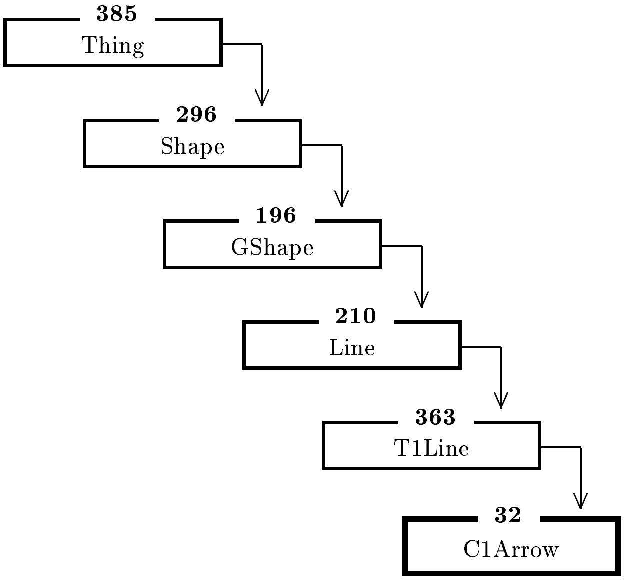
void	**Init** ()	

32

class **C1Arrow** : public T1Line

arrow having card. constraint at 'from' side (ERD and CRD).

Inheritance



Public Members

	C1Arrow (ShapeView *v, Grafport *g, GShape *n1, GShape *n2, List<Point *> *line, bool Curved)
	C1Arrow (ShapeView *v, Grafport *g, GShape *n1, GShape *n2, bool Curved)
Shape*	**Clone** ()
bool	**SetAssocSubject** (AssocList *)
void	**SetTextShape** ()

Private Members

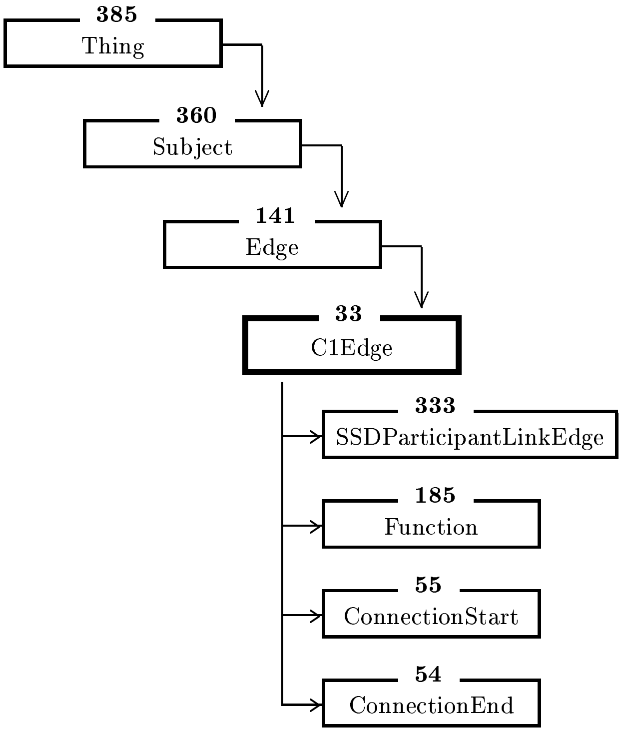
void **Initialize** ()

33

class **C1Edge** : public Edge

Abstract edge class having one extra 'constraint' string.

Inheritance



Public Members

	C1Edge (Graph *g, Subject *n1, Subject *n2)
Subject*	**Clone** ()
virtual bool	**SetConstraint** (const string *s)
const string*	**GetConstraint** () const
void	**WriteMembers** (OutputFile *f)

bool **ReadMembers** (InputFile *f, double format)

Private Members

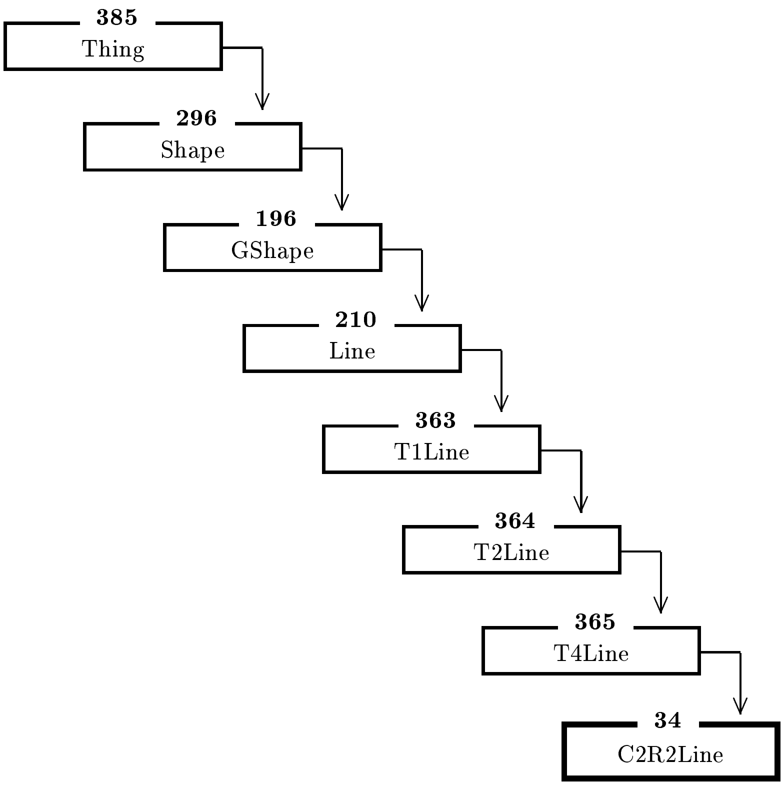
string **constraint**

34

class **C2R2Line** : public T4Line

line having card. constraints and role names at both sides (ERD and CRD).

Inheritance



Public Members

C2R2Line (ShapeView *v, Grafport *g, GShape *n1, GShape *n2, List<Point *> *line, bool Curved)

C2R2Line (ShapeView *v, Grafport *g, GShape *n1, GShape *n2, bool Curved)

Shape*	**Clone** ()
int	**GetClassType** () const
bool	**SetAssocSubject** (AssocList *)
void	**SetTextShape** ()
TextShape*	**ChooseTextShape** (int x, int y)
TextShape*	**HitTextShape** (int x, int y)
void	**WriteMembers** (OutputFile *f)
bool	**ReadMembers** (InputFile *f, double format)

Protected Members

void	**DrawShape** ()

Private Members

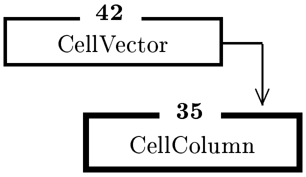
void	**InitTextShapes** ()

35

class **CellColumn** : public CellVector

(table) column class.

Inheritance



Public Members

	CellColumn (TableViewer *v, unsigned columnNumber, int width)
friend int	**Compare** (CellColumn *c1, CellColumn *c2)
void	**UpdateWidth** (int width) *make each cell width wide.*
void	**UpdatePosition** (int delta) *move each cell delta left/right.*
void	**UpdateHeight** (unsigned rownr) *resize all cells from index rownr and lower.*
LinePiece*	**HitLinePiece** (int x, int y) *if (x, y) is on a linepiece, return it.*
void	**AddCell** (Cell *c, bool redraw=True) *append cell to column.*
void	**DeleteCell** (Cell *c, bool redraw=True) *delete cell from column.*
void	**InsertCell** (Cell *c, unsigned n, bool redraw=True) *insert cell in column at position n.*
bool	**ContainsPt** (int x, int y) const

void	**Write** (OutputFile *f)	*write attributes (but not cell contents) to file.*
void	**WritePartial** (OutputFile *f, int i, List<int> *rowNumbers)	*write part of column to file (only given rows).*
bool	**Read** (InputFile *f, unsigned fromRow, double format)	*read attributes back from file starting at row n.*
int	**GetMinimalWidth** ()	
int	**GetWidth** () const	
int	**GetMarginWidth** () const	
void	**UpdateTextPositions** ()	
void	**CalcLabelPositions** ()	

Private Members

int	**width**	*width of a column = width of its cells which span 1 column.*

36

class **Cell**

(table) cell class.

Public Members

	Cell (Grafport *g, XFont *ft, CellRow *row, CellColumn *column, int x, int y, int wd, int ht)	
virtual	**~Cell** ()	
friend int	**operator==** (const Cell &c1, const Cell &c2)	*alphabetical text comparison.*
void	**Draw** ()	*draw the cell.*
void	**Undraw** ()	*erase the cell.*
void	**UpdatePosition** (const Point *pt)	*reposition and redraw*
void	**UpdateSize** (int wd, int ht)	*resize and redraw.*
void	**UpdateSelect** (bool)	*(de)select and redraw.*
void	**UpdateFont** (XFont *ft)	*update text font and redraw.*
void	**UpdateTextPosition** ()	*recalculate text position and redraw.*
void	**SetTextSize** ()	*recalculate text size.*
void	**SetPosition** (const Point *pt)	*reposition cell + text*
void	**SetSize** (int wd, int ht)	*resize cell.*
void	**SetGrafport** (Grafport *g)	*change grafport*
bool	**ContainsPt** (int x, int y) const	*true iff (x, y) is inside cell.*
void	**Write** (OutputFile *f)	

write cell attributes to output stream (file).

bool **Read** (InputFile *f, double format)
read cell attributes from input stream (file).

void **UpdateText** (const string *)
update cell text and redraw.

void **ClearText** () *make cell text empty.*

bool **HasText** (const string *s) const
true iff celltext is 's'.

bool **HasText** (const string *s, bool c1, bool c2)
true iff celltext contains 's' (c1: case sensitive).

void **SetRow** (CellRow *r)

void **SetColumn** (CellColumn *c)

CellRow* **GetRow** () const

CellColumn* **GetColumn** () const

const Point* **GetPosition** () const

const Point* **GetTopLeft** () const

int **GetWidth** () const

int **GetHeight** () const

bool **IsSelected** () const

bool **IsEditable** () const

void **SetEditable** (bool b)

bool **IsVisible** () const

void **SetVisible** (bool b)

CellText* **GetCellText** () const

const string* **GetText** () const

void **SetFont** (XFont *ft)

XFont* **GetFont** () const

void **SetXLFD** (const char *descr)

int	**GetTextWidth** () const	
int	**GetTextHeight** () const	
TextAlign::Type	**GetColumnAlignment** () const	
void	**CalcCellTextPosition** ()	*set cell text pos. given row/column alignment + margin width/height.*
void	**SetAnnotation** (const string *s)	
const string*	**GetAnnotation** () const	

Private Members

CellRow*	**row**	*row of cell*
CellColumn*	**column**	*column of cell.*
Point	**position**	*the center position of the cell on the drawing area.*
Point	**topLeft**	*the top-left position of the cell on the drawing area.*
int	**width**	*the width of the cell*
int	**height**	*the height of the cell*
int	**spanRows**	*how many rows are spanned by the cell.*
int	**spanColumns**	*how many columns are spanned by the cell.*
bool	**selected**	*is cell selected? (represented by highlighting the cell).*
bool	**visible**	*is cell currently visible?*
bool	**editable**	*can the cell text be edited?*
string	**annotation**	*cell annotation.*
CellText*	**cellText**	*the text label inside the cell.*
Grafport*	**grafport**	
void	**PositiveCoord** ()	*change position so that shape has only positive coordinates.*

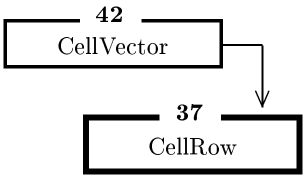
void	**DrawSelect** ()	*draw selection rectangle into cell.*

37

class **CellRow** : public CellVector

(table) row class.

Inheritance



Public Members

CellRow (TableViewer *v, unsigned rowNumber, int height)

friend int **Compare** (CellRow *r1, CellRow *r2)

void **UpdateHeight** (int height)
each cell will be this height.

void **UpdatePosition** (int delta)
each cell is moved delta up/down

void **UpdateWidth** (unsigned celnr)
resize cells and lines from index cellnr and higher.

Cell* **HitCell** (int x, int y)
if (x, y) is in a cell, return it.

LinePiece* **HitLinePiece** (int x, int y)
if (x, y) is on a line piece, return it.

void **AddCell** (Cell *c, bool redraw=True)
append cell to row.

void **DeleteCell** (Cell *c, bool redraw=True)
delete cell from row.

void **InsertCell** (Cell *c, unsigned p, bool redraw=True)
insert cell in row, at position p.

void **Write** (OutputFile *f)

write attributes (including cell contents to file.

void **WritePartial** (OutputFile *f, int i, List<int> *columnNumbers)
write part of row to file (only given columns).

bool **Read** (InputFile *f, unsigned fromColumn, double format)
read attributes back from file starting at position n.

bool **ContainsPt** (int x, int y) const

int **GetMinimalHeight** ()

int **GetHeight** () const

int **GetMarginHeight** () const

void **UpdateTextPositions** ()
recalculate text positions in cells.

void **SetTextSizes** () *recalculate text size in cells.*

void **CalcLabelPositions** ()

Private Members

int **height** *the height of a row is the height of its cells which span 1 row.*

38

class **CellSelection**

list of selected cells.

Public Members

	CellSelection ()
virtual	**~CellSelection** ()
void	**Draw** ()
void	**Undraw** ()
void	**SelectCell** (Cell *)
void	**SelectCells** (List<Cell *> *)
void	**DeselectCell** (Cell *)
void	**DeselectCells** (List<Cell *> *)
void	**DeselectAll** ()
int	**NrSelected** () const
Cell*	**FirstSelected** ()
Cell*	**NextSelected** ()

Private Members

List <Cell *> *
cells *inverse of cell.Selected.*

39

class **CellTextBuffer**

cut-paste buffer class for cell texts.

Public Members

	CellTextBuffer ()	
virtual	**~CellTextBuffer** ()	
const Point	**GetRcTopLeft** () const	*topleft row-column position.*
const Point	**GetTopLeft** () const	*topleft of the paste box.*
const Point	**GetBottomRight** () const	*bottomright of the paste box.*
void	**PutTexts** (List<CellText *> *c)	*put _copy_ of c in buffer.*
void	**PutPositions** (List<Point *> *p)	*put row/column positions in the buffer.*
List <CellText *> *	**GetTexts** ()	*return _copy_ of c.*
List <Point *> *	**GetPositions** () const	
void	**UndoPut** ()	*move prevCelltexts to cell-texts.*
bool	**IsEmpty** ()	*is buffer empty ?*

Private Members

Point	**rcTopLeft**
Point	**topLeft**
Point	**bottomRight**
List <CellText *> *	

cellTexts

List <CellText *> *
prevCellTexts *for undo.*

List <Point *> *
rcPositions

List <Point *> *
prevRcPositions

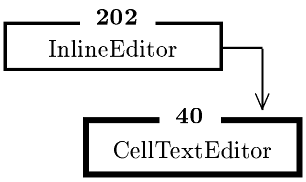
void **CalcSizeBuffer** ()

40

class **CellTextEditor** : public InlineEditor

cell text editor class.

Inheritance



Public Members

	CellTextEditor (TableViewer *v)	
void	**Start** (CellText *)	*start inline editing, initialized by cell text.*
void	**Stop** ()	
void	**Abort** ()	
void	**StartWindow** (CellText *)	*start window editing, initialized by cell text.*
CellText*	**GetCellText** () const	
bool	**IsUnderlined** () const	

Private Members

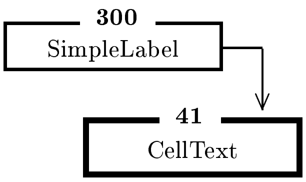
CellText*	**cellText**	*the cell-text that will be updated when editing is finished.*
void	**UpdateText** ()	

41

class **CellText** : public SimpleLabel

cell text class.

Inheritance



Public Members

CellText (Grafport *g, XFont *ft, Cell *parent)
creation.

CellText (const CellText &text)
creation + initialization.

void **Draw** ()

void **DrawAligned** (TextAlign::Type a)
draw cell text.

void **UndrawAligned** (TextAlign::Type a)
erase cell cell.

void **UpdatePosition** (const Point *)
update position and redraw.

void **UpdateText** (const string *)
update text and redraw.

void **UpdateFont** (XFont *)
update font and redraw.

virtual bool
IsUnderlined () const

void **SetUnderlined** (bool b)

void **UpdateUnderlined** (bool b)

Cell* **GetParent** () const

void	**Write** (OutputFile *f)	
bool	**Read** (InputFile *f, double format)	

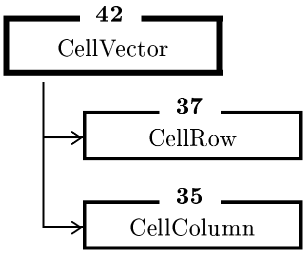
Private Members

Cell*	**parent**	*cell that contains this text (can be 0) = inverse of cell.celltext.*
bool	**underlined**	*is the cell text underlined*

42 class **CellVector**

abstract base class of table rows and columns.

Inheritance



Public Members

	CellVector (TableViewer *v, unsigned number)	*creation.*
friend int	**operator==** (const CellVector &c1, const CellVector &c2)	*-1, 0, 1 iff c1.number ==, <, > c2.number*
friend int	**Compare** (CellVector *c1, CellVector *c2)	*-1, 0, 1 iff c1.number ==, <, > c2.number*
void	**Draw** ()	*draw cell vector.*
void	**Undraw** ()	*erase cell vector.*
void	**DrawCells** ()	
void	**UndrawCells** ()	
void	**DrawLines** ()	
void	**UndrawLines** ()	
void	**DrawLabels** ()	
void	**UndrawLabels** ()	
void	**SetGrafport** (Grafport *g)	
void	**SetVisible** (bool b)	

Make all cells and lines (in)visible.

TableViewer*
GetViewer () const

virtual void
UpdatePosition (int delta)

virtual LinePiece*
HitLinePiece (int x, int y)

bool **HasCell** (Cell *c)

RowColumnLabel*
HitLabel (int x, int y) const

virtual void
AddCell (Cell *c, bool redraw=False)
append cell to column.

virtual void
DeleteCell (Cell *c, bool redraw=False)
delete cell from column.

virtual void
InsertCell (Cell *c, unsigned n,
bool redraw=False)
insert cell in column.

virtual bool
ContainsPt (int x, int y) const

virtual void
Write (OutputFile *f)

virtual void
WritePartial (OutputFile *f, int i, List<int> *nrs)

virtual bool
Read (InputFile *f, unsigned from = 0,
double format = 1.0)

TextAlign::Type
GetAlignment () const

void **SetAlignment** (TextAlign::Type a)

void **UpdateAlignment** (TextAlign::Type a)

virtual void
UpdateNumber (unsigned nr)

virtual void

	SetLabelsVisible (bool s)	
void	**SetNumber** (unsigned nr)	
Cell*	**FirstCell** ()	*return first cell, 0 if cells is empty.*
Cell*	**NextCell** ()	*return next cell, 0 if cells is empty.*
Cell*	**NthCell** (unsigned n)	*return n-th cell, 0 if n-th cell does not exist.*
LinePiece*	**FirstLine** ()	*return first line, 0 if lines is empty.*
LinePiece*	**NextLine** ()	*return next line, 0 if lines is empty.*
LinePiece*	**NthLine** (unsigned n)	*return n-th line, 0 if n-th line does not exist.*
unsigned	**GetNumber** () const	
RowColumnLabel*	**GetLabel1** () const	
RowColumnLabel*	**GetLabel2** () const	
void	**UpdateLabel1** (const string *txt)	
void	**UpdateLabel2** (const string *txt)	
unsigned	**NrCells** () const	
int	**GetIndex** (LinePiece *l)	
unsigned	**HasEmptyText** ()	*return nr of cells having empty texts.*
unsigned	**HasText** (const string *s)	*return nr of cells having this text.*
bool	**IsAllEmpty** ()	
void	**EmptyCells** ()	
virtual void	**CalcLabelPositions** ()	

Protected Members

List <Cell *> *
cells *the cells in the vector.*

List <LinePiece *> *
lines *cell boundaries between the cells in the vector.*

Grafport* **GetGrafport** () const

Private Members

TableViewer*
viewer

Grafport* **grafport**

unsigned **number** *the row or column numbers.*

RowColumnLabel*
label1 *a number label drawn near beginning of vector (like spreadsheet)*

RowColumnLabel*
label2 *a number label drawn near end of vector.*

bool **labelsVisible** *is label number displayed ?*

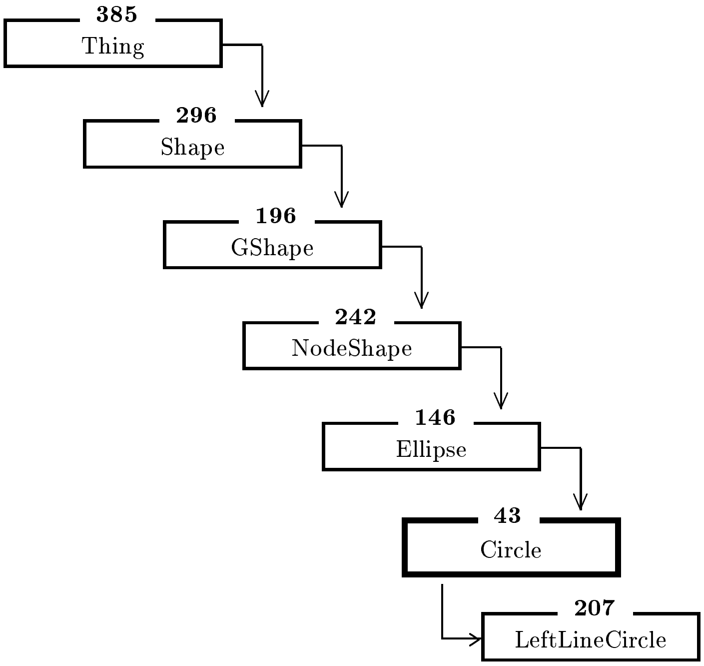
TextAlign::Type
alignment *beginning=left, end=right, center=center.*

43

class **Circle** : public Ellipse

ellipse class always having equal width and height.

Inheritance



Public Members

static const int
CIRCLE_WIDTH

Circle (ShapeView *v, Grafport *g, int x, int y, int diameter = CIRCLE_WIDTH)

Circle (const NodeShape &)

Shape* **Clone** ()

int **GetClassType** () const

Protected Members

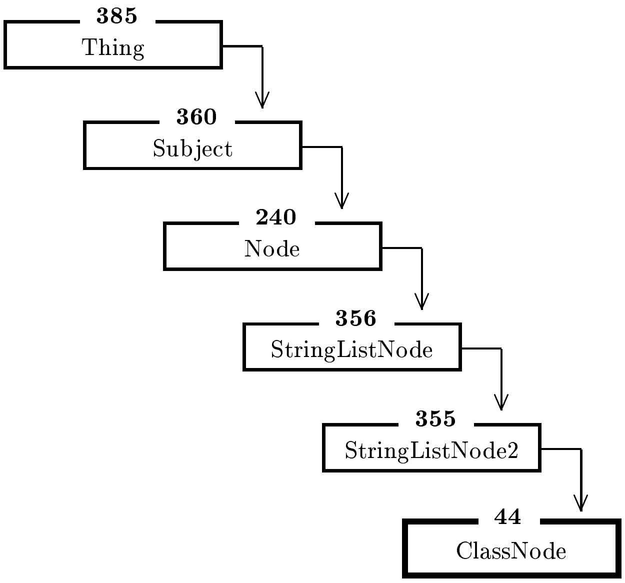
void **SetSize** (int width, int height)

44

class **ClassNode** : public StringListNode2

object class (CRD) node class.

Inheritance



Public Members

	ClassNode (CRGraph *g)
Subject*	**Clone** ()
int	**GetClassType** () const
enum	**TextErrType**
ClassNode::TextErrType	**SetAttribute** (const string *s, unsigned n, bool update)
const string*	**GetAttribute** (unsigned n)
unsigned	**NrAttributes** ()
bool	**HasAttribute** (const string *s)

ClassNode::TextErrType
SetOperation (const string *s, unsigned n,
bool update)

const string*
GetOperation (unsigned n)

unsigned **NrOperations** ()

bool **HasOperation** (const string *s)

bool **ReadMembers** (InputFile *f, double format)

45

class **Code**

Functions and constants for shape and subject codes.

Public Members

enum **Type**

static const char*
GetName (int code)
returns the name of a type having this code (0 if it doesn't exist).

static int **GetCode** (const char* name)
returns the code of type name (0 if it doesn't exist).

static int **GetIndex** (int code, const int codelist[])
returns code subject index (> 1) or 0 if code isn't in code list.

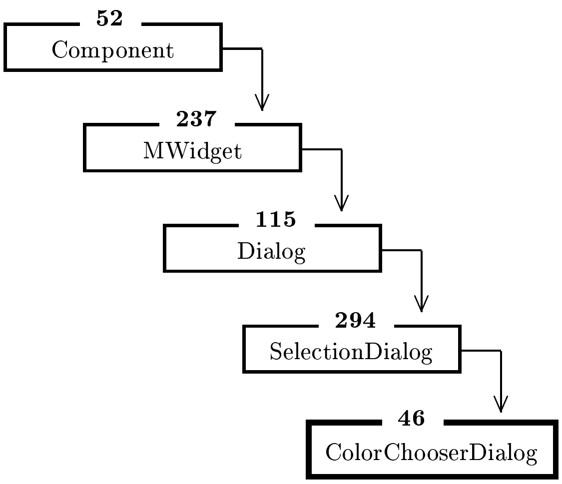
static TypeName
typeNames []

46

class **ColorChooserDialog** : public SelectionDialog

color chooser dialog class.

Inheritance



Public Members

	ColorChooserDialog (Widget parent)
	~ColorChooserDialog ()
const char*	**GetClassName** ()
static void	**UpdateColorCB** (Widget, XtPointer, XtPointer)
void	**ShowColor** (unsigned p)
void	**ManageFillToggle** (bool b)
void	**SetFillToggle** (bool b)
bool	**FillToggleOn** ()
void	**SetFillToggleLabel** (const char *s)
void	**Popup** ()

Protected Members

void	**Configure** ()	

Private Members

Widget	**colorLabel**	
List <Color *> *	**colors**	
Colormap	**colormap**	*X colormap.*
Widget	**fillToggle**	

47

class **Color**

Associates a color name with a pixel value and a RGB value.

Public Members

Color (string x, double r, double g, double b)

Color (const char *x, double r, double g, double b)

Color (string x, unsigned long p)

Color (const char *x, unsigned long p)

string **name**

double **red**

double **green**

double **blue**

unsigned long **pixel**

bool **allocated**

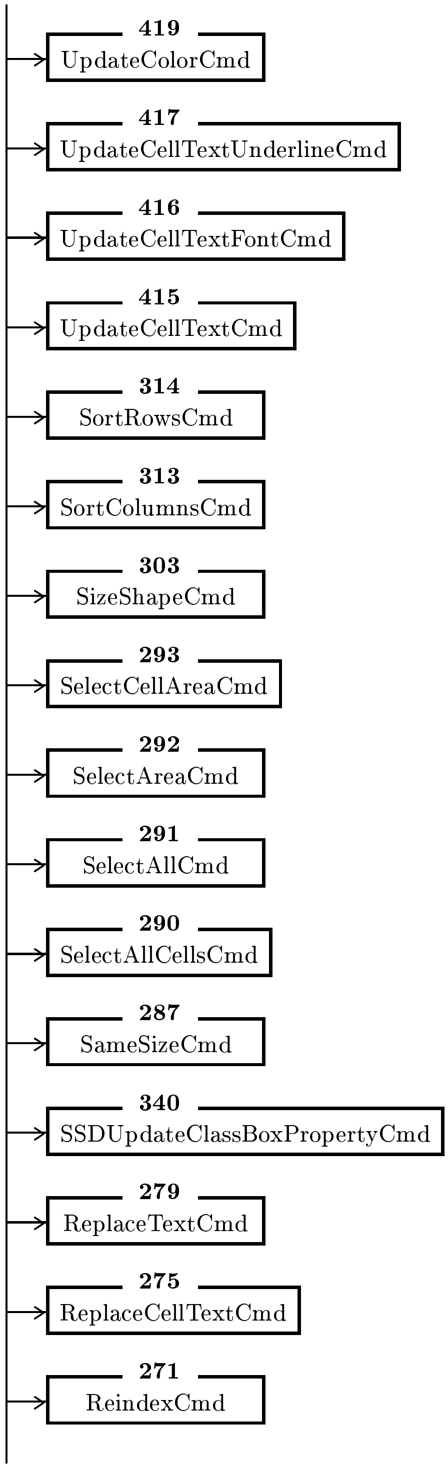
48

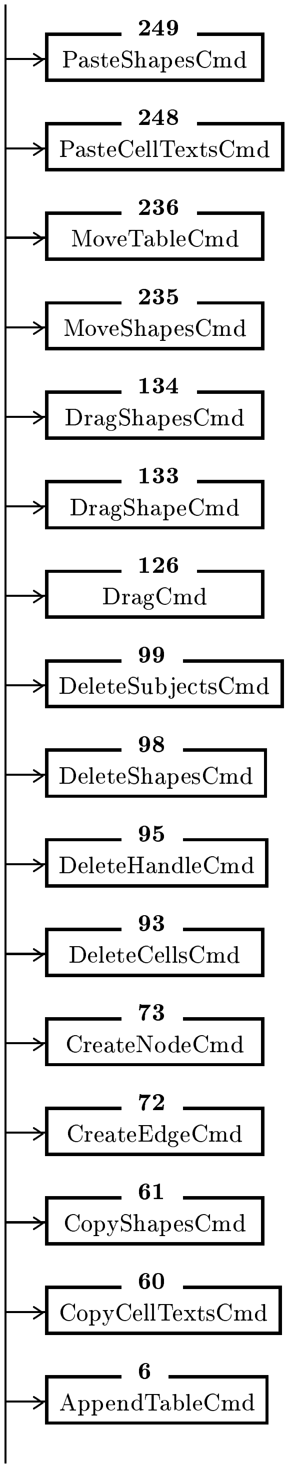
class **Command**

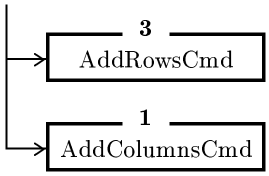
(abstract) undo-able command class.

Inheritance

- 48 Command
  - 439 UpdateVectorAlignmentCmd
  - 438 UpdateTypeCmd
  - 437 UpdateTextUnderlineCmd
  - 435 UpdateTextCmd
  - 434 UpdateTextAlignmentCmd
  - 432 UpdateReadDirectionCmd
  - 429 UpdateLineWidthCmd
  - 428 UpdateLineTableCmd
  - 426 UpdateLineStyleCmd
  - 425 UpdateLineEndCmd
  - 423 UpdateFontCmd
  - 421 UpdateCurveCmd


419
UpdateColorCmd
417
UpdateCellTextUnderlineCmd
416
UpdateCellTextFontCmd
415
UpdateCellTextCmd
314
SortRowsCmd
313
SortColumnsCmd
303
SizeShapeCmd
293
SelectCellAreaCmd
292
SelectAreaCmd
291
SelectAllCmd
290
SelectAllCellsCmd
287
SameSizeCmd
340
SSDUpdateClassBoxPropertyCmd
279
ReplaceTextCmd
275
ReplaceCellTextCmd
271
ReindexCmd


249
PasteShapesCmd
248
PasteCellTextsCmd
236
MoveTableCmd
235
MoveShapesCmd
134
DragShapesCmd
133
DragShapeCmd
126
DragCmd
99
DeleteSubjectsCmd
98
DeleteShapesCmd
95
DeleteHandleCmd
93
DeleteCellsCmd
73
CreateNodeCmd
72
CreateEdgeCmd
61
CopyShapesCmd
60
CopyCellTextsCmd
6
AppendTableCmd



Public Members

Command (Document *d, Viewer *v)

virtual **~Command** ()

enum **TrackType**

virtual void
TrackMouse (TrackType, Point*, Point*, Point*)
Called while mouse button is pressed dragged and released.

virtual void
Execute () *Perform the command.*

virtual void
UnExecute () *Undo the command.*

virtual void
ReExecute () *Redo the command.*

virtual void
Abort () *Abort the command.*

bool **CmdDone** () const
command is executed.

virtual const char*
GetName () const

int **ScaleCorrect** (int x) const

Point **ScaleCorrect** (const Point *pt) const

int **Scale** (int x) const

Point **Scale** (const Point *pt) const

void **SayAborted** ()

Protected Members

void **SayCommited** ()

void **SayUndone** ()

DrawWindow*
GetMainWindow () const

Viewer* **GetViewer** () const

Document*
GetDocument () const

Grafport* **GetGrafport** () const

void **SetCmdDone** (bool b)

Private Members

bool **cmdDone**

Document*
document

Viewer* **viewer**

DrawWindow*
mainwindow

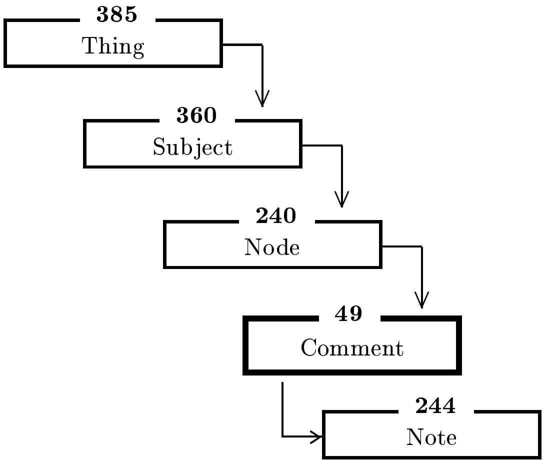
Grafport* **grafport**

49

class **Comment** : public Node

Comment node class.

Inheritance



Public Members

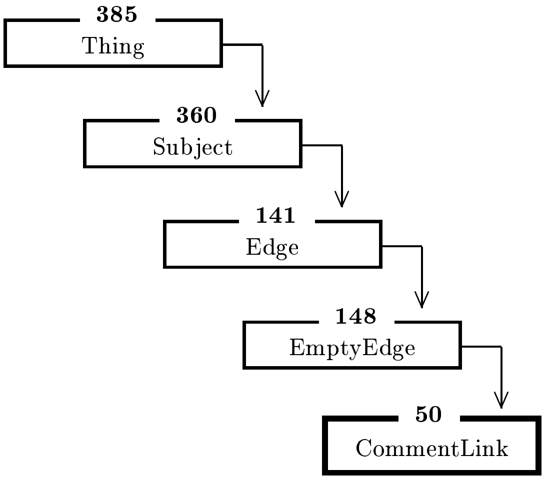
	Comment (Graph *g)
Subject*	**Clone** ()
int	**GetClassType** () const

50

class **CommentLink** : public EmptyEdge

edge that connects comments to other subjects.

Inheritance



Public Members

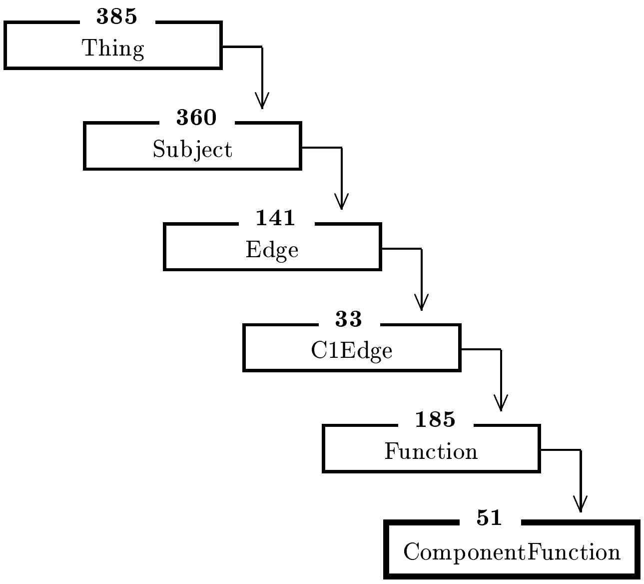
	CommentLink (Graph *g, Subject *n1, Subject *n2)
Subject*	**Clone** ()
int	**GetClassType** () const

51

class **ComponentFunction** : public Function

component function (CRD) edge class.

Inheritance



Public Members

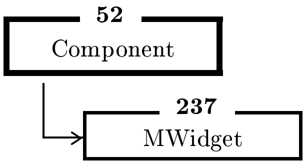
	ComponentFunction (CRGraph *g, Subject *n1, Subject *n2)
Subject*	**Clone** ()
int	**GetClassType** () const

52

class **Component**

base class for all UI components.

Inheritance



Public Members

	Component (const char *name)	
virtual	**~Component** ()	
virtual void	**Manage** ()	*Manage widget tree*
bool	**IsManaged** ()	*Is widget tree managed.*
virtual void	**Unmanage** ()	*Unmanage widget tree*
Widget	**GetWidget** ()	
Display*	**GetDisplay** ()	
Window	**GetWindow** ()	

Protected Members

void	**SetWidget** (Widget w)	
const string*	**GetName** () const	
void	**SetName** (const string *s)	
void	**SetName** (const char *s)	

Private Members

string	**name**
Widget	**widget**

53 class **Config**

class for maintaining global tcm configuration.

Public Members

	Config (const char* fn)
	~Config ()
bool	**Load** (const string* file)
bool	**Save** ()
void	**SetDefault** ()
const string*	**GetExportFormat** ()
const string*	**GetPrinterName** ()
const string*	**GetPrintCommand** ()
const string*	**GetPreviewCommand** ()
const string*	**GetFig2DevCommand** ()
const string*	**GetPrinterQueueCommand** ()
const string*	**GetPrinterRemoveCommand** ()
bool	**GetPrintColors** ()
bool	**GetPrintDuplexPages** ()
bool	**GetPrintTumbledPages** ()
const string*	**GetPostScriptFilterCommand** ()
const string*	**GetTmpFile** ()
bool	**GetSendEmailOnCrash** ()
bool	**GetSendEmailOnStartup** ()

const string*
GetEmailAddress ()

const string*
GetText2PSFilterCommand ()

enum **CharacterEncodingType**

int **GetCharacterEncoding** ()

enum **BannerPageType**

BannerPageType
GetBannerPage ()

enum **PageOrientationType**

enum **ShowHeaderType**

PageOrientationType
GetPageOrientation ()

bool **GetShowPageBoundary** ()

PageSize::Type
GetPageSize ()

bool **GetIncludePageNumbers** ()

bool **GetShowDocumentInfoAsHeader** ()

bool **GetShowDocumentInfoAsFooter** ()

double **GetScaleFactor** ()

bool **GetInlineEditor** ()

bool **GetAutoResizing** ()

bool **GetAskBeforeQuit** ()

const string*
GetFontFoundry ()

int **GetFontFamily** ()

int **GetFontStyle** ()

int **GetPointSize** ()

bool **GetScalableFonts** ()

bool **GetUnderlined** ()

TextAlign::Type
GetTextAlignment ()

int **GetLineWidth** ()

const string*

GetLineColor ()

const string*
GetTextColor ()

const string*
GetFillColor ()

::LineStyle::Type
GetLineStyle ()

const string*
GetDrawingBackground ()

const string*
GetDrawingForeground ()

int **GetDrawingScrollWidth** ()

int **GetDrawingScrollHeight** ()

int **GetDrawingWidth** ()

int **GetDrawingHeight** ()

int **GetDrawingMaxWidth** ()

int **GetDrawingMaxHeight** ()

bool **GetShowGrid** ()

bool **GetPointSnapping** ()

int **GetGridSize** ()

int **GetPointDistance** ()

int **GetClassicGUIstyle** ()
GUI style view mode

bool **GetPrivateColormap** ()

int **GetStartupSmall** ()
startup window view mode

int **GetTableMarginWidth** ()
margin width in pixels

int **GetTableMarginHeight** ()
margin height in pixels

TextAlign::Type
GetColumnAlignment ()

TextAlign::Type
GetRowAlignment ()

int **GetDefaultRowHeight** ()

initial/default row height in pixels

int	**GetDefaultColumnWidth** ()	*standard/default column width in pixels*
int	**GetNumberOfRows** ()	*initial number of rows*
int	**GetNumberOfColumns** ()	*initial number of columns*
bool	**GetShowRowColumnLabels** ()	
bool	**GetPrintRowColumnLabels** ()	
Point	**GetTableTopLeft** ()	*default topleft position of a table.*
int	**GetFETcolumnWidth** ()	
int	**GetTUTcolumnWidth** ()	
int	**GetTUTnumberOfColumns** ()	

Private Members

InputFile*	**ifile**	
static const char	**SYS_CONFIG** []	*system configuration file.*
static const char	**DEF_CONFIG** []	*default configuration file.*
static const char	**USER_CONFIG** []	*user configuration file.*
string	**exportFormat**	*various configurations.*
string	**printerName**	
string	**printCommand**	
string	**previewCommand**	
string	**fig2devCommand**	
string	**postScriptFilterCommand**	
string	**text2PSFilterCommand**	
string	**printerQueueCommand**	

string **printerRemoveCommand**

CharacterEncodingType
characterEncoding

string **tmpFile**

bool **sendEmailOnCrash**

bool **sendEmailOnStartup**

string **emailAddress**

bool **printColors**

bool **printDuplexPages**

bool **printTumbledPages**

BannerPageType
bannerPage

bool **showPageBoundary**

PageSize::Type
pageSize

PageOrientationType
pageOrientation

bool **includePageNumbers**

bool **showDocumentInfoAsHeader**

bool **showDocumentInfoAsFooter**

double **scaleFactor**

bool **inlineEditor**

bool **autoResizing**

bool **askBeforeQuit**

string **fontFoundry**

int **fontFamily**

int **fontStyle**

int **pointSize**

TextAlign::Type
textAlignment

bool **scalableFonts**

bool **underlined**

int **lineWidth**

LineStyle::Type

	lineStyle
string	**lineColor**
string	**textColor**
string	**fillColor**
int	**drawingScrollWidth**
int	**drawingScrollHeight**
string	**drawingBackground**
string	**drawingForeground**
int	**drawingWidth**
int	**drawingHeight**
int	**drawingMaxWidth**
int	**drawingMaxHeight**
bool	**showGrid**
bool	**pointSnapping**
int	**gridSize**
int	**pointDistance**
bool	**classicGUIstyle**
bool	**privateColormap**
bool	**startupSmall**
int	**tableMarginWidth**
int	**tableMarginHeight**
TextAlign::Type	**columnAlignment**
TextAlign::Type	**rowAlignment**
int	**defaultRowHeight**
int	**defaultColumnWidth**
int	**numberOfRows**
int	**numberOfColumns**
bool	**showRowColumnLabels**
bool	**printRowColumnLabels**
Point	**tableTopLeft**
int	**fetColumnWidth**

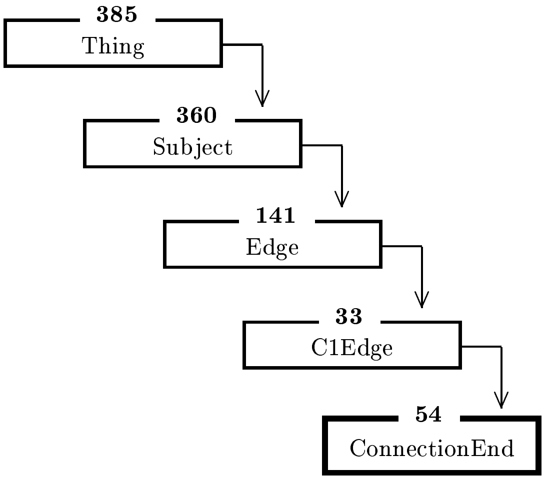
int	**tutColumnWidth**
int	**tutNumberOfColumns**
bool	**LoadEntries** ()
void	**CheckPrintCommand** ()
void	**CheckPrinterQueueCommand** ()
void	**CheckPrinterRemoveCommand** ()
void	**CheckPreviewCommand** ()
void	**CheckFig2DevCommand** ()
void	**CheckPSFilterCommand** ()
void	**CheckText2PSCommand** ()
void	**DefaultPrintCommand** ()
void	**DefaultPrinterQueueCommand** ()
void	**DefaultPrinterRemoveCommand** ()
void	**DefaultPreviewCommand** ()
void	**DefaultFig2DevCommand** ()
void	**DefaultPSFilterCommand** ()
void	**DefaultText2PSCommand** ()

54

class **ConnectionEnd** : public C1Edge

connection end (system network) edge class.

Inheritance



Public Members

	ConnectionEnd (SNGraph *g, Subject *n1, Subject *n2)
int	**GetClassType** () const
Subject*	**Clone** ()
Subject::NameErrType	**SetName** (const string *s) *returns WRONG_NAME.*
bool	**SetConstraint** (const string *s)

Private Members

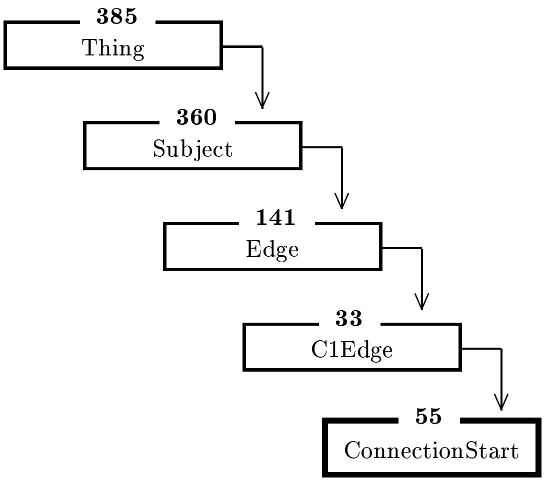
bool	**ParseCardConstraint** (const string *s)

55

class **ConnectionStart** : public C1Edge

connection start (system network) edge class.

Inheritance



Public Members

	ConnectionStart (SNGraph *g, Subject *n1, Subject *n2)
int	**GetClassType** () const
Subject*	**Clone** ()
Subject::NameErrType	**SetName** (const string *s) *returns WRONG_NAME.*
bool	**SetConstraint** (const string *s)

Private Members

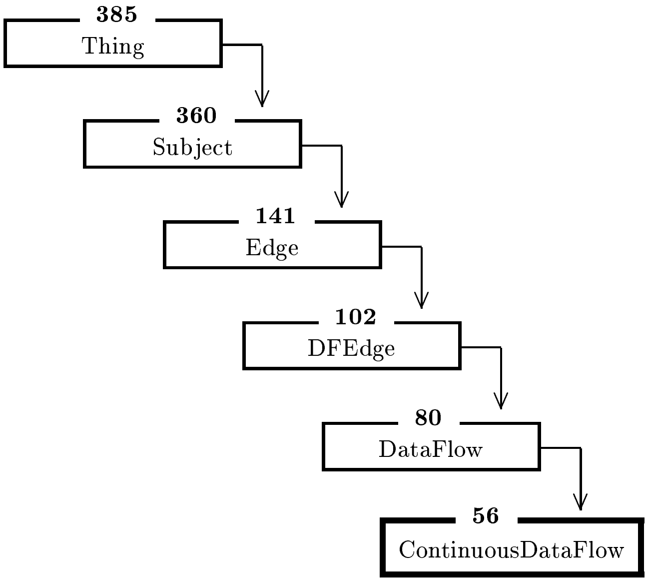
bool	**ParseCardConstraint** (const string *s)

56

class **ContinuousDataFlow** : public DataFlow

continuous data flow (DEFD) edge class.

Inheritance



Public Members

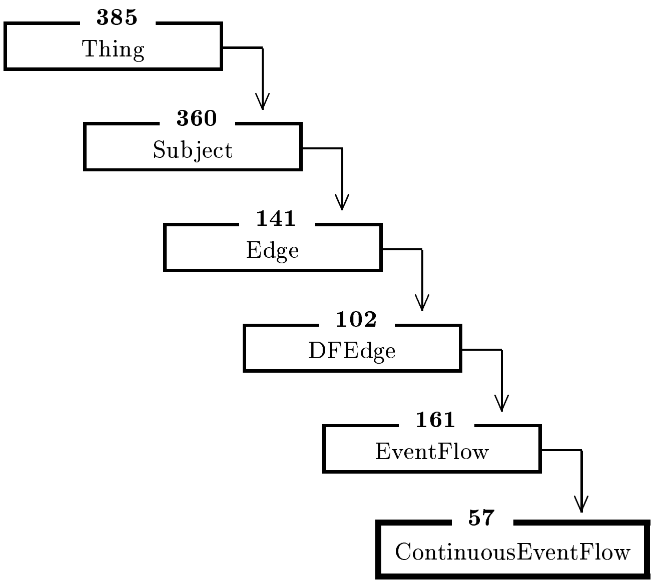
	ContinuousDataFlow (DCFGraph *g, Subject *n1, Subject *n2)
int	**GetClassType** () const
Subject*	**Clone** ()

57

class **ContinuousEventFlow** : public EventFlow

continuous event flow (EFD) edge class.

Inheritance



Public Members

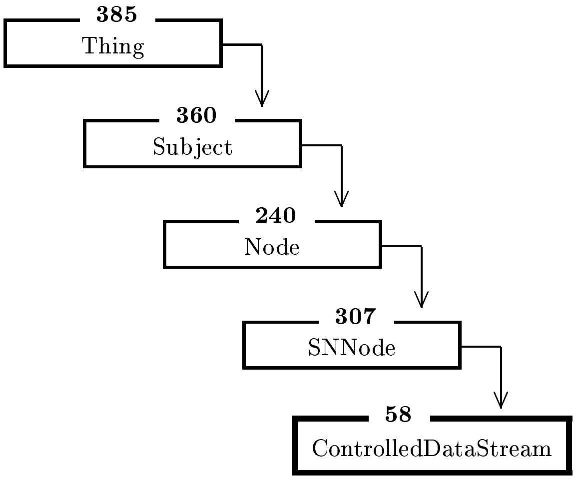
	ContinuousEventFlow (DCFGraph *g, Subject *n1, Subject *n2)
int	**GetClassType** () const
Subject*	**Clone** ()
Subject::NameErrType	**SetName** (const string *name)

58

class **ControlledDataStream** : public SNNode

system network controlled data stream node class.

Inheritance



Public Members

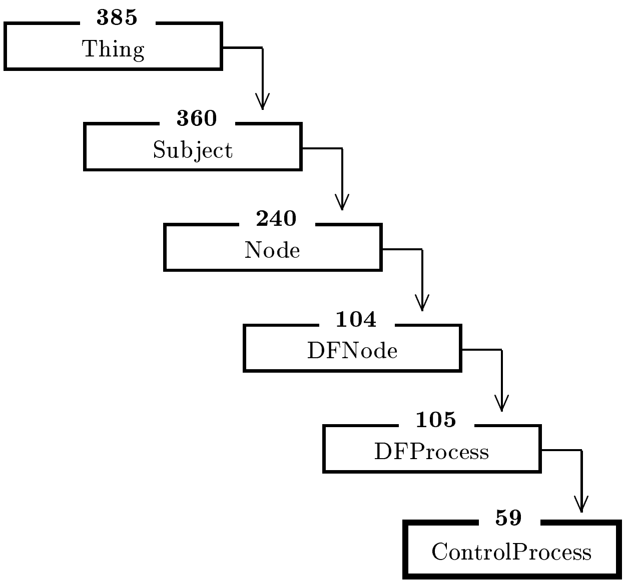
	ControlledDataStream (SNGraph *g)
int	**GetClassType** () const
Subject*	**Clone** ()

59

class **ControlProcess** : public DFProcess

control process (EFD) node class.

Inheritance



Public Members

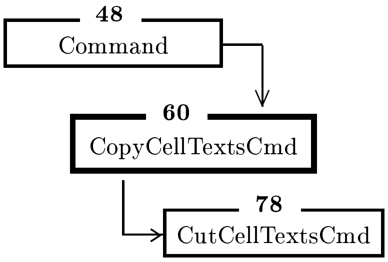
	ControlProcess (DCFGraph *g)
int	**GetClassType** () const
Subject*	**Clone** ()

60

class **CopyCellTextsCmd** : public Command

copy selected cell texts command.

Inheritance



Public Members

	CopyCellTextsCmd (Table *t, TableViewer *v)
virtual	**~CopyCellTextsCmd** ()
void	**Execute** ()
void	**UnExecute** ()
const char*	**GetName** () const

Protected Members

TableViewer*
GetTableViewer () const

List <CellText *> *
GetCellTexts () const

Private Members

List <CellText *> *
cellTexts

List <Point *> *

rcPositions

TableViewer*
tviewer

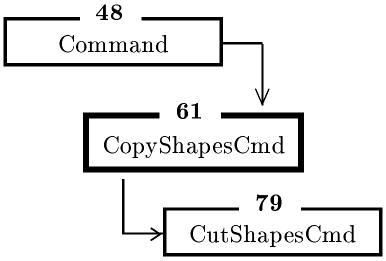
CellTextBuffer*
buffer

61

class **CopyShapesCmd** : public Command

command class for copying shapes.

Inheritance



Public Members

	CopyShapesCmd (ShapeView *v)
virtual	**~CopyShapesCmd** ()
void	**Execute** ()
void	**UnExecute** ()
void	**Copy** (List<GShape *> *shapes)
const char*	**GetName** () const

Protected Members

ShapeView*	**GetView** () const
Diagram*	**GetDiagram** () const

Private Members

Subjects*	**subjects**
Shapes*	**shapes**

Subjects* **bufferSubjects**

Shapes* **bufferShapes**

ShapeView*
view

Diagram* **diagram**

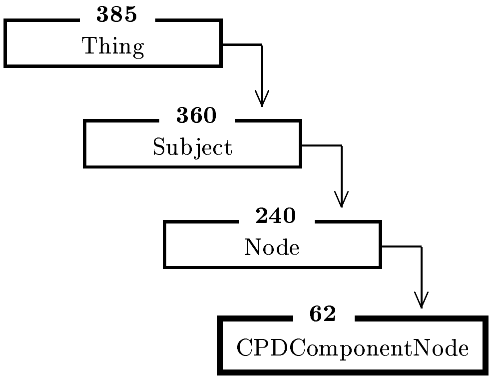
Buffer* **buffer**

62

class **CPDComponentNode** : public Node

UML Component node class.

Inheritance



Public Members

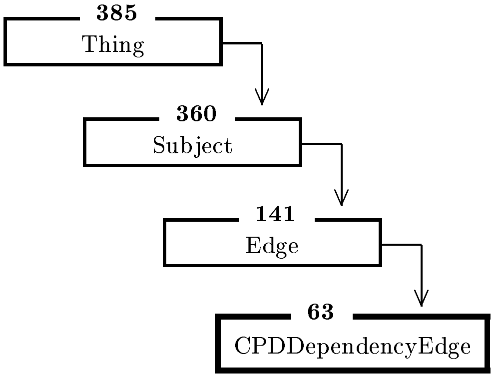
	CPDComponentNode (CPGraph *g)
int	**GetClassType** () const
Subject*	**Clone** ()

63

class **CPDDependencyEdge** : public Edge

Component Dependency edge class

Inheritance



Public Members

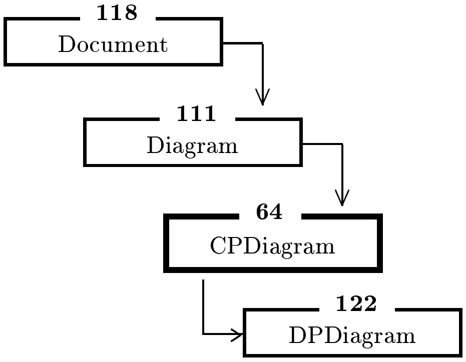
	CPDDependencyEdge (CPGraph *g, Subject *s1, Subject *s2)
int	**GetClassType** () const
Subject*	**Clone** ()

64

class **CPDiagram** : public Diagram

component diagram class.

Inheritance



Public Members

CPDiagram (Config *, CPWindow *, CPViewer *, CPGraph *)

Node* **CreateNode** ()

Edge* **CreateEdge** (Subject *n1, Subject *n2)

NodeShape* **CreateNodeShape** (Node *node, int x, int y)

Line* **CreateLine** (Edge *edge, GShape *fromShape, GShape *toShape, List<Point *> *line)

void **UpdateNodeType** (int n)

void **UpdateEdgeType** (int n)

static const int **DOT_WIDTH**

Protected Members

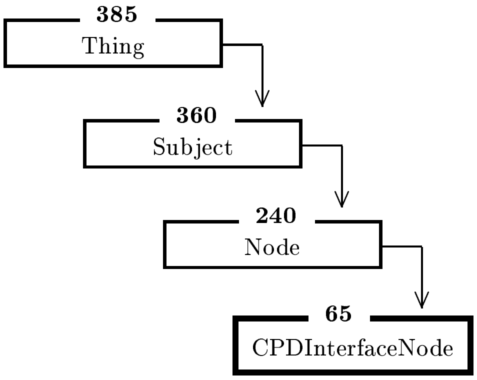
Thing*	**CreateThing** (int classNr)
bool	**HasIndexNode** (int code)
bool	**HasIndexShape** (int code)

65

class **CPDInterfaceNode** : public Node

UML Interface node class.

Inheritance



Public Members

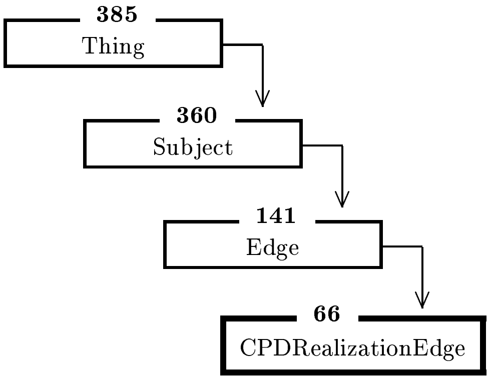
	CPDInterfaceNode (CPGraph *g)
int	**GetClassType** () const
Subject*	**Clone** ()

66

class **CPDRealizationEdge** : public Edge

Component realization edge class

Inheritance



Public Members

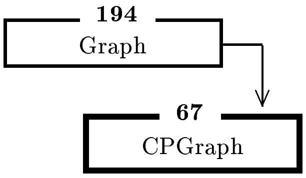
	CPDRealizationEdge (CPGraph *g, Subject *s1, Subject *s2)
int	**GetClassType** () const
Subject*	**Clone** ()

67

class **CPGraph** : public Graph

component diagram graph class.

Inheritance



Public Members

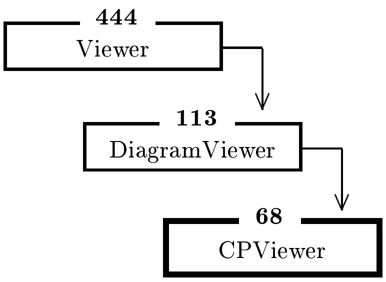
	CPGraph ()
void	**InitConnections** ()

68

class **CPViewer** : public DiagramViewer

component diagram viewer class.

Inheritance



Public Members

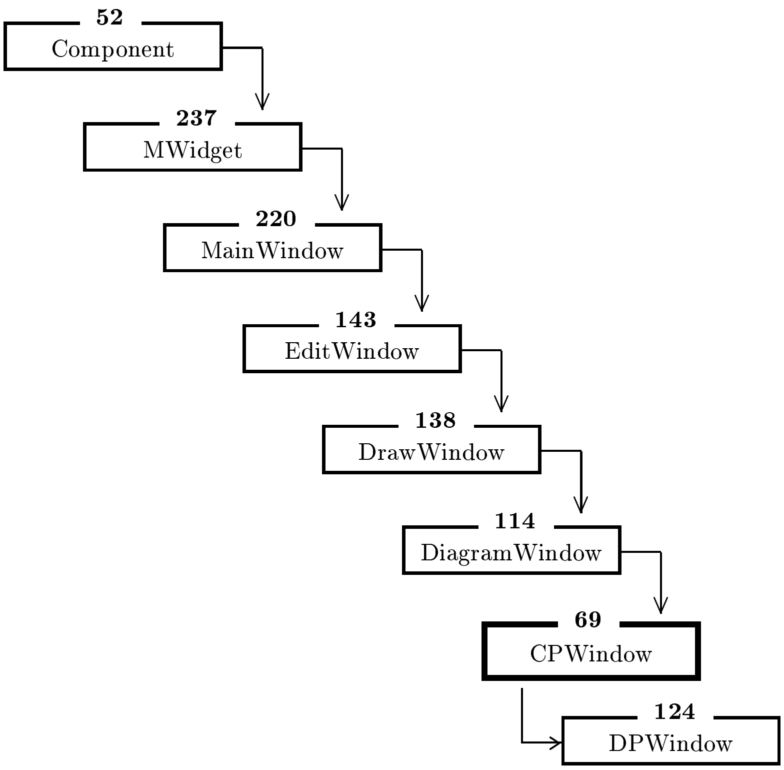
CPViewer (Config *c, CPWindow *w)

69

class **CPWindow** : public DiagramWindow

component diagram edit window class.

Inheritance



Public Members

CPWindow (const char *name)

Private Members

int **GetTool** ()

DiagramViewer*

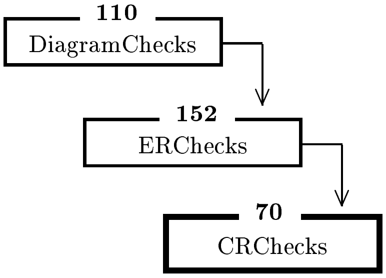
	CreateViewer (Config *c)
Graph*	**CreateGraph** ()
Diagram*	**CreateDiagram** (Config *c, DiagramViewer *v, Graph *g)
void	**DetermineIcons** ()
static const char*	**CP_NODE_NAMES** []
static const Bitmap	**CP_NODE_ICONS** []
static const char*	**CP_EDGE_NAMES** []
static const Bitmap	**CP_EDGE_ICONS** []

70

class **CRChecks** : public ERChecks

implements checks for soft constraints on CR diagrams.

Inheritance



Public Members

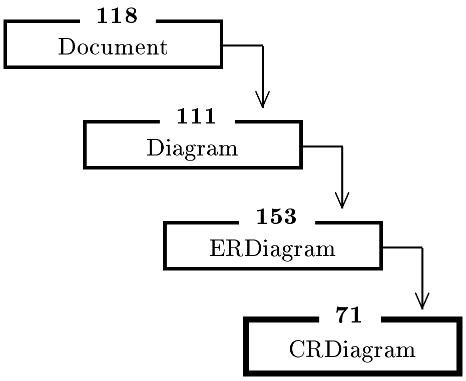
	CRChecks (Diagram *d, Graph *g)
unsigned	**CheckDoubleNamelessFunctions** (int nodetype1, int nodetype2, string &buf)

71

class **CRDiagram** : public ERDiagram

class-relationship diagram class.

Inheritance



Public Members

	CRDiagram (Config *, CRWindow *, CRViewer*, CRGraph *)
virtual	**~CRDiagram** ()
Node*	**CreateNode** ()
Edge*	**CreateEdge** (Subject *n1, Subject *n2)
NodeShape*	**CreateNodeShape** (Node *node, int x, int y)
Line*	**CreateLine** (Edge *edge, GShape *fromShape, GShape *toShape, List<Point *> *line)
void	**UpdateNodeType** (int n)
void	**UpdateEdgeType** (int n)
bool	**SetText** (TextShape *t, const string *s)
void	**CheckDocument** ()
bool	**SetAttribute** (ClassNode *obj, const string *s, unsigned nr)

Set Attribute in obj at position nr.

bool **SetOperation** (ClassNode *obj, const string *s, unsigned nr)
Set Operation in obj at position nr.

Protected Members

void **MakeErrorMessage** (ClassNode *obj, ClassNode::TextErrType result, const string *s, string *msg)
Give error after failing to update attribute or operation.

bool **CheckTaxonomyCombination** (Edge *edge)
Checks if combining TaxonomyJunction and ModeJunction is correct.

Thing* **CreateThing** (int classNr)

CRChecks* **GetCRChecks** ()

Private Members

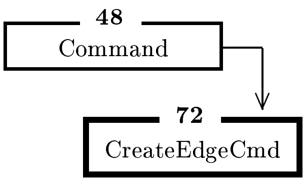
CRChecks* **crChecks**

72

class **CreateEdgeCmd** : public Command

command class for creating edges.

Inheritance



Public Members

	CreateEdgeCmd (GShape *from)
virtual	**~CreateEdgeCmd** ()
void	**TrackMouse** (TrackType aPhase, Point *anchorPoint, Point *previousPoint, Point *nextPoint)
void	**Execute** ()
void	**UnExecute** ()
void	**ReExecute** ()
void	**Abort** ()
const char*	**GetName** () const

Protected Members

void	**AddPoint** (Point *pt)
void	**EraseLines** ()

Private Members

Diagram*	**diagram**

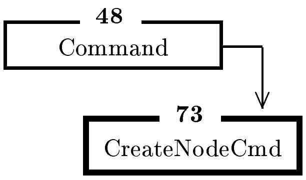
ShapeView*	**view**
Edge*	**edge**
Line*	**line**
GShape*	**fromShape**
GShape*	**toShape**
List <Point *> *	**points**
Point	**abortPoint**
bool	**connectEdges**

73

class **CreateNodeCmd** : public Command

command class for creating nodes.

Inheritance



Public Members

	CreateNodeCmd (ShapeView *v, int x, int y)
virtual	**~CreateNodeCmd** ()
void	**Execute** ()
void	**UnExecute** ()
const char*	**GetName** () const

Private Members

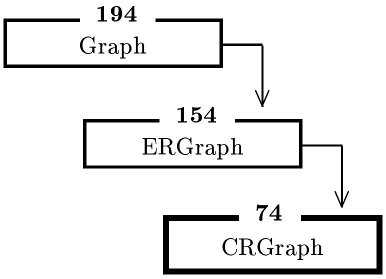
Diagram*	**diagram**
ShapeView*	**view**
Node*	**node**
NodeShape*	**shape**

74

class **CRGraph** : public ERGraph

class-relationship graph class

Inheritance



Public Members

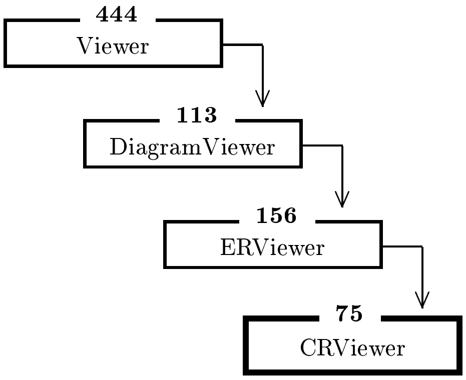
	CRGraph ()
void	**InitConnections** ()

75

class **CRViewer** : public ERViewer

class-relationship viewer class.

Inheritance



Public Members

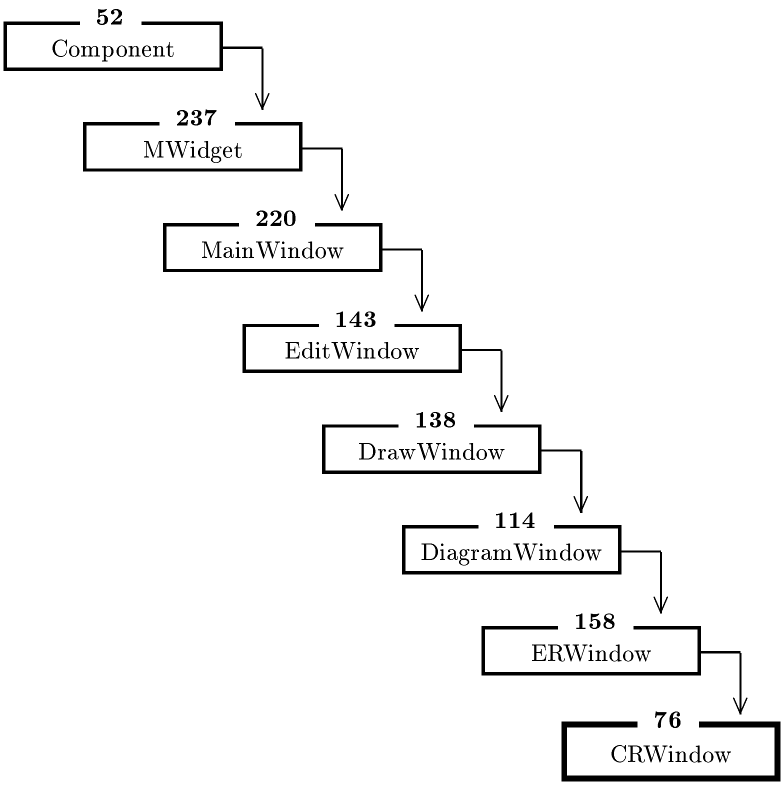
	CRViewer (Config *c, CRWindow *w)	
void	**UpdateNodeShapeType** (int)	*change node shape type of selected types.*

76

class **CRWindow** : public ERWindow

class-relationship edit window class.

Inheritance



Public Members

	CRWindow (const char *name)
virtual	**~CRWindow** ()

Private Members

int	**GetTool** ()

DiagramViewer*
CreateViewer (Config *config)

Graph* **CreateGraph** ()

Diagram* **CreateDiagram** (Config *c, DiagramViewer *v, Graph *g)

void **DetermineIcons** ()

MenuItem*
diagramCRBoxItems

void **InitMenuItems** ()

static const char*
CR_NODE_NAMES []

static const Bitmap
CR_NODE_ICONS []

static const char*
CR_EDGE_NAMES []

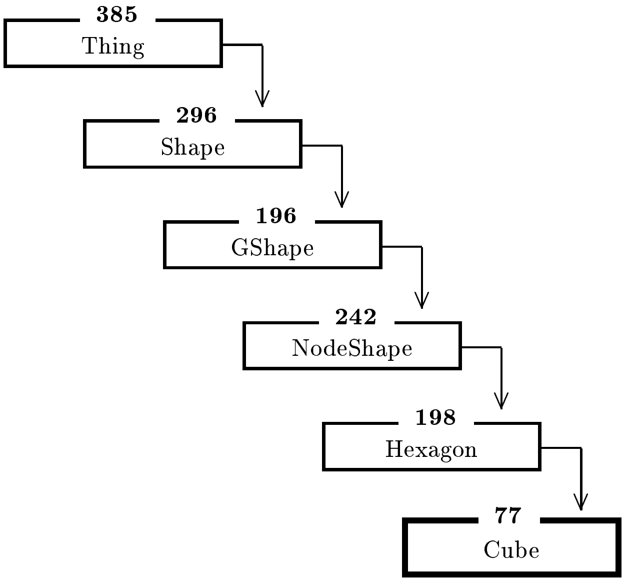
static const Bitmap
CR_EDGE_ICONS []

77

class **Cube** : public Hexagon

Cube shape

Inheritance



Public Members

	Cube (ShapeView *v, Grafport *g, int x, int y)
	Cube (const NodeShape &)
Shape*	**Clone** ()

Protected Members

void	**DrawOutLine** (const Point* p)
void	**SetPosition** (const Point *position, bool snap=True)
void	**SetSize** (int w, int h)

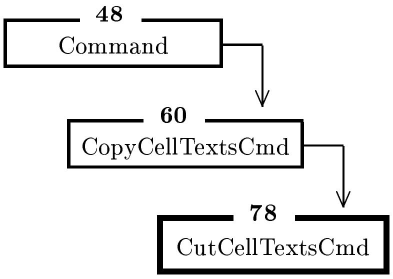
void	**DrawShape** ()
void	**CalcPoints** (int x, int y)
Point	**GiveSnp** (Point to) const
Point	**GiveSnp** (Point to, int order, int count, int& code)
int	**GetClassType** () const
void	**CalcPositionName** ()
void	**SetTextShape** ()

78

class **CutCellTextsCmd** : public CopyCellTextsCmd

cut selected cell texts command.

Inheritance



Public Members

	CutCellTextsCmd (Table *t, TableViewer *v)
virtual	**~CutCellTextsCmd** ()
void	**Execute** ()
void	**UnExecute** ()
const char*	**GetName** () const

Private Members

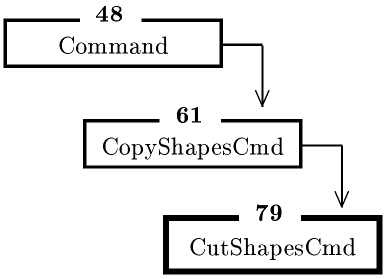
List <Cell *> *	**selection**
List <string *> *	**oldStrings**

79

class **CutShapesCmd** : public CopyShapesCmd

command class for cutting shapes.

Inheritance



Public Members

	CutShapesCmd (ShapeView *v)
virtual	**~CutShapesCmd** ()
void	**Execute** ()
void	**UnExecute** ()
const char*	**GetName** () const

Private Members

List <Subject *> *
cutSubjects

List <Subject *> *
updateSubjects

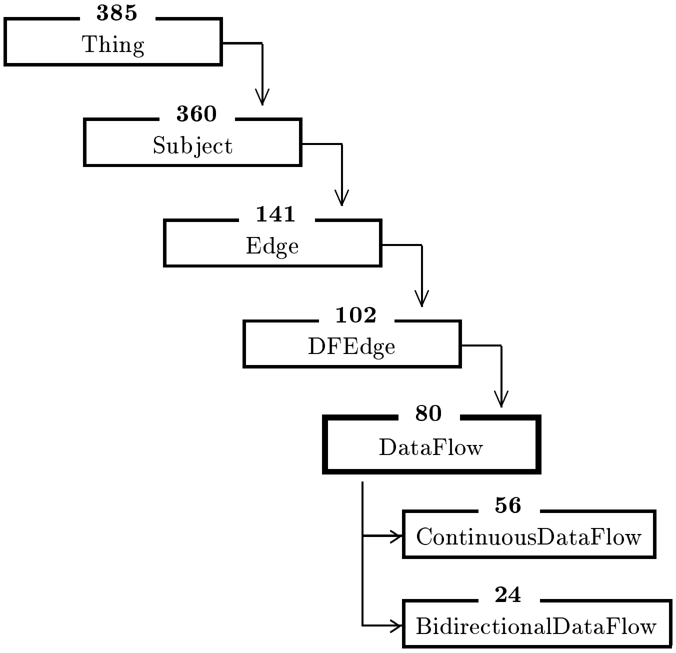
List <GShape *> *
cutShapes

80

class **DataFlow** : public DFEdge

data flow edge class.

Inheritance



Public Members

	DataFlow (DFGraph *g, Subject *n1, Subject *n2)
int	**GetClassType** () const
Subject*	**Clone** ()
Subject::NameErrType	**SetName** (const string *name)
enum	**ContentType**
void	**WriteMembers** (OutputFile *f)

bool **ReadMembers** (InputFile *f, double format)

Private Members

ContentType **contentType**

string **content**

static void **ContentType2String** (ContentType a, string *s)

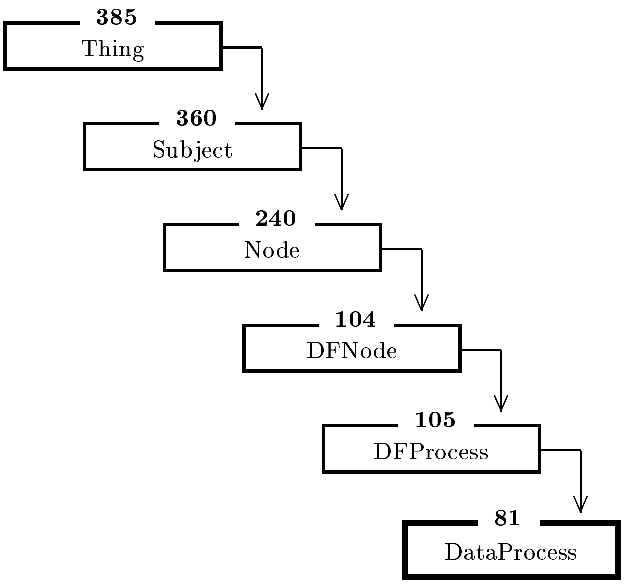
static ContentType **String2ContentType** (const string *s)

81

class **DataProcess** : public DFProcess

data process (DF) node class.

Inheritance



Public Members

	DataProcess (DFGraph *g)
int	**GetClassType** () const
Subject*	**Clone** ()
enum	**ActivationType**
void	**WriteMembers** (OutputFile *f)
bool	**ReadMembers** (InputFile *f, double format)
bool	**SetReferences** (AssocList *)
bool	**IsProcessGroup** () const

bool **IsInstantaneous** () const

void **SetInstantaneous** (bool b)

void **SetProcessGroup** (bool b)

const string*
GetMinispec () const

void **SetMinispec** (const string *s)

ActivationType
GetActivationMechanism () const

void **SetActivationMechanism** (ActivationType a)

const string*
GetStimulus () const

void **SetStimulus** (const string *s)

const string*
GetTimeExpression () const

void **SetTimeExpression** (const string *s)

static void
ActivationType2String (ActivationType a,
string *s)

static ActivationType
String2ActivationType (const string *s)

Private Members

string **minispec**

bool **instantaneous**

ActivationType
activationMechanism

string **timeExpression**

string **stimulus**

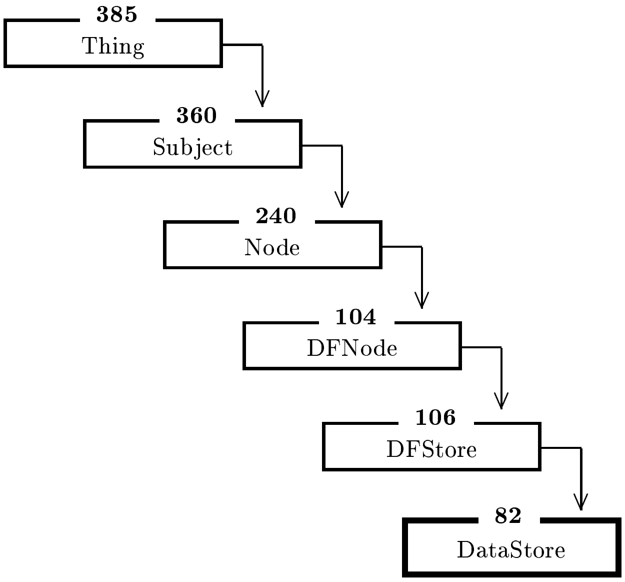
bool **processGroup**

82

class **DataStore** : public DFStore

data store (DF) node class.

Inheritance



Public Members

	DataStore (DFGraph *g)
	DataStore (const DataStore &dt)
virtual	**~DataStore** ()
int	**GetClassType** () const
Subject*	**Clone** ()
void	**WriteMembers** (OutputFile *f)
bool	**ReadMembers** (InputFile *f, double format)

Private Members

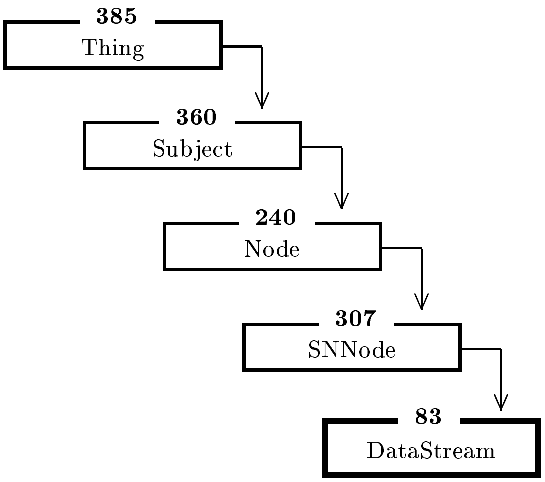
List <string *> *
atomicSubjects

83

class **DataStream** : public SNNode

system network data stream node class.

Inheritance



Public Members

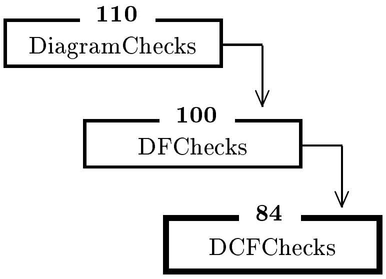
	DataStream (SNGraph *g)
int	**GetClassType** () const
Subject*	**Clone** ()

84

class **DCFChecks** : public DFChecks

implements checks for soft constraints on EFDs.

Inheritance



Public Members

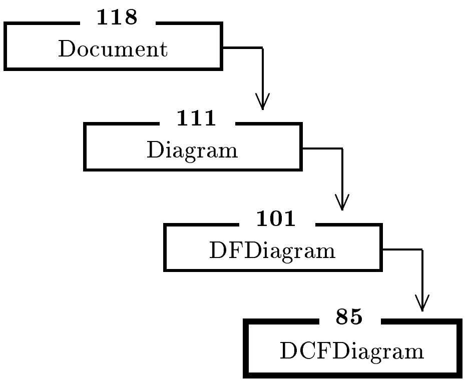
	DCFChecks (Diagram *d, Graph *g)	
unsigned	**CheckControlProcessCoherence** (string &chkbuf)	
unsigned	**CheckContinuousSplitMergeEdges** (string &chkbuf)	
unsigned	**CheckActivationMechanisms** (string &chkbuf)	*Check if activation mechanisms are correctly specified.*
unsigned	**CheckPersistences** (string &chkbuf)	*Check discrete processes do not output continuous flows.*

85

class **DCFDiagram** : public DFDiagram

data and event flow diagram class

Inheritance



Public Members

	DCFDiagram (Config *, DCFWindow *, DCFViewer *, DCFGraph *)
virtual	**~DCFDiagram** ()
Node*	**CreateNode** ()
Edge*	**CreateEdge** (Subject *s1, Subject *s2)
NodeShape*	**CreateNodeShape** (Node *node, int x, int y)
Line*	**CreateLine** (Edge *edge, GShape *fromShape, GShape *toShape, List<Point *> *line)
void	**UpdateNodeType** (int n)
void	**UpdateEdgeType** (int n)
void	**AddSubject** (Subject *)
void	**CheckDocument** ()

void **EditPersistence** (Subject *s)

void **EditActivation** (Subject *s)

void **SetPersistence** ()

void **SetActivation** ()

void **UpdateActivationToggle** ()

Protected Members

Thing* **CreateThing** (int classNr)

bool **CheckEdgeConstraints** (Subject *s1, Subject *s2)

void **UpdateActivationDialog** (int n)

Private Members

ToggleListDialog*
persistenceDialog

ToggleListDialog*
activationDialog

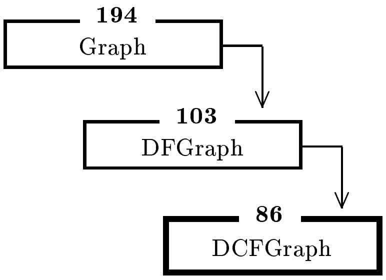
DCFChecks*
dcfChecks

86

class **DCFGraph** : public DFGraph

data and event flow graph class

Inheritance



Public Members

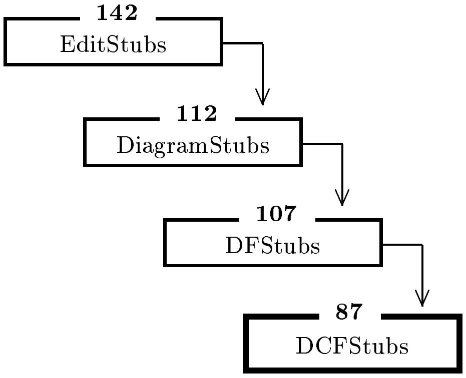
	DCFGraph ()
void	**InitConnections** ()

87

class **DCFStubs** : public DFStubs

Callback stubs for specific EFD functions.

Inheritance



Public Members

static void
PersistenceCB (Widget w, XtPointer, XtPointer)

static void
PersistenceOKCB (Widget, XtPointer, XtPointer)

static void
ActivationToggleCB (Widget w, XtPointer, XtPointer)

static void
ActivationOKCB (Widget w, XtPointer, XtPointer)

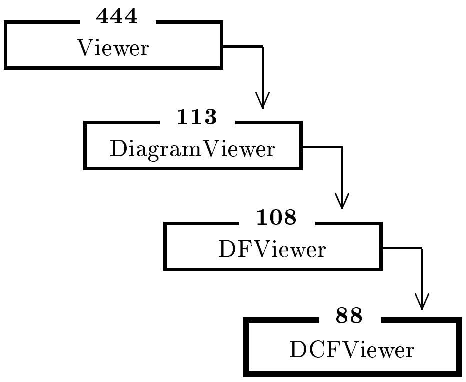
static void
ActivationCB (Widget, XtPointer, XtPointer)

88

class **DCFViewer** : public DFViewer

data and event flow viewer class.

Inheritance



Public Members

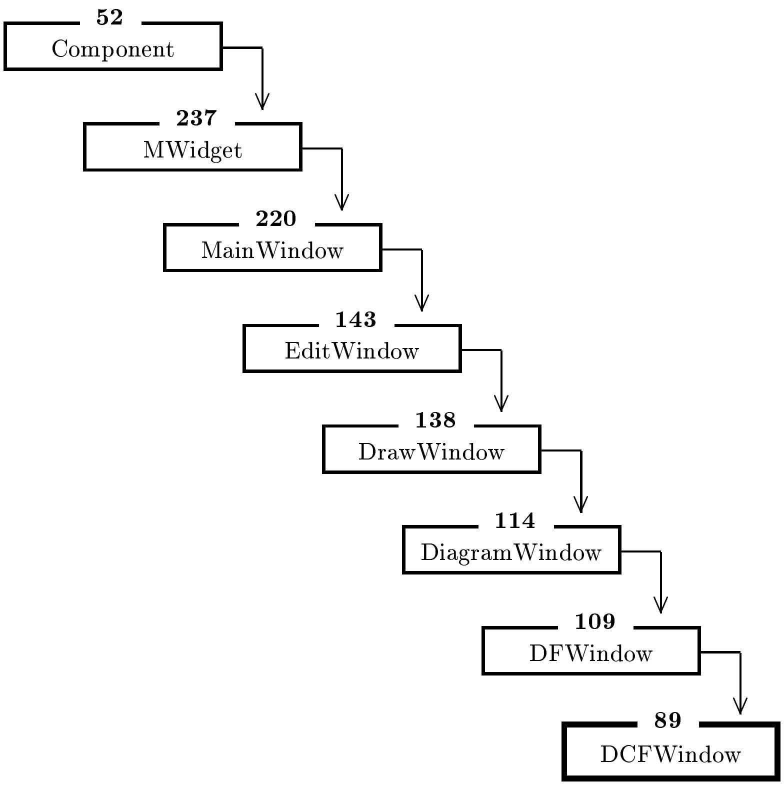
	DCFViewer (Config *c, DCFWindow *w)
void	**Persistence** ()
void	**Activation** ()

89

class **DCFWindow** : public DFWindow

data and event flow edit window class.

Inheritance



Public Members

DCFWindow (const char *name)

Private Members

int **GetTool** ()

DiagramViewer*

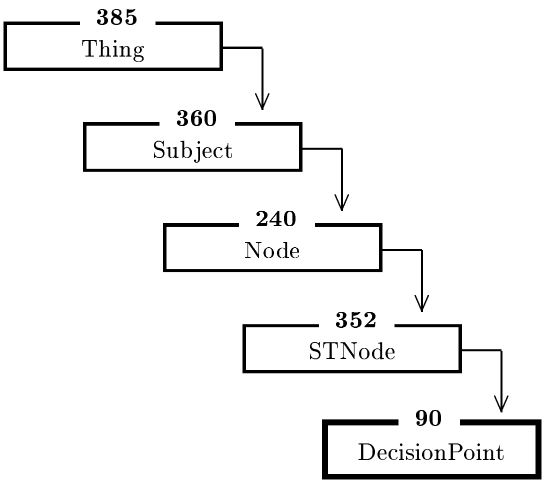
	CreateViewer (Config *c)
Graph*	**CreateGraph** ()
Diagram*	**CreateDiagram** (Config *c, DiagramViewer *v, Graph *g)
void	**DetermineIcons** ()
void	**InitMenuItems** ()
static const char*	**DCF_NODE_NAMES** []
static const Bitmap	**DCF_NODE_ICONS** []
static const char*	**DCF_EDGE_NAMES** []
static const Bitmap	**DCF_EDGE_ICONS** []

90

class **DecisionPoint** : public STNode

decision point (STD) node class.

Inheritance



Public Members

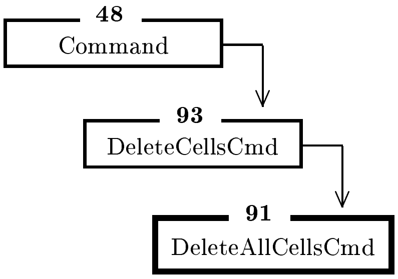
	DecisionPoint (STGraph *g)
Subject*	**Clone** ()
int	**GetClassType** () const

91

class **DeleteAllCellsCmd** : public DeleteCellsCmd

delete table contents command.

Inheritance



Public Members

	DeleteAllCellsCmd (Table *t, TableViewer *v)
void	**Execute** ()
void	**UnExecute** ()
const char*	**GetName** () const

Private Members

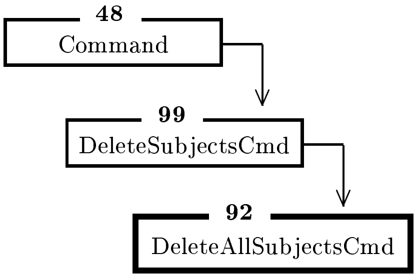
void	**DetermineRows** ()

92

class **DeleteAllSubjectsCmd** : public DeleteSubjectsCmd

command class for deleting all subjects in view.

Inheritance



Public Members

DeleteAllSubjectsCmd (ShapeView *v)

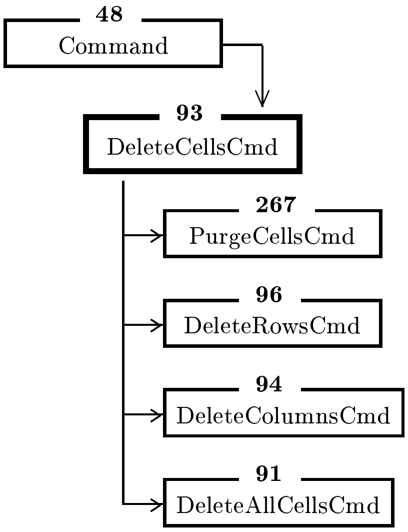
const char*
GetName () const

93

class **DeleteCellsCmd** : public Command

abstract delete cells from table command.

Inheritance



Public Members

DeleteCellsCmd (Table *t, TableViewer *v)

virtual **~DeleteCellsCmd** ()

void **Execute** ()

void **UnExecute** ()

const char*
GetName () const

Protected Members

List <CellRow *> *
GetRows () const

List <CellColumn *> *

GetColumns () const

TableViewer*
GetTableViewer () const

Private Members

List <CellRow *> *
rows

List <CellColumn *> *
columns

TableViewer*
tviewer

virtual void
DetermineRows ()

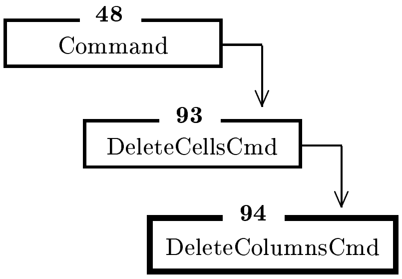
virtual void
DetermineColumns ()

94

class **DeleteColumnsCmd** : public DeleteCellsCmd

delete columns from table command.

Inheritance



Public Members

	DeleteColumnsCmd (Table *t, TableViewer *v)
void	**Execute** ()
void	**UnExecute** ()
const char*	**GetName** () const

Private Members

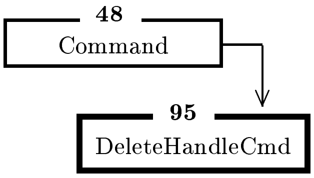
void	**DetermineColumns** ()

95

class **DeleteHandleCmd** : public Command

command class for delete (intermediate) line handles.

Inheritance



Public Members

	DeleteHandleCmd (Line *s, unsigned which)
void	**UnExecute** ()
void	**Execute** ()
const char*	**GetName** () const

Private Members

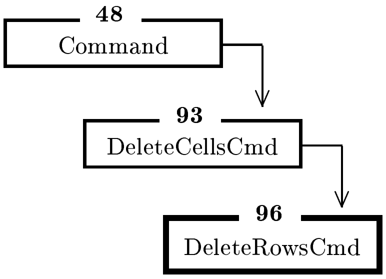
Line*	**line**
Point	**oldPoint**
unsigned	**whichHandle**

96

class **DeleteRowsCmd** : public DeleteCellsCmd

delete rows from table command.

Inheritance



Public Members

	DeleteRowsCmd (Table *t, TableViewer *v)
void	**Execute** ()
void	**UnExecute** ()
const char*	**GetName** () const

Private Members

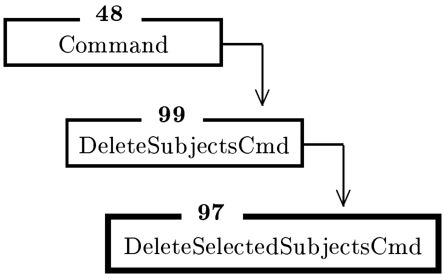
void	**DetermineRows** ()

97

class **DeleteSelectedSubjectsCmd** : public DeleteSubjectsCmd

command class for deleting all subjects of selected shapes.

Inheritance



Public Members

DeleteSelectedSubjectsCmd (ShapeView *v)

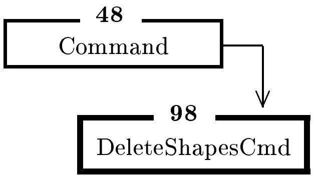
const char*
GetName () const

98

class **DeleteShapesCmd** : public Command

command class for deleting (possibly) duplicate node shapes.

Inheritance



Public Members

	DeleteShapesCmd (ShapeView *v)
virtual	**~DeleteShapesCmd** ()
void	**Execute** ()
void	**UnExecute** ()
const char*	**GetName** () const

Private Members

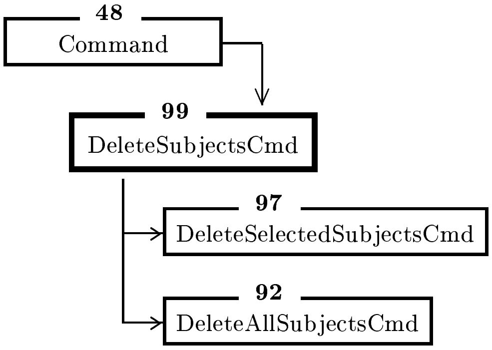
List <GShape *> *	**shapes**	*the shapes that have to be deleted.*
List <Subject *> *	**deleteSubjects**	*subjects that are deleted.*
List <Subject *> *	**updateSubjects**	*subject whose duplication marks have to be updated.*
ShapeView*	**view**	
Diagram*	**diagram**	
DiagramViewer*	**dviewer**	

99

class **DeleteSubjectsCmd** : public Command

command class for deleting subjects and their shapes.

Inheritance



Public Members

	DeleteSubjectsCmd (ShapeView *v)
virtual	**~DeleteSubjectsCmd** ()
void	**Execute** ()
void	**UnExecute** ()

Protected Members

List <Subject *> *	**GetSubjects** () const
List <GShape *> *	**GetShapes** () const
Diagram*	**GetDiagram** () const
ShapeView*	**GetView** () const

Private Members

List <Subject *> *
subjects

List <GShape *> *
shapes

Diagram* **diagram**

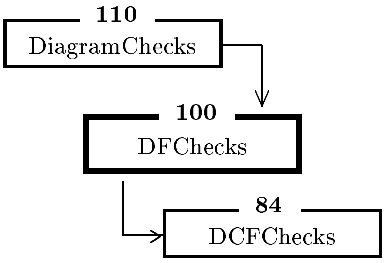
ShapeView*
view

100

class **DFChecks** : public DiagramChecks

implements checks for soft constraints on DFDs.

Inheritance



Public Members

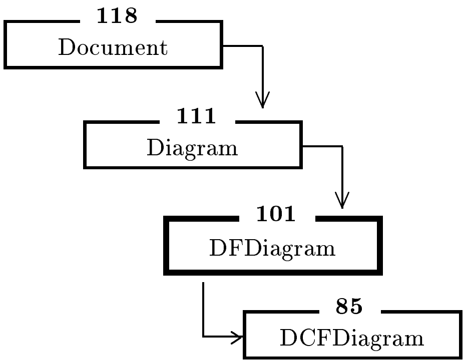
	DFChecks (Diagram *d, Graph *g)	
unsigned	**CheckDataProcessCoherence** (string &chkbuf)	*each data process should have an input and an output flow.*
unsigned	**CheckSplitMergeNodeCoherence** (string &chkbuf)	*Check if splitmerge nodes are properly connected.*
unsigned	**CheckNamelessSplitMergeEdges** (string &chkbuf)	*empty names are not allowed except for identical split/merge.*
unsigned	**CheckNamelessFlows** (int edgetype, int nodetype1, int nodetype2, string &chkbuf)	*Check that flows have a name.*
unsigned	**CheckMinispecs** (string &chkbuf)	*Check that each data process has a minispec.*
void	**WriteCheckProcess** (DFProcess *d, string &chkbuf)	

101

class **DFDiagram** : public Diagram

data flow diagram class.

Inheritance



Public Members

	DFDiagram (Config *, DFWindow *, DFViewer *, DFGraph *)
virtual	**~DFDiagram** ()
Node*	**CreateNode** ()
Edge*	**CreateEdge** (Subject *s1, Subject *s2)
NodeShape*	**CreateNodeShape** (Node *node, int x, int y)
Line*	**CreateLine** (Edge *edge, GShape *fromShape, GShape *toShape, List<Point *> *line)
void	**UpdateNodeType** (int n)
void	**UpdateEdgeType** (int n)
void	**CheckDocument** ()
void	**SetDiagram** (const string *diagram) *Check and set DF diagram.*
void	**EditMinispec** (Subject *s)

void **SetMinispec** ()

Protected Members

void **SetEditProcess** (DataProcess *e)

DataProcess*
GetEditProcess ()

Thing* **CreateThing** (int classNr)

virtual bool
CheckEdgeConstraints (Subject *s1, Subject *s2)

bool **HasIndexNode** (int code)

bool **HasIndexShape** (int code)

Private Members

TextEditDialog*
minispecDialog

DataProcess*
editProcess

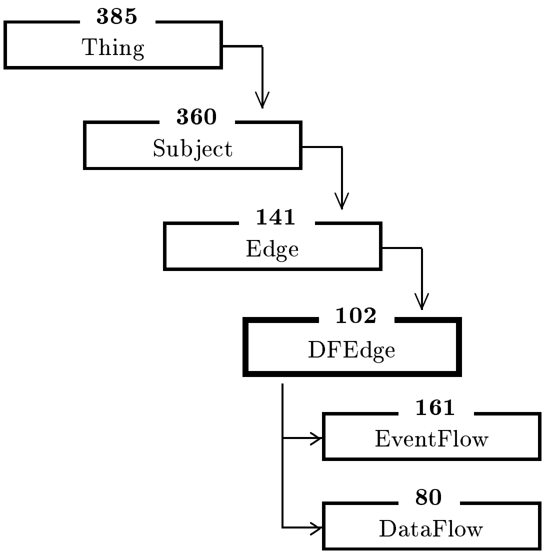
DFChecks*
dfChecks

102

class **DFEdge** : public Edge

an edge in a data flow diagram.

Inheritance



Public Members

	DFEdge (DFGraph *g, Subject *n1, Subject *n2)
	DFEdge (const DFEdge &e)
virtual	**~DFEdge** ()
static bool	**TriggerLabel** (const string *s)
static bool	**E_D_Label** (const string *s)
void	**WriteMembers** (OutputFile *f)
bool	**ReadMembers** (InputFile *f, double format)
bool	**SetReferences** (AssocList *)

Protected Members

List <DFEdge *> *
GetComponents ()

Private Members

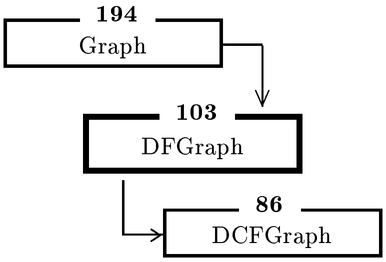
List <DFEdge *> *
components

103

class **DFGraph** : public Graph

data flow graph class.

Inheritance



Public Members

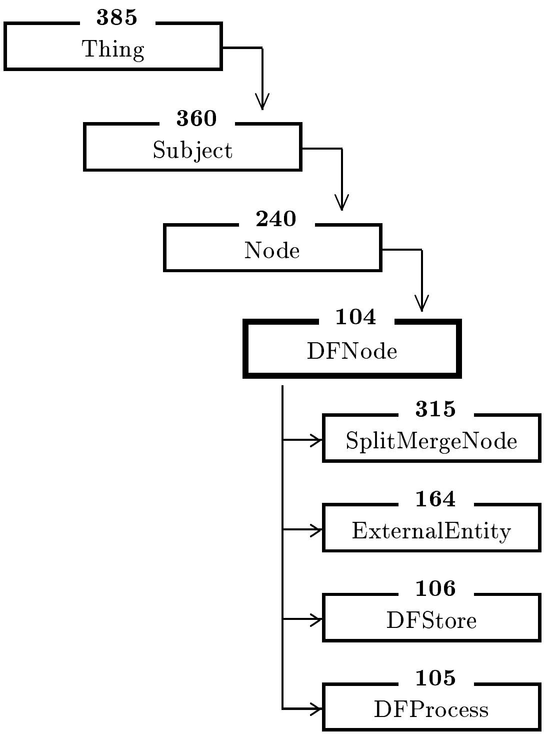
	DFGraph ()
void	**InitConnections** ()

104

class **DFNode** : public Node

abstract data flow node class.

Inheritance



Public Members

DFNode (DFGraph *g)

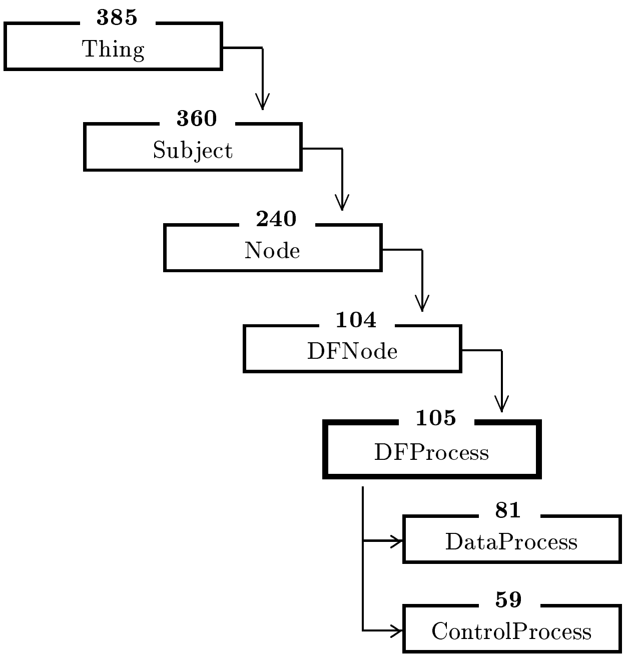
Subject::NameErrType
SetName (const string *s)

105

class **DFProcess** : public DFNode

abstract DFD process node class.

Inheritance



Public Members

DFProcess (DFGraph *g)

Subject::NameErrType
SetIndex (const string *s)

static bool
CorrectIndex (const string *s)

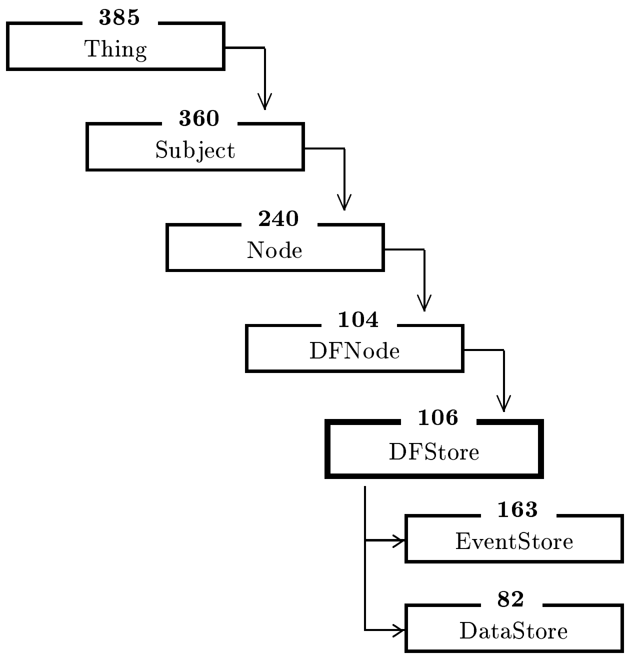
bool **ReadMembers** (InputFile *ifile, double format)

106

class **DFStore** : public DFNode

data store (DF) node class.

Inheritance



Public Members

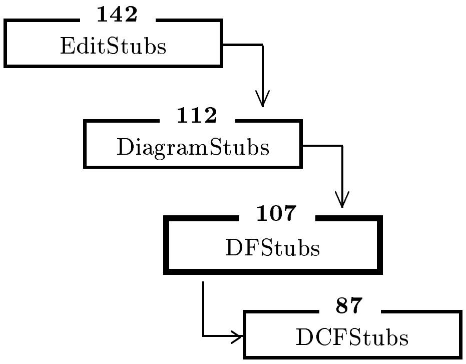
DFStore (DFGraph *g)

107

class **DFStubs** : public DiagramStubs

Callback stubs for specific DFD functions.

Inheritance



Public Members

static void
ChangeDiagramCB (Widget w, XtPointer, XtPointer)

static void
MinispecCB (Widget, XtPointer, XtPointer)

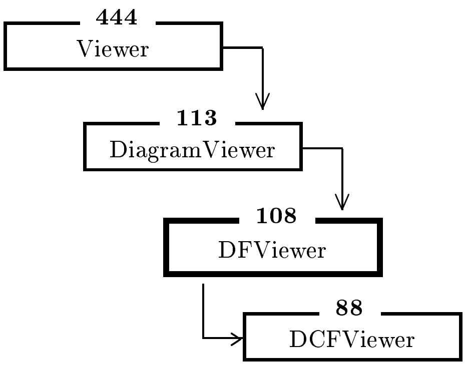
static void
MinispecOKCB (Widget w, XtPointer, XtPointer)

108

class **DFViewer** : public DiagramViewer

data flow viewer class.

Inheritance



Public Members

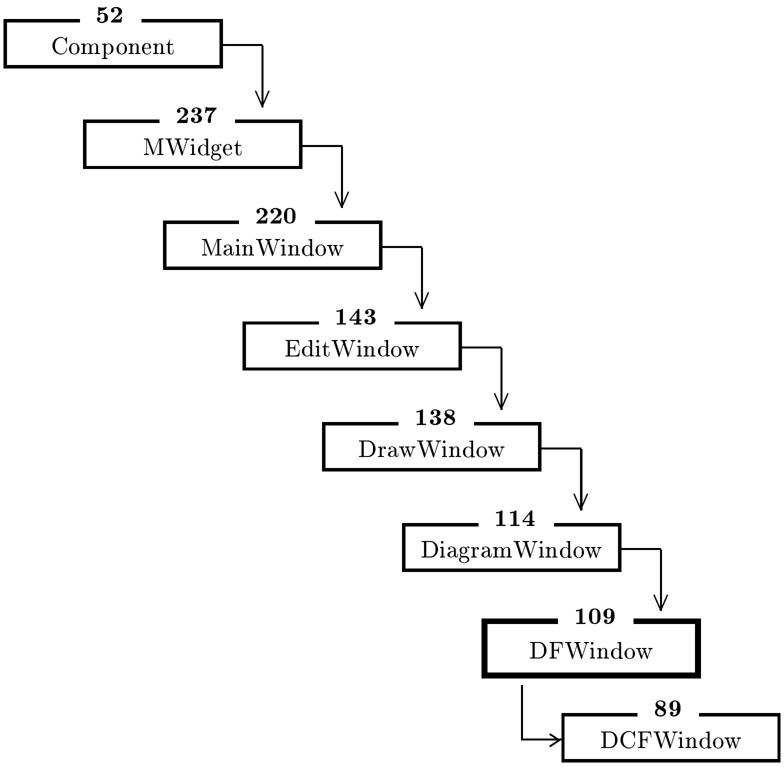
	DFViewer (Config *c, DFWindow *w)
void	**Minispec** ()

109

class **DFWindow** : public DiagramWindow

data flow edit window class.

Inheritance



Public Members

	DFWindow (const char *name)
virtual	**~DFWindow** ()
void	**SetDiagram** (const char *s) *dataflow diagrams: set diagram field.*
void	**SetDiagram** (const string *s)

Protected Members

MenuItem*
dfdItems

Private Members

int **GetTool** ()

bool **ShowIndexToggle** ()

Widget **CreateMenuBar** (Widget)

DiagramViewer*
CreateViewer (Config *config)

Graph* **CreateGraph** ()

Diagram* **CreateDiagram** (Config *c, DiagramViewer *v, Graph *g)

void **DetermineIcons** ()

Widget **CreateTiles** (Widget parent)

void **CreateDiagramField** (Widget parent)

Widget **dfDiagram** *Motif TextField for data flow diagram.*

Menu* **dfdMenu**

void **InitMenuItems** ()

static const char*
DF_NODE_NAMES []

static const Bitmap
DF_NODE_ICONS []

static const char*
DF_EDGE_NAMES []

static const Bitmap
DF_EDGE_ICONS []

static const char*
DF_INDEX_NAME

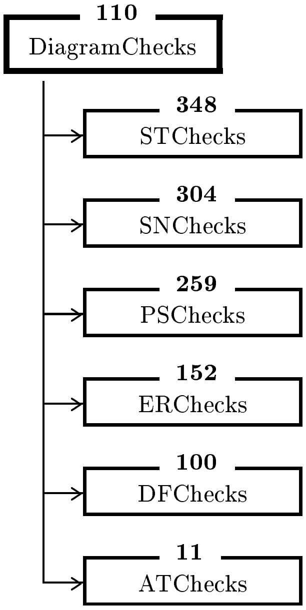
static const Bitmap
DF_INDEX_ICON

110

class **DiagramChecks**

implements generic checks for soft constraints on diagrams.

Inheritance



Public Members

	DiagramChecks (Diagram *d, Graph *g)
unsigned	**CheckNodeCount** (unsigned count, int nodeType, string &chkbuf) *diagram should have count nodes of nodeType.*
unsigned	**CheckNodeCount** (unsigned min, unsigned max, int nodeType, string &chkbuf) *diagram should have min <= count <= max nodes of nodeType.*
unsigned	**CheckEdgeCount** (unsigned count, int edgeType, string &chkbuf)

diagram should have count edges of edgeType

unsigned **CheckEdgeCount** (unsigned min, unsigned max, int edgeType, string &chkbuf)
diagram should have min <= count <= max edges of edgeType.

unsigned **CheckDoubleIndexes** (string &chkbuf)
check that nodes unique indexes.

unsigned **CheckNamelessNodes** (int nodeType, string &chkbuf)
diagram should have no nameless nodes of nodeType

unsigned **CheckNamelessEdges** (int edgeType, string &chkbuf)
diagram should have no nameless edges of edgeType

unsigned **CheckIllegalNodeNames** (int nodeType, const string *name, string &chkbuf)
diagram should have no nodes of nodeType named 'name'.

unsigned **CheckIllegalEdgeNames** (int edgeType, const string *name, string &chkbuf)
diagram should have no edges of edgeType named 'name'.

unsigned **CheckNamelessEdges** (int edgeType, int type1, int type2, string &chkbuf)
diagram should have no nameless edges of edgeType between types.

unsigned **CheckDoubleNamelessEdges** (int edgeType, int type1, int type2, string &chkbuf)
diagram shouldn't have double nameless edges between these types.

unsigned **CheckDoubleNodes** (int nodeType, string &chkbuf)

diagram should not have multiple nodes of nodeType with same name.

unsigned **CheckConnected** (int nodeType, bool index, string &chkbuf)
nodeType nodes should be connected (index: name or index in msg)

unsigned **CheckConnected** (int nodeType, int toNode, int min, int max, bool index, string &chkbuf)
nodeType nodes should be connected (index: name or index in msg)

unsigned **CheckJunctionCoherence** (int nodeType, int parenttype, int childtype, unsigned minChildren, string &chkbuf)
nodes of nodeType connected by 1 parent edge and minChildren edges.

unsigned **CheckCountEdgesFrom** (int nodeType, int edgeType, unsigned min, unsigned max, bool zeroAllowed, bool index, string &chkbuf)
nodes of nodeType should have min/max departing edges of edgeType.

unsigned **CheckReachability** (int rootType, int nodeType, bool index, string &chkbuf)
nodes of nodeType are reachable from >= 1 nodes of rootType.

Protected Members

Graph* **GetGraph** () const

Diagram* **GetDiagram** () const

unsigned **CheckCount** (unsigned min, unsigned max, int type, string &chkbuf, bool node)

unsigned **CheckIllegalNames** (int type, const string *name, string &chkbuf, bool node)

Private Members

Graph* **graph**

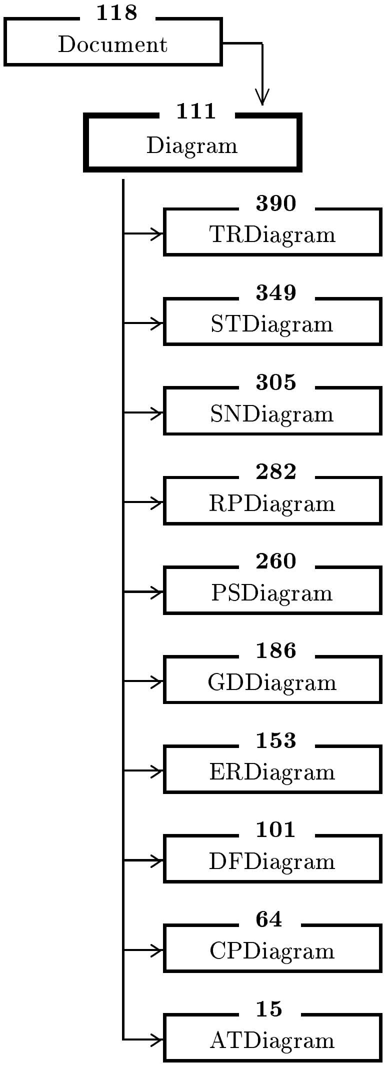
Diagram* **diagram**

111

class **Diagram** : public Document

(abstract) diagram class.

Inheritance



Public Members

Diagram (Config *, DiagramWindow *, DiagramViewer *, Graph *)

virtual **~Diagram** ()

void **Initialize** ()

Graph* **GetGraph** () const

AssocList* **GetAssocList** () const

virtual Node*
CreateNode () *Create a new node subject. Depends on current node type.*

virtual Edge*
CreateEdge (Subject *n1, Subject *n2)
Create edge, checks if connection is allowed.

virtual NodeShape*
CreateNodeShape (Node *node, int x, int y)
Create a node shape at (x, y), with default size and node as subject.

virtual Line*
CreateLine (Edge *edge, GShape *fromShape, GShape *toShape, List<Point *> *line)
Create a line of lineType from fromShape to toShape via points.

virtual void
UpdateNodeType (int num)
set/reset current node shape type to/from num.

virtual void
UpdateEdgeType (int num)
set/reset current edge shape type to/from num.

void **SetCurve** (bool set)
set/reset curved edge shapes.

bool **IsCurve** () const

bool **CheckConnection** (Subject *n1, Subject *n2)

Check if connection between those two nodes is allowed.

virtual void
AddSubject (Subject *s)
Add node/edge to graph data structure.

void **AddSubjects** (List<Subject *> *s)
Add nodes and edges to graph data structure.

virtual void
RemoveSubject (Subject *s)
Remove node/edge from graph data structure.

void **RemoveSubjects** (List<Subject *> *s)
Remove nodes and edges from graph data structure.

void **CompleteSubjects** (List<Subject *> *subjects)
add edges to subjects which connect the subjects in subjects.

void **CompleteEdges** (List<Subject *> *subjects)
add subjects to subjects which are connected by edges in subjects.

void **CompleteSubject** (List<Subject *> *edges, Subject *subject)
add to edges all edges of which subject is part.

void **RemoveAll** () *Remove all nodes and edges.*

void **AnnotateSubject** (Subject *s)
Annotate subject via pop-up dialog.

void **SetSubjectAnnotation** (string *s)
Annotate annotationSubject

bool **RedirectEdge** (Line *line, bool first, const Point *pt)
Redirect line from first or last subject that covers point pt.

void **Move** (MoveType) *move diagram in main window.*

void **SelectSubject** (Subject *s)

select all shapes of subject in current view.

void **SelectSubjects** (List<Subject *> *s)
select all shapes of subjects in list in current view.

virtual bool
SetText (TextShape *t, const string *s)
Update the text of text shape t and the corresponding subject.

void **FindAll** (const string *s, bool send, bool substring, bool nameOnly)

void **FindNext** (const string *s, bool send, bool substring, bool nameOnly)

void **ReplaceAll** (const string *s1, const string *s2, bool send, bool substring, bool nameOnly)

void **ReplaceNext** (const string *s1, const string *s2, bool send, bool substring, bool nameOnly)

virtual bool
HasIndexNode (int)

virtual bool
HasIndexShape (int)

void **Reindex** ()

void **SetNextIndex** (Node *n)
Give Node next index nr. and update shapes.

static Sequence
sequence

Protected Members

DiagramViewer*
GetDiagramViewer () const

int **GetNodeType** () const

void **SetNodeType** (int n)

int **GetEdgeType** () const

void **SetEdgeType** (int n)

int **GetNodeShapeType** () const

void **SetNodeShapeType** (int n)

void **SetLineType** (int n)

int **GetLineType** () const

LineStyle::Type
GetNodeLineStyle ()

LineStyle::Type
GetEdgeLineStyle ()

void **SetNodeLineStyle** (LineStyle::Type n)

void **SetEdgeLineStyle** (LineStyle::Type e)

LineEnd::Type
GetLineEnd1 ()

LineEnd::Type
GetLineEnd2 ()

void **SetLineEnd1** (LineEnd::Type t)

void **SetLineEnd2** (LineEnd::Type t)

bool **SetSubjectName** (Subject *subject, const string *s)
Sets name of subject to s. Checks if name is allowed.

bool **SetIndex** (Node *n, const string *index)
Check and set index of a node.

virtual void
PlaceShapes ()

void **LoadEntries** ()

void **SaveEntries** ()

virtual Thing*
CreateThing (int classNr)
Creates a new shape/subject depending on classNr.

void **UpdateNameStrings** (Subject *subject) *update all name strings of shapes representing subject.*

Private Members

DiagramViewer* **dviewer**

Graph* **graph**

AssocList* **assocList**

List <ShapeView *> **viewsRead**

unsigned long **maxid**

Subject* **annotationSubject**

void **FitShapes** () *fit shapes to drawing area + some extra edit space.*

void **PlaceViews** ()

void **PurgeViews** ()

void **CheckSubjects** ()

void **ReadThing** (int classNr, unsigned long value, bool dashedShape)

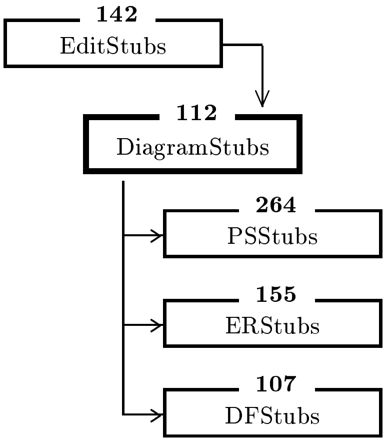
void **DeleteDiagram** ()

112

class **DiagramStubs** : public EditStubs

Callback stubs specific for all diagram editors.

Inheritance



Public Members

static void
SetNodeCB (Widget w, XtPointer num, XtPointer cbs)
Callbacks for setting current node and edge.

static void
SetEdgeCB (Widget w, XtPointer num, XtPointer cbs)

static void
SetCurveCB (Widget w, XtPointer num, XtPointer cbs)

static void
AlignNodeShapesCB (Widget, XtPointer, XtPointer)
Callbacks for the edit functions.

static void

AlignNodeShapesOKCB (Widget, XtPointer, XtPointer)

static void
AlignNodeShapesDefaultCB (Widget, XtPointer, XtPointer)

static void
DeleteCB (Widget, XtPointer, XtPointer)

static void
DeleteAllCB (Widget, XtPointer, XtPointer)

static void
CutCB (Widget, XtPointer, XtPointer)

static void
CopyCB (Widget, XtPointer, XtPointer)

static void
PasteCB (Widget, XtPointer, XtPointer)

static void
DuplicateCB (Widget, XtPointer, XtPointer)

static void
SelectAllCB (Widget, XtPointer, XtPointer)

static void
SameSizeCB (Widget, XtPointer, XtPointer)

static void
AnnotateCB (Widget, XtPointer, XtPointer)

static void
SubjectAnnotationOKCB (Widget, XtPointer, XtPointer)

static void
ShowIndexesCB (Widget w, XtPointer, XtPointer)

static void
ZoomInCB (Widget, XtPointer, XtPointer)
Callbacks for view functions.

static void
ZoomOutCB (Widget, XtPointer, XtPointer)

static void

TopLevelCB (Widget, XtPointer, XtPointer)

static void
ReindexCB (Widget w, XtPointer, XtPointer)

Callbacks for properties functions.

static void
DefaultNodeLineStyleCB (Widget, XtPointer, XtPointer)

static void
DefaultEdgeLineStyleCB (Widget, XtPointer, XtPointer)

static void
UpdateNodeShapeTypeCB (Widget, XtPointer, XtPointer)

static void
ConvertNodeShapeTypeCB (Widget, XtPointer, XtPointer)

static void
ConvertNodeShapesOKCB (Widget, XtPointer, XtPointer)

static void
ConvertNodeShapesDefaultCB (Widget, XtPointer, XtPointer)

static void
UpdateLineEndCB (Widget, XtPointer, XtPointer)

static void
UpdateLineEndDefaultCB (Widget, XtPointer, XtPointer)

static void
UpdateLineEndOKCB (Widget, XtPointer, XtPointer)

static void
UpdateCurveCB (Widget, XtPointer, XtPointer)

static void

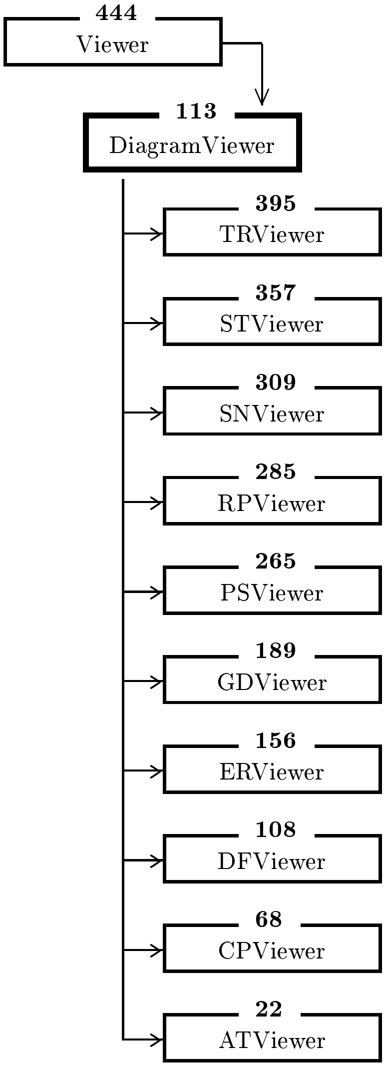
UpdateTextUnderlineCB (Widget, XtPointer, XtPointer)

113

class **DiagramViewer** : public Viewer

diagram viewer class, responsible for the contents of the drawing area.

Inheritance



Public Members

	DiagramViewer (Config *c, DiagramWindow *w)
virtual	**~DiagramViewer** ()
void	**Initialize** ()
void	**SetDiagram** (Diagram *d)
Diagram*	**GetDiagram** () const
void	**SetGrafport** (Grafport *g)
Buffer*	**GetBuffer** () const
Grid*	**GetGrid** () const
int	**GetDefaultNodeType** () const
int	**GetDefaultEdgeType** () const
int	**GetDefaultNodeShapeType** () const
int	**GetDefaultLineType** () const
bool	**IsDefaultCurve** () const
void	**SetDefaultNodeType** (int n)
void	**SetDefaultEdgeType** (int n)
void	**SetDefaultNodeShapeType** (int n)
void	**SetDefaultLineType** (int n)
void	**SetDefaultCurve** (bool s)
void	**SetShowIndexes** (bool b)
bool	**IsShowIndexes** ()
LineStyle::Type	**GetDefaultNodeLineStyle** () const
LineStyle::Type	**GetDefaultEdgeLineStyle** () const
void	**SetDefaultNodeLineStyle** (LineStyle::Type l)
void	**SetDefaultEdgeLineStyle** (LineStyle::Type l)
LineEnd::Type	**GetDefaultLineEnd1** ()
LineEnd::Type	

	GetDefaultLineEnd2 ()	
void	**SetDefaultLineEnd1** (LineEnd::Type t)	
void	**SetDefaultLineEnd2** (LineEnd::Type t)	
int	**Snap** (int i) const	*forwarding to grid object.*
void	**Snap** (Point *p)	
void	**SetStatus** (const char *msg)	
void	**SetStatus** (const string *msg)	
void	**SetCursor** (int cursor)	
void	**SetSizeCursor** (int cursor)	
virtual void	**Draw** ()	*Draw all shapes of current view to grafport.*
virtual void	**Undraw** ()	*Erase all shapes of current view from grafport.*
void	**Redraw** ()	*Redraw everything including grid.*
ShapeView*	**GetCurView** () const	*return current view.*
void	**DeleteAllViews** ()	
unsigned int	**NumberOfViews** () const	
bool	**AddView** (ShapeView *v)	
void	**SetView** (ShapeView *v)	
void	**UpdateView** (ShapeView *v)	
bool	**HasView** (const string *index)	
bool	**HasView** (ShapeView *v)	
ShapeView*	**GetView** (const string *index)	
void	**CheckShapes** ()	*Check all shapes in all views.*
void	**WriteShapes** (OutputFile *f)	*write all shapes from all views to ofile.*
void	**WriteSelection** (OutputFile *f)	

write all selected shapes from current view to ofile.

void **Select** (int x, int y)
perform the button1 functions (includes creating and moving shapes).

void **Adjust** (int x, int y)
Perform the 2nd button function: (de)selection.

Command*
Drag (int x, int y)
drag shapes and handles of line-segments.

Command*
Connect (int x, int y)
create an edge between nodes.

void **MovingPointer** (int x, int y)
change cursor to show the possible command (resize, move or edit).

void **KeyTyped** (int x, int y, int key)
start textmode, or (if in textmode) handle key. (x, y) is position.

void **TextModeOff** () *stop being in editing mode.*

void **DeselectAll** () *deselect all shapes in all views in the viewer.*

GShape* **GetShape** (Subject *subject)
return first shape from shapes having given subject (in any view).

void **GetShapes** (Subject *subject, List<GShape *> *shapes)
put in shapes all shapes from all views having given subject.

virtual void
Cut () *Cut subjects/shapes cmd from current view to buffer.*

virtual void
Copy () *Copy subjects/shapes cmd from current view to buffer.*

virtual void

Paste () *Paste from buffer to current view and graph.*

virtual void
Duplicate () *Node shapes in selection are duplicated, pasted in current view.*

virtual void
AlignNodeShapes ()
Align selection in current view according to given align type.

virtual void
AlignNodes (NodeAlign::Type alignment)
Align selection in current view according to given align type.

virtual void
UpdateCurve () *Convert selected curves from/to selected segmented lines.*

virtual void
UpdateTextUnderline ()
Set/Unset the text underlining of the names of the selected shapes.

virtual void
FindAll (const string *s, bool sens, bool substr, bool nameOnly)
select all shapes having string s (sens case sensitive).

virtual void
FindNext (const string *s, bool sens, bool substr, bool nameOnly)
select next shape having string s, and set cursor before it.

virtual void
ReplaceNext (const string *s1, const string *s2, bool sens, bool substr, bool nameOnly)
replace string in next shape in current view.

virtual void

ReplaceAll (const string *s1, const string *s2, bool sens, bool substr, bool nameOnly)
replace all strings in current view.

virtual void **Annotate** () *edit annotation of first selected subject in current view.*

virtual void **DeleteShapes** () *Delete shapes (and event. subjects) from selection.*

virtual void **DeleteAll** () *All shapes (and subjects) are deleted from current view.*

virtual void **SameSize** () *makes the size of selection shapes equal to size of the first.*

virtual void **MoveAllShapes** (Document::MoveType t)
move all shapes in current view according to move type.

virtual void **SelectAll** () *select all shapes in the current view.*

void **UpdateTextAlignmentSelection** (TextAlign::Type a)
update text alignment of selection in current view.

void **UpdateText** (TextShape *t, const string *newtext, const string *oldtext)
update text of textshape.

void **UpdateLineWidthSelection** (unsigned i)
update line width of selected shapes.

void **UpdateLineStyleSelection** (LineStyle::Type s)
update line style of selected shapes.

void **UpdateFontSelection** (int family, int style, int size) *update the fonts of selected shapes.*

void **UpdateLineColorSelection** (const string *color)
update the colors of the lines of the selected shapes.

void **UpdateTextColorSelection** (const string *color)
update the colors of the texts of the selected shapes.

void **UpdateFillColorSelection** (const string *color, FillStyle::Type f)
update the fill colors of the selected shapes.

void **CalcSizeElements** (Point &topLeft, Point &bottomRight)
calculate size of shapes in current view.

void **CalcSizeSelection** (Point &topLeft, Point &bottomRight)
calc size of selection in current view.

void **CalcSizeShapes** (List<GShape *> *shapes, Point &topLeft, Point &bottomRight)
calculate covered area of shapes in grafport.

void **ShapePositionUpdate** (GShape *shape)
for each line connecting shape, position is propagated.

void **UpdateDuplicationMarks** (List<Subject *> *s)
Update the duplication marks of shapes of given subjects.

void **UpdateDuplicationMarks** (Subject *s)
Update the duplication marks of shapes of given subject.

void **UpdateMoveShapes** (List<GShape *> *shapes, const Point *delta)
shapes are moved delta and result is propagated to drawing area.

void **MoveShapes** (List<GShape *> *shapes, const Point *delta) *all shapes are moved delta (result not propagated).*

virtual void
UpdateNodeShapeType (int)
convert selected node shapes to given code

void **ConvertNodeShapes** ()
convert selected node shapes.

virtual void
DeleteSubjects ()

void **SetIndexes** (bool b)

void **Reindex** (List<Node *> *nodes)

virtual void
ZoomIn ()

void **ZoomOut** ()

void **TopLevel** ()

LineEndDialog*
GetLineEndDialog ()

void **UpdateLineEnd** ()

void **ShowDefaultLineEnd** ()

void **UpdateLineEndSelection** (LineEnd::Type e1, LineEnd::Type e2)

NodeAlignmentDialog*
GetNodeAlignmentDialog ()

void **ShowDefaultNodeAlignment** ()

NodeShapeDialog*
GetNodeShapeDialog ()

void **ShowDefaultNodeShape** ()

Protected Members

List <ShapeView *> *
GetViews () const

void **EmptySelectionMessage** ()

error message because of empty selection.

void **EmptyShapesMessage** ()
error message because there are no shapes.

Private Members

List <ShapeView *> * **views** *The different views.*

ShapeView* **curView** *The current view that is displayed.*

Buffer* **buffer** *buffer object for cut, copy and paste.*

Grid* **grid** *object for aligning shapes to a grid.*

Diagram* **diagram**

int **defaultNodeType**

int **defaultNodeShapeType**

int **defaultEdgeType**

int **defaultLineType**

bool **defaultCurve**

bool **showIndexes**

LineStyle::Type **defaultNodeLineStyle**

LineStyle::Type **defaultEdgeLineStyle**

LineEnd::Type **defaultLineEnd1**

LineEnd::Type **defaultLineEnd2**

LineEndDialog*

lineEndDialog

NodeAlignmentDialog*
alignDialog

NodeShapeDialog*
nodeShapeDialog

void **CreateNode** (int x, int y)
create a node in current view.

bool **GoIntoTextMode** (GShape *shape, int x, int y)
if textshape at (x, y), edit it.

bool **TextMode** (TextShape *)
start editing textshape.

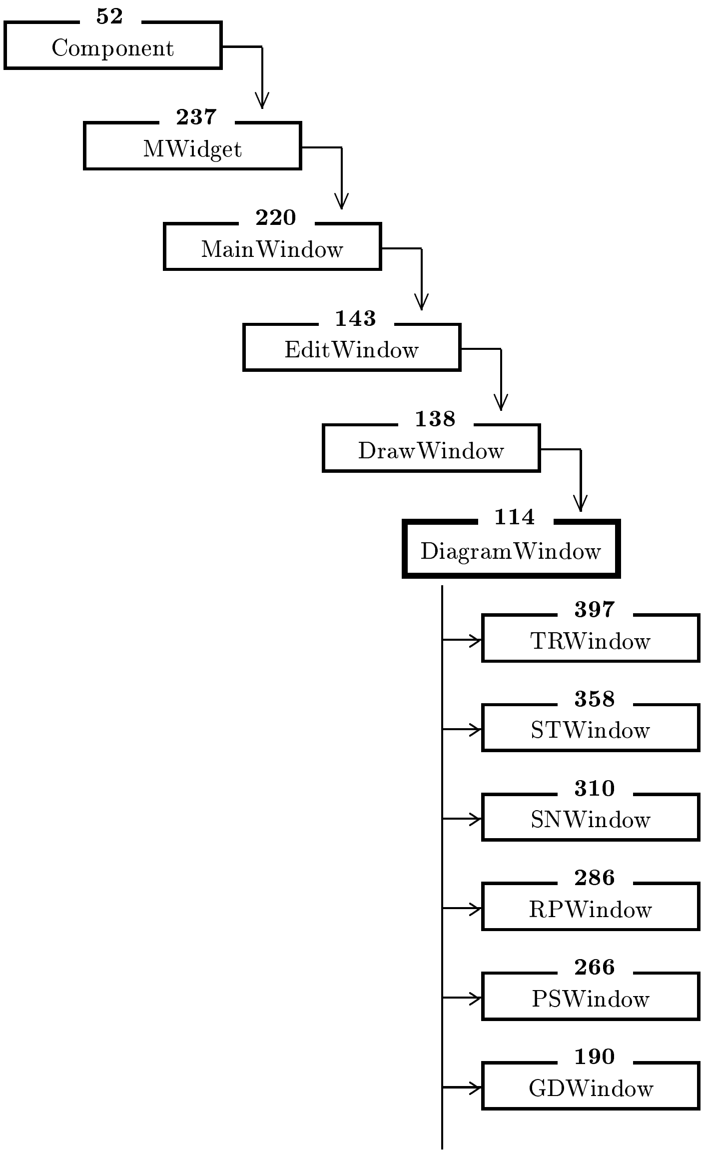
void **SetCursor** (int x, int y)
if editing and text shape is hit: set edit cursor.

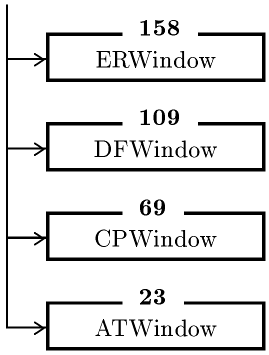
114

class **DiagramWindow** : public DrawWindow

edit window class that is common to all diagram editors.

Inheritance





Public Members

	DiagramWindow (const char *name)	
virtual	**~DiagramWindow** ()	
void	**SetNodeName** (int n)	*Show in status that node type n is selected.*
void	**SetEdgeName** (int n)	*Show in status that edge type n is selected.*

Protected Members

const char**	**nodeNames**	*points to the names of the node shapes.*
const char**	**edgeNames**	*points to the names of the edge shapes.*
int	**nrNodes**	
int	**nrEdges**	
const Bitmap*	**nodeIcons**	*List of Icons for the "Nodes" tiled buttons.*
const Bitmap*	**edgeIcons**	*List of Icons for the "Edges" tiled buttons.*
Widget	**CreateMenuBar** (Widget parent)	

create menu bar

virtual Widget
CreateTiles (Widget parent)
created tiled buttons.

void **CreateNodeIcons** (Widget parent)

void **CreateEdgeIcons** (Widget parent)

void **CreateIndexToggle** (Widget parent)

DiagramViewer*
GetDiagramViewer () const

void **SetShowLineStyleOptionMenu** (bool b)

Widget **GetNodeIconsRC** () const

Widget **GetEdgeIconsRC** () const

void **CreateLineStyleOptionMenu** (Widget w,
bool nodes,
bool minimize)

void **InitMenuItems** ()

virtual bool
ShowIndexToggle ()

void **CalcIconColumns** ()

int **GetIconsScrollHeight** (int rows)

Menu* **gridMenu**

MenuItem*
diagramAlignItems

MenuItem*
defPropertiesItems

MenuItem*
gridItems

MenuItem*
nodeLineStyleItems

MenuItem*
edgeLineStyleItems

MenuItem*
updateNodeShapeTypeItems

Private Members

DiagramViewer*
viewer

void **ParseUserOptions** (int argc, char** argv)
parse the command line options.

void **WriteUsage** (const char *prog)

void **CreateComponents** (Widget window)
initialize window: create menubar, tiles, scrolled window, etc

virtual Menu*
BuildViewMenu (Widget bar)

Widget **tiles** *Motif RowColumn widget containing tiled buttons.*

void **Finalize** ()

void **CreateDocument** ()

virtual DiagramViewer*
CreateViewer (Config *c)

virtual Graph*
CreateGraph ()

virtual Diagram*
CreateDiagram (Config *c, DiagramViewer *v, Graph *g)

virtual void
DetermineIcons ()
Determine icon lists to be used for the window.

void **GetNodeName** (int i, string *name)

void **GetEdgeName** (int i, string *name)

static const Bitmap
CURVE_ICON

static const Bitmap
SOLID_LINE_ICON

static const Bitmap
DASHED_LINE_ICON

static const Bitmap
DOTTED_LINE_ICON

Widget **edgeIconsRC**

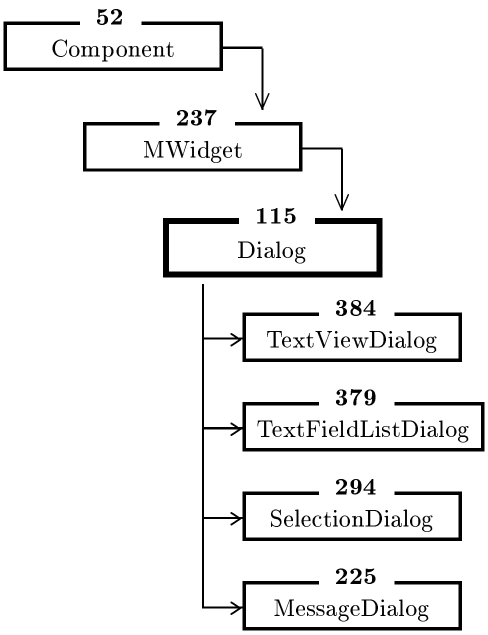
Widget	**nodeIconsRC**
int	**iconColumns**
bool	**showLineStyleOptionMenu**

115

class **Dialog** : public MWidget

abstract dialog class.

Inheritance



Public Members

	Dialog (Widget parent)
virtual	**~Dialog** ()
const char*	**GetClassName** ()
void	**Initialize** ()
virtual void	**SetOKCallback** (XtCallbackProc fun, XtPointer clientData)
virtual void	

SetCancelCallback (XtCallbackProc fun, XtPointer clientData)

virtual void
SetHelpCallback (XtCallbackProc fun, XtPointer clientData)

void **SetOKButtonLabel** (const char *txt)

void **SetCancelButtonLabel** (const char *txt)

void **SetHelpButtonLabel** (const char *txt)

void **ManageOKButton** (bool)

void **ManageCancelButton** (bool)

void **ManageHelpButton** (bool)

virtual void
SetAutoDestroy (bool b)
delete itself when unmapped.

virtual void
Popup () *pop up dialog.*

virtual void
Popdown () *pop down dialog.*

void **SetTitle** (const char *title)
set title string.

void **SetTitle** (const string *title)

void **GetTitle** (char *title)
get title string.

void **GetTitle** (string *title)

void **SetModal** (bool b)
make dialog application modal.

void **SetAutoUnmanage** (bool b)

bool **GetAutoUnmanage** ()

void **SetCenteredOnPointer** (bool b)

static void
UnmanageCB (Widget, XtPointer, XtPointer)

static void

DestroyCB (Widget, XtPointer, XtPointer)

static void
DestroyWidgetCB (Widget, XtPointer, XtPointer)

static void
TraverseNextCB (Widget, XtPointer, XtPointer)

static void
TraverseNextGroupCB (Widget, XtPointer, XtPointer)

static const unsigned
MAXTITLE

Protected Members

Widget **GetParent** ()

void **ManageWidget** (Widget, bool)

virtual void
CreateWidget ()

virtual void
Configure ()

void **ManageCenteredOnPointer** ()

Private Members

Widget **parent**

bool **autoDestroy**

bool **autoUnmanage**

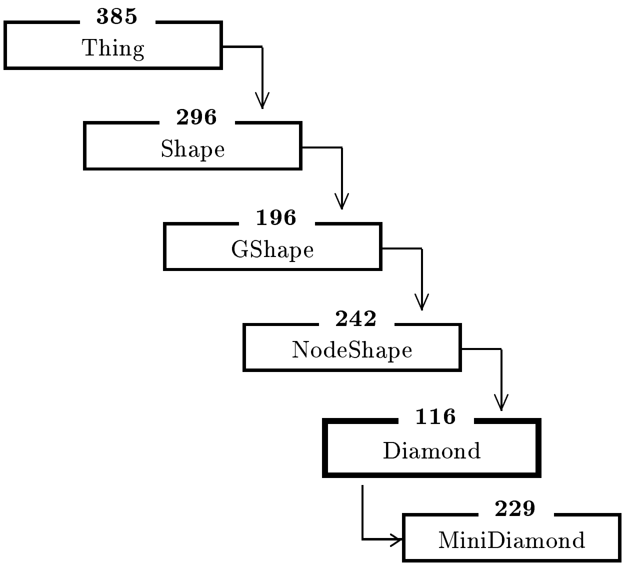
bool **centeredOnPointer**

116

class **Diamond** : public NodeShape

diamond shape class.

Inheritance



Public Members

static const int
DIAMOND_WIDTH

static const int
DIAMOND_HEIGHT

Diamond (ShapeView *v, Grafport *g, int x, int y, int w = DIAMOND_WIDTH, int h = DIAMOND_HEIGHT)

Diamond (const Diamond &)

Diamond (const NodeShape &)

virtual **~Diamond** ()

Shape* **Clone** ()

void	**DrawOutLine** (const Point *center)	*draws dotted diamant snapped to the grid.*
Point	**GiveSnp** (Point to) const	
Point	**GiveSnp** (Point p, int order, int count, int &code)	
bool	**ContainsPt** (int x, int y)	
int	**GetClassType** () const	
static const int	**DIAMOND_HANDLES**	

Protected Members

void	**DrawShape** ()	*draws a diamond*
void	**CalcPoints** (int x, int y)	
const Point*	**GetPoints** () const	

Private Members

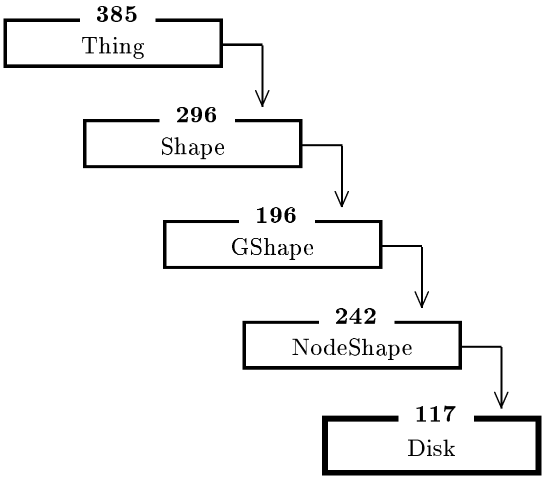
Point*	**points**

117

class **Disk** : public NodeShape

Disk shape class.

Inheritance



Public Members

static const int
DISK_WIDTH *default width*

static const int
DISK_HEIGHT *default height*

static const double
ELLIPSE_FACTOR
relative size of the ellipse.

Disk (ShapeView *v, Grafport *g, int x, int y,
int w = DISK_WIDTH,
int h = DISK_HEIGHT)

Disk (const NodeShape &)

Shape* **Clone** ()

void **DrawOutLine** (const Point *center)

draw disk-form dotted outline snaps to the grid.

Point **GiveSnp** (Point to) const
See Shape::

Point **GiveOrthoSnp** (Point to) const
See Shape::

Point **GiveSnp** (Point p, int order, int count, int &code)
order and count are for multiple edges. code for horz or vert point.

int **GetClassType** () const

static const int **DISK_HANDLES**
default nr. of handles.

Protected Members

void **DrawShape** () *draws a disk*

Point **GiveBoxOrthoSnp** (Point to) const

void **SetPosition** (const Point *position, bool snap=True)

void **SetSize** (int w, int h)

void **SetTextShape** ()

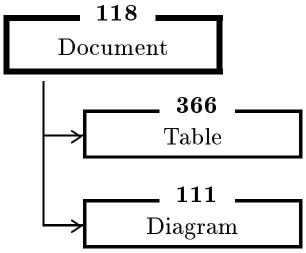
void **CalcPositionName** ()

118

class **Document**

(abstract) document class; generalization of diagram, tree, table, etc.

Inheritance



Public Members

	Document (Config *c, EditWindow *d, Viewer *v)	
virtual	**~Document** ()	
void	**New** ()	*opens new document called untitled.'suffix'.*
virtual void	**Load** ()	*popup file selector dialog for loading a file.*
virtual void	**Append** ()	*same as load but old document remains unchanged.*
void	**Save** ()	*save document to file.*
void	**SaveAs** ()	*popup file selector dialog for saving to a file.*
void	**SaveSelectionAs** ()	*popup file selector dialog for saving selection to a file.*
void	**Quit** ()	*closes document (asks confirmation) and quits application.*
void	**ShowDocInfo** ()	*popup info dialog containing information about this document.*

void	**ShowSource** ()	*popup text edit dialog with document source from file.*
void	**SaveSource** ()	*save edited source back to file.*
virtual void	**ShowSummary** ()	*popup info dialog containing summary of contents of this document.*
virtual void	**CheckDocument** ()	*popup warning dialog containing soft constraint violations (syntactic)*
virtual void	**ModelCheckDocument** ()	*popup dialog to model check the document (semantic check)*
void	**Annotate** ()	*popup text edit dialog in which doc. annotation can be given.*
void	**LoadSave** (const string *file)	*called from file selector dialog callbacks.*
virtual void	**DoLoad** (const string *file)	*load document from file.*
void	**DoSaveAs** (const string *file)	
void	**DoSaveSelectionAs** (const string *file)	
double	**GetLoadFormat** () const	*return format that was found in file that is read in.*
virtual void	**Initialize** ()	*perform some document type specific initialization.*
void	**NewFile** (const string *file)	*Checks if file is legal and opens new document with file as name.*
virtual void	**RemoveAll** ()	*Delete contents of the document.*
bool	**NewName** (const string *newname)	

Checks if newname is legal, and if so, changes name of document.

void **SetName** (const string *newname)

const string*
GetName () const

const string*
GetDocType () const

const string*
GetSuffix () const

const string*
GetToolName () const

const string*
GetAuthor () const

const string*
GetCreationTime () const

bool **IsDefault** () *return whether doc.name is the default name.*

void **NewDir** (const string *newdir)
Checks if newdir is legal, and if so, changes current directory.

void **SetDir** (const string *newdir)

const string*
GetDir () const

void **SetFSDir** () *Set dir from file selector.*

void **IncChanges** () *increases number of changes.*

void **DecChanges** () *decrease number of changes.*

int **GetChanges** () const

virtual void

SetInlineEdit (bool)
set inline edit (doc. type specific).

enum **MoveType**

virtual void
Move (MoveType) *move document in mainwindow.*

bool **IsModified** () const

void **SetModified** (bool set)

bool **IsLoaded** () const

bool **MayWrite** (const string *path)
ask user to confirm that a file may be overwritten

void **SetStatus** (const char *msg)
set main window status message to msg.

void **SetStatus** (const string *msg)

void **SetAnnotation** (const string *s)
set document annotation text to s.

virtual void
FindAll (const string *, bool, bool, bool)
find all strings.

virtual void
FindNext (const string *, bool, bool, bool)
find next string.

virtual void
ReplaceAll (const string *, const string *, bool, bool, bool)
replace all strings.

virtual void
ReplaceNext (const string *, const string *, bool, bool, bool)
replace next string.

virtual void
Find () *issue a popup dialog for getting a search string.*

virtual void

	Replace ()	*issue a popup dialog for search and replace strings.*
void	**UpdateWindow** ()	*update main window document text fields*
EditWindow*	**GetMainWindow** () const	
Config*	**GetConfig** () const	
Viewer*	**GetViewer** () const	
InputFile*	**GetInputFile** () const	
OutputFile*	**GetOutputFile** () const	
TextEditDialog*	**GetSourceEditDialog** () const	
TextEditDialog*	**GetDocAnnotationDialog** () const	
TextEditDialog*	**GetEltAnnotationDialog** () const	
FileSelectionDialog*	**GetFileSelectionDialog** () const	
FindDialog*	**GetFindDialog** () const	
ReplaceDialog*	**GetReplaceDialog** () const	

Protected Members

virtual void	**LoadEntries** ()	...
virtual void	**SaveEntries** ()	...
bool	**IsAppending** () const	
bool	**IsSaveSelection** () const	
virtual void		

	UpdateDirectory (const string *s)	*Set directory in all dialogs.*
void	**ShowDialog** (MessageDialog::DialogType, const char *title, const char *text)	
void	**ShowDialog** (MessageDialog::DialogType, const char *title, const string *text)	
void	**ReportCheck** (int errors, const string *msg)	*Give results of Check Document in popup dialog.*
string	**chkbuf**	*message string filled by Check Document.*
bool	**Save** (const string *file)	*function needed for saving to file.*

Private Members

EditWindow*	**mainwindow**
Viewer*	**viewer**
Widget	**mwwidget**
TextEditDialog*	**sourceEditDialog**
TextEditDialog*	**docAnnotationDialog**
TextEditDialog*	**eltAnnotationDialog**
FileSelectionDialog*	**fileSelectionDialog**
TextViewDialog*	**checkDialog**
TextViewDialog*	**infoDialog**
FindDialog*	**findDialog**
ReplaceDialog*	

	replaceDialog	
Config*	**config**	
InputFile*	**ifile**	
OutputFile*	**ofile**	
int	**changes**	*is document currently modified ?*
bool	**loaded**	*is document loaded or created.*
bool	**appending**	*is document loaded (overwritten) or appended.*
bool	**saveSelection**	*is selection saved or entire document;*
Toolkit::ToolType	**tool**	*current tool.*
string	**suffix**	*current document suffix*
string	**login**	*login of person who runs tool.*
string	**toolName**	*name of the current tool.*
string	**dir**	*current working directory.*
double	**fileFormat**	*current version of file format (format that will be written).*
string	**name**	*current document name.*
string	**docType**	*current document type.*
string	**author**	*initial author of the document.*
string	**createdOn**	*creation time and date.*
string	**fileName**	*file name from which document is loaded.*
double	**loadFormat**	*file format of loaded document.*
string	**generatedFrom**	*tool that wrote loaded document.*
string	**writtenOn**	*writing time and date of loaded document.*
string	**writtenBy**	*login that wrote document to file.*
string	**annotation**	*document annotation text.*
string	**saveName**	*file in which document is saved.*
void	**CreateDialogs** ()	*Create all popup dialog for document class.*

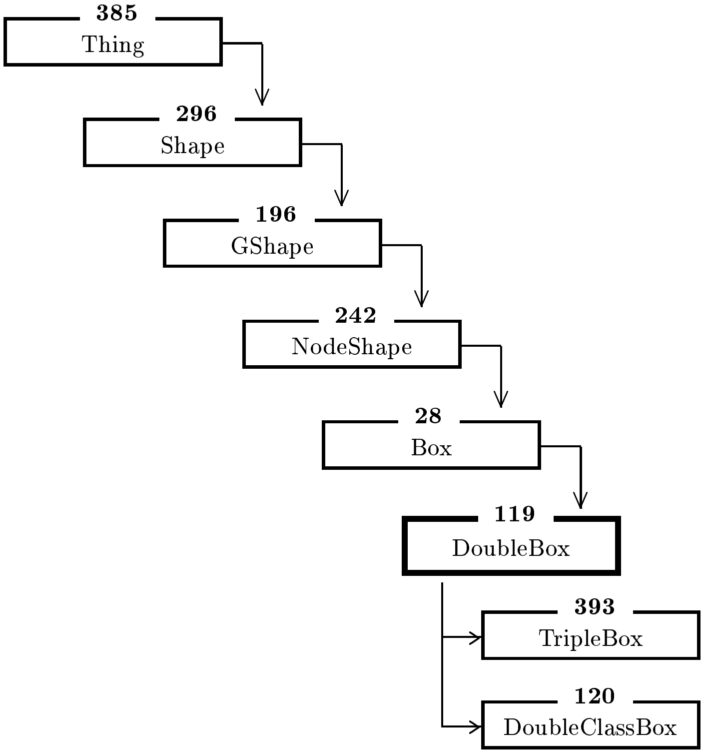
bool	**AskSave** ()	*function needed for loading document from file.*
bool	**Load** (const string *file)	...
bool	**LoadDocInfo** ()	...
bool	**CheckDocInfo** ()	...
virtual bool	**LoadHeader** ()	
virtual bool	**LoadEditInfo** ()	...
virtual bool	**CheckEditInfo** ()	...
bool	**CheckFormat** ()	...
bool	**CheckDocumentType** ()	...
virtual void	**SaveDocInfo** ()	...
virtual void	**SaveEditInfo** ()	...
void	**PrepareSaveDialog** ()	

119

class **DoubleBox** : public Box

(abstract) box with two compartments: 1st for name, 2nd for a label list.

Inheritance



Public Members

DoubleBox (ShapeView *v, Grafport *g, int x, int y, int w, int h)

DoubleBox (const DoubleBox &s)

DoubleBox (const Box &s)

DoubleBox (const NodeShape &s)

virtual **~DoubleBox** ()

int	**GetClassType** () const
Shape*	**Clone** ()
void	**SetSize** (int w, int h)
void	**SetPosition** (const Point *p, bool snap=True)
void	**SetFont** (XFont *ft)
void	**SetTextColor** (const string *color)
TextShape*	**ChooseTextShape** (int x, int y)
bool	**HasTextShape** (TextShape *t) const
void	**AdjustSize** ()
void	**UpdateLabel** (const string *s, unsigned n, bool update) *update or insert string of n-th label.*
TextShape*	**GetLabel** (unsigned n)
unsigned	**NrLabels** ()
bool	**SetAssocSubject** (AssocList *)
void	**SetTextShape** ()
void	**SetGrafport** (Grafport *g)
void	**SetView** (ShapeView *v)
bool	**HasString** (const string *s, bool sens, bool sub) const *contains s as string (case sensitive, substring).*
bool	**HasString** (const string*, bool, bool, List<TextShape *> *list)
LineStyle::Type	**GetSeparatorLineStyle** () const
void	**SetSeparatorLineStyle** (LineStyle::Type x)
int	**GetInitialHeight** () const

Protected Members

void	**DrawShape** ()

void **DrawTextShapes** ()

virtual int **RequiredHeight** ()

virtual void
CalcPositionLabels ()
calculates name position in 1st compartiment, labels in 2nd.

virtual void
AdjustSizes (const string *s)

List <TextShape *> *
GetLabels ()

static const int
TEXTMARGIN

string **labelDescription**

int **initialHeight**

Private Members

List <TextShape *> *
labels *the list of labels in the second compartiment.*

LineStyle::Type
separatorLineStyle
line style of separator line (between name and labels).

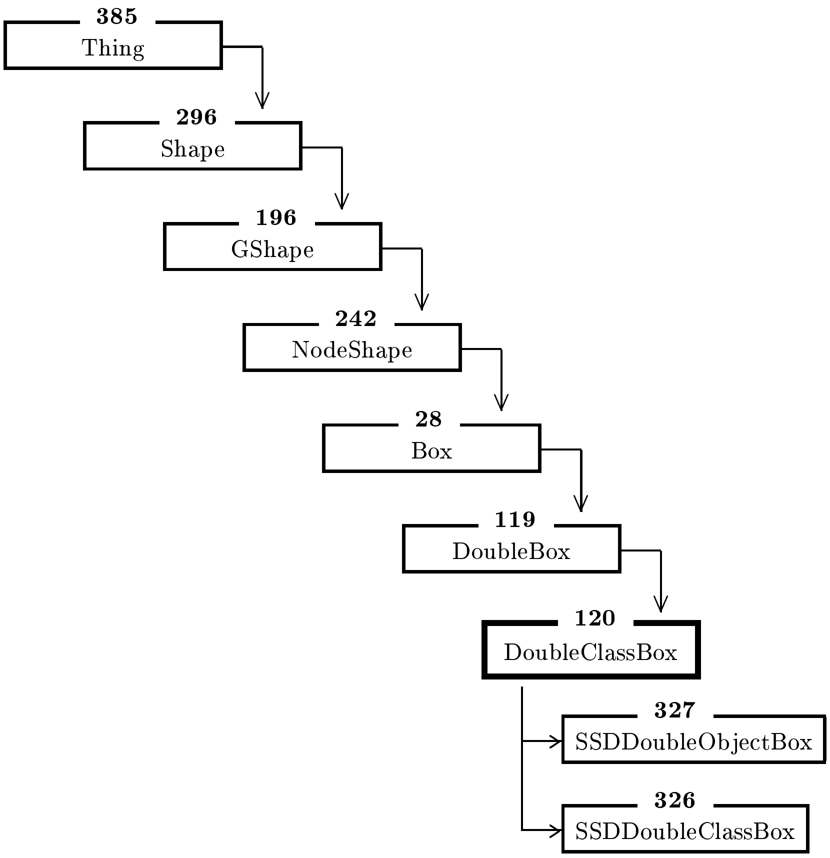
void **Init** ()

120

class **DoubleClassBox** : public DoubleBox

A double box used for as class or object box with attributes.

Inheritance



Public Members

static const int **DOUBLECLASSBOX_WIDTH**

static const int

DOUBLECLASSBOX_HEIGHT

DoubleClassBox (ShapeView *v, Grafport *g,
int x, int y, int w =
DOUBLECLASSBOX_WIDTH,
int h =
DOUBLECLASSBOX_HEIGHT)

DoubleClassBox (const Box &s)

Shape* **Clone** ()

bool **SetAssocSubject** (AssocList *)

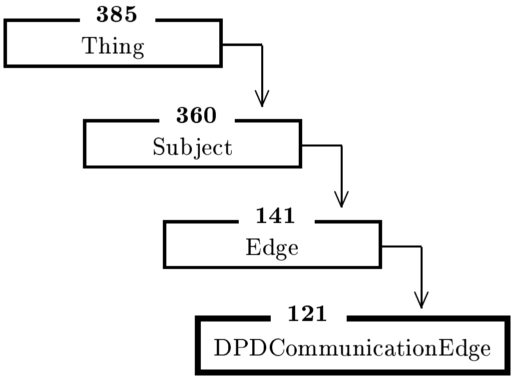
virtual void
SetTextShape ()

121

class **DPDCommunicationEdge** : public Edge

Component Dependency edge class

Inheritance



Public Members

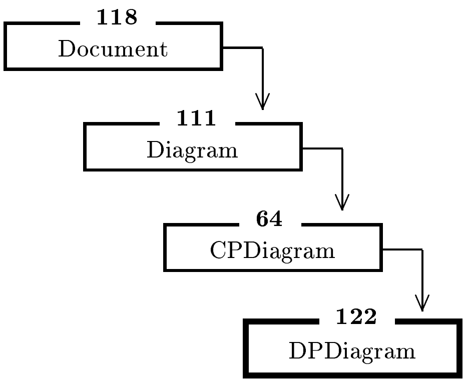
	DPDCommunicationEdge (CPGraph *g, Subject *s1, Subject *s2)
int	**GetClassType** () const
Subject*	**Clone** ()

122

class **DPDiagram** : public CPDiagram

deployment diagram class.

Inheritance



Public Members

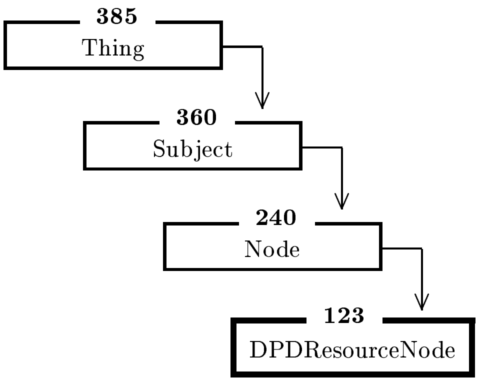
	DPDiagram (Config *, DPWindow *, CPViewer *, CPGraph *)
void	**UpdateNodeType** (int n)
void	**UpdateEdgeType** (int n)

123

class **DPDResourceNode** : public Node

UML Resource node class.

Inheritance



Public Members

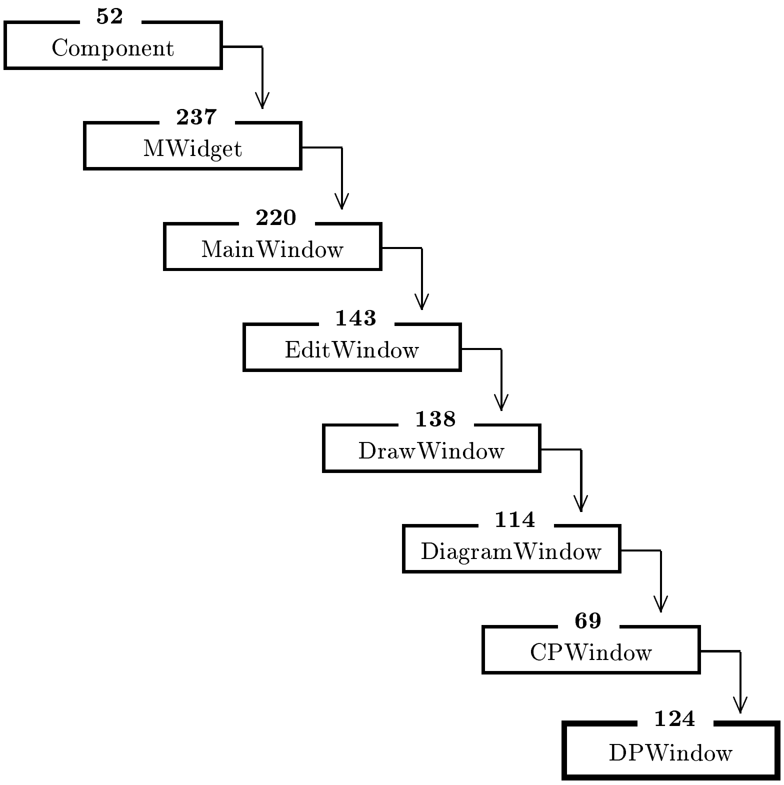
	DPDResourceNode (CPGraph *g)
int	**GetClassType** () const
Subject*	**Clone** ()

124

class **DPWindow** : public CPWindow

deployment diagram edit window class.

Inheritance



Public Members

DPWindow (const char *name)

Private Members

int	**GetTool** ()
Diagram*	**CreateDiagram** (Config *c, DiagramViewer *v, Graph *g)

void **DetermineIcons** ()

static const char*
DP_NODE_NAMES []

static const Bitmap
DP_NODE_ICONS []

static const char*
DP_EDGE_NAMES []

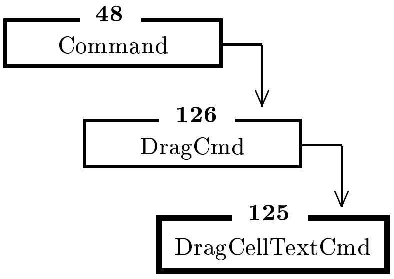
static const Bitmap
DP_EDGE_ICONS []

125

class **DragCellTextCmd** : public DragCmd

drag cell text in table command.

Inheritance



Public Members

	DragCellTextCmd (Table *t, TableViewer *v, CellText *c)
void	**Execute** ()
void	**UnExecute** ()

Private Members

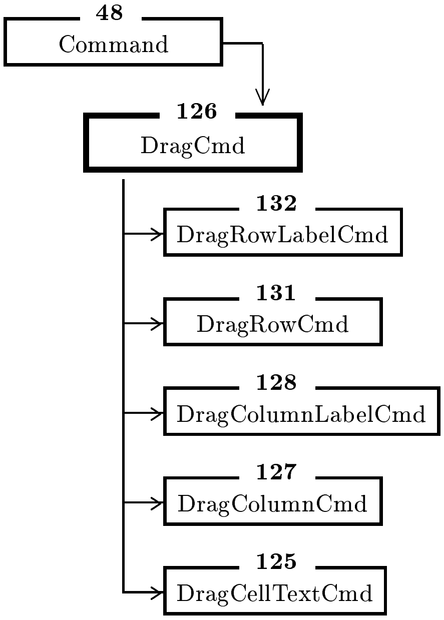
CellText*	**srcCellText**	
CellText*	**destCellText**	
string	**oldText**	
XFont*	**oldFont**	
bool	**copy**	*is cell text copied or moved ?*
const char*	**GetName** () const	
void	**DrawOutLine** (const Point *)	

126

class **DragCmd** : public Command

abstract table editor drag command.

Inheritance



Public Members

	DragCmd (Table *t, TableViewer *v)
void	**Execute** ()
void	**UnExecute** ()
void	**TrackMouse** (TrackType, Point*, Point*, Point*)
const char*	**GetName** () const

Protected Members

TableViewer*

	GetTableViewer () const
Point*	**GetOldPosition** ()
void	**SetOldPosition** (const Point *p)
Point*	**GetDelta** ()

Private Members

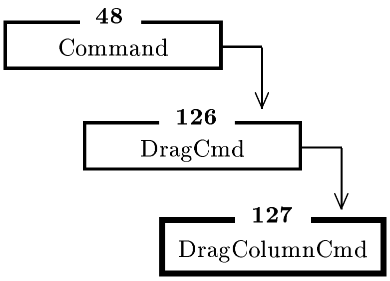
TableViewer*	**tviewer**
Point	**oldPosition**
Point	**delta**
virtual void	**DrawOutLine** (const Point *)

127

class **DragColumnCmd** : public DragCmd

drag (resize) column in table command.

Inheritance



Public Members

	DragColumnCmd (Table *t, TableViewer *v, LinePiece *l)
void	**Execute** ()
void	**UnExecute** ()
const char*	**GetName** () const

Private Members

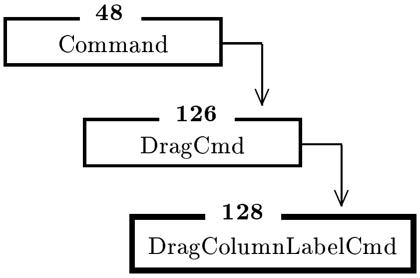
CellColumn*	**column**
int	**oldSize**
void	**DrawOutLine** (const Point *)

128

class **DragColumnLabelCmd** : public DragCmd

drag column label in table command (move column).

Inheritance



Public Members

	DragColumnLabelCmd (Table *t, TableViewer *v, RowColumnLabel *rc)
void	**Execute** ()
void	**UnExecute** ()
const char*	**GetName** () const

Private Members

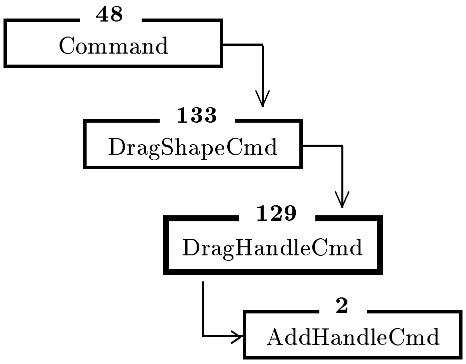
int	**oldNr**
int	**newNr**
CellColumn*	**column**
RowColumnLabel*	**label**
void	**DrawOutLine** (const Point *)

129

class **DragHandleCmd** : public DragShapeCmd

command class for dragging line handles.

Inheritance



Public Members

	DragHandleCmd (Line *line, unsigned which)
void	**Execute** ()
void	**UnExecute** ()
void	**Executable** ()
void	**DrawOutLine** (const Point *delta)
const char*	**GetName** () const

Protected Members

void	**Erase** ()
void	**Redraw** ()
void	**ReverseDelta** ()
Point*	**GetOldPt** ()
Point*	**GetNewPt** ()

Line*	**GetLine** ()
void	**DrawLines** (const Point *p1, const Point *p2, const Point *p3)
unsigned	**GetHandleNr** ()
virtual void	**CalcPoints** ()

Private Members

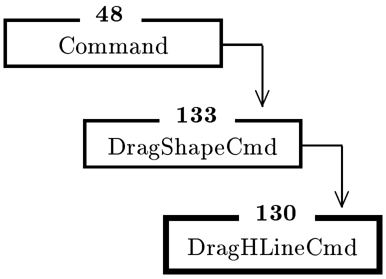
Point	**oldPt**
Point	**newPt**
Line*	**line**
unsigned	**handleNr**

130

class **DragHLineCmd** : public DragShapeCmd

command for dragging transition line (STD).

Inheritance



Public Members

	DragHLineCmd (TransitionArrow *shape)
void	**Executable** ()
void	**UnExecute** ()
const char*	**GetName** () const

Protected Members

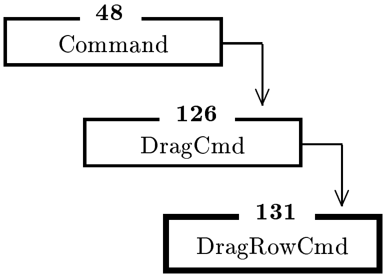
void	**DrawOutLine** (const Point *d)

131

class **DragRowCmd** : public DragCmd

drag (resize) row in table command.

Inheritance



Public Members

DragRowCmd (Table *t, TableViewer *v, LinePiece *l)

void **Execute** ()

void **UnExecute** ()

const char* **GetName** () const

Private Members

CellRow* **row**

int **oldSize**

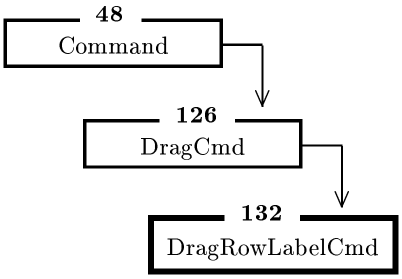
void **DrawOutLine** (const Point *)

132

class **DragRowLabelCmd** : public DragCmd

command for dragging the row labels of a table.

Inheritance



Public Members

	DragRowLabelCmd (Table *t, TableViewer *v, RowColumnLabel *rc)
void	**Execute** ()
void	**UnExecute** ()
const char*	**GetName** () const

Private Members

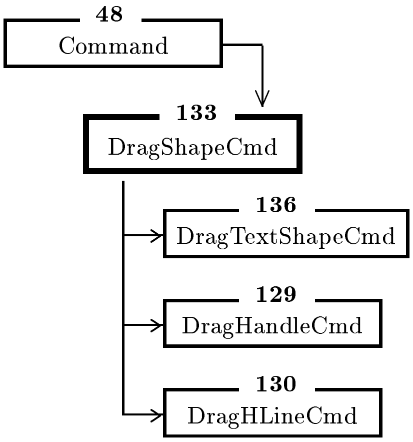
int	**oldNr**
int	**newNr**
CellRow*	**row**
RowColumnLabel*	**label**
void	**DrawOutLine** (const Point *)

133

class **DragShapeCmd** : public Command

command class for dragging a shape.

Inheritance



Public Members

	DragShapeCmd (Shape *shape)
void	**TrackMouse** (TrackType aPhase, Point *anchorPoint, Point *previousPoint, Point *nextPoint)
void	**Execute** ()
void	**UnExecute** ()
void	**ReExecute** ()
virtual void	**Executable** ()
const char*	**GetName** () const

Protected Members

virtual void
DrawOutLine (const Point *delta)

Shape* **GetShape** () const

ShapeView*
GetView () const

const Point*
GetDelta () const

void **SetDelta** (const Point *p)

const Point*
GetOldPosition () const

void **SetOldPosition** (const Point *p)

Private Members

Shape* **shape**

ShapeView*
view

Point **oldPosition**

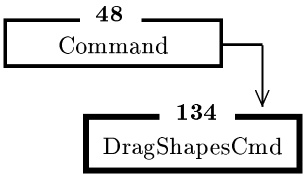
Point **delta**

134

class **DragShapesCmd** : public Command

command class for dragging multiple shapes.

Inheritance



Public Members

	DragShapesCmd (ShapeView *v)
virtual	**~DragShapesCmd** ()
void	**TrackMouse** (TrackType aPhase, Point *anchorPoint, Point *previousPoint, Point *nextPoint)
void	**Execute** ()
void	**UnExecute** ()
void	**ReExecute** ()
void	**Executable** ()
const char*	**GetName** () const

Protected Members

void	**DrawOutLines** (Point &delta)

Private Members

List <Point *> *

oldPositions

List <GShape *> *
shapes

Point **delta**

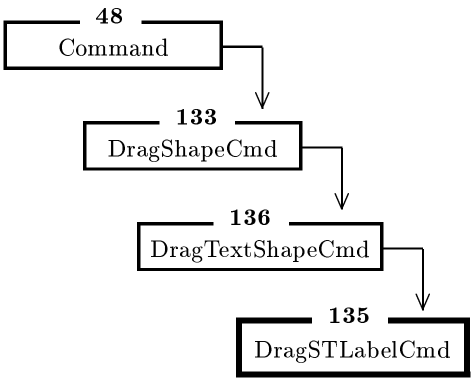
ShapeView*
view

135

class **DragSTLabelCmd** : public DragTextShapeCmd

command class for dragging text shapes in a STD.

Inheritance



Public Members

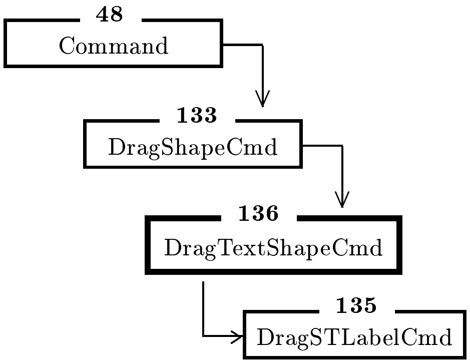
	DragSTLabelCmd (TextShape *shape)
void	**Executable** ()

136

class **DragTextShapeCmd** : public DragShapeCmd

command class for dragging text shapes.

Inheritance



Public Members

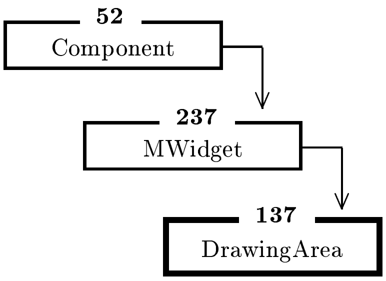
	DragTextShapeCmd (TextShape *shape)
void	**Executable** ()
const char*	**GetName** () const

137

class **DrawingArea** : public MWidget

wrapper class around the Motif drawing area widget.

Inheritance



Public Members

DrawingArea (Config *config, Widget parent, const char *name, DrawWindow *w)
Description determines the size, if empty the screen sz is used.

const char* **GetClassName** ()

void **SetViewer** (Viewer *v)

XGrafport* **GetGrafport** () const

int **GetWidth** () const

int **GetHeight** () const

void **UpdateSize** (int w, int h)

void **FitDocument** () *resize drawing area to fit the viewer's contents.*

void **FitDocument** (int wd, int ht)
resize drawing area to given size.

void **EnableUndo** (bool flag)

enable/disable edit popup menu entry.

void **EnableRedo** (bool flag)
enable/disable edit popup menu entry.

void **Enable** (const char *label, bool flag)
enable/disable popup menu entry.

void **SetUndoName** (const char *name)
change edit popup menu undo entry name.

void **SetRedoName** (const char *name)
change edit popup menu redo entry name.

Pixel **GetBackgroundColor** ()

Pixel **GetForegroundColor** ()

Private Members

DrawWindow* **drawwindow**

Viewer* **viewer**

XGrafport* **grafport**

static char **drawTranslations** []
translations used for drawing area callback.

void **CreatePopupMenu** (int tool)
create the popup edit menu as a child of the drawing area.

void **Dispatch** (XtPointer event)
Dispatch event: ButtonPress, ButtonRelease, KeyPress, MotionNotify.

void **Redraw** (XEvent *event)
Called after exposure (e.g. resulting from scrollbar movement)

void **PopupMenu** (XEvent *e)

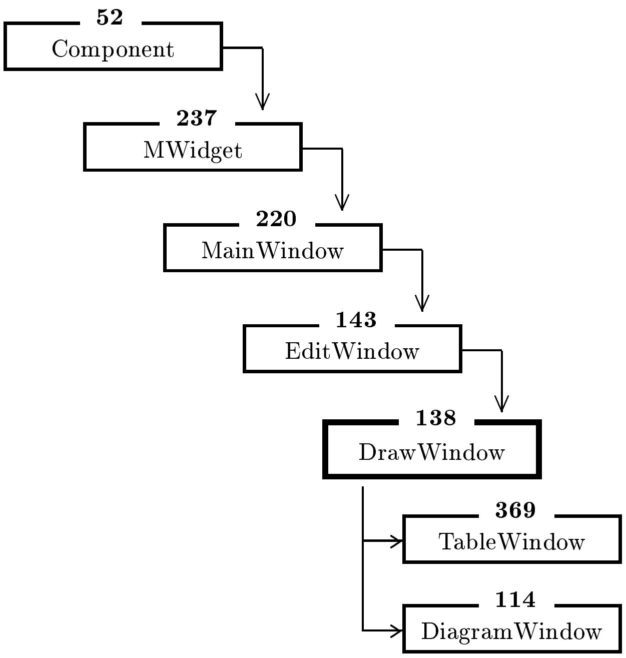
		when button-3 pressed this is called and edit menu is popup up.
int	**MapKeySym** (KeySym keysym)	*map keysym (from X KeyPress event) from non-ascii char to own code.*
Point	**anchorPoint**	*anchor points used for dragging.*
Point	**previousPoint**	*previous point used for dragging.*
Point	**nextPoint**	*next point used for dragging.*
bool	**beenDragging**	*busy dragging.*
char	**buf** [5]	*save last typed characters.*
int	**x**	*save last event x coordinate.*
int	**y**	*save last event y coordinate.*
Menu*	**popupMenu**	*a Motif PopupMenu widget.*
int	**width**	*width of the drawing area in pixels*
int	**height**	*height of the drawing area in pixels*
int	**maxWidth**	*maximum width of the drawing area in pixels*
int	**maxHeight**	*maximum height of the drawing area in pixels*

138

class **DrawWindow** : public EditWindow

(abstract) main window class common to all drawing editors.

Inheritance



Public Members

	DrawWindow (const char *name)
virtual	**~DrawWindow** ()
const char*	**GetClassName** ()
void	**SetAutoResize** (bool set)
void	**SetInlineEdit** (bool set)
void	**SetScaleValue** (const char *s) *set scale value string.*
DrawingArea*	

	GetDrawingArea ()
void	**FitDocument** ()
void	**FitDocument** (int wd, int ht)
void	**ShowPosition** (int x, int y)
void	**SetUndoName** (const char *s) *sets name of edit undo menu item.*
void	**SetRedoName** (const char *s) *sets name of edit redo menu item.*
void	**EnableUndo** (bool flag) *enables edit undo menu item.*
void	**EnableRedo** (bool flag) *enables edit redo menu item.*
void	**EnableDocumentSource** (bool flag)

Protected Members

virtual void	**WriteUsage** (const char *s)
void	**ParseUserOptions** (int argc, char **argv) *parse user options.*
Widget	**GetArrowButtons** ()
void	**SetArrowButtons** (Widget w)
Widget	**GetScrolledWindow** ()
void	**SetScrolledWindow** (Widget w)
Widget	**GetScaleValue** ()
void	**SetScaleValue** (Widget w)
void	**SetScrolledWindowWidth** (int n)
int	**GetScrolledWindowWidth** ()
void	**SetScrolledWindowHeight** (int n)
int	**GetScrolledWindowHeight** ()
void	**CreateDrawingArea** ()
void	**Finalize** ()

Widget	**CreateControlArea** (Widget parent)	
Widget	**CreateScrolledWindow** (Widget parent)	
Widget	**CreateScaleValue** (Widget parent)	
Widget	**CreateArrowButtonSquare** (Widget parent)	
void	**InitMenuItems** ()	
Menu*	**documentMenu**	
Menu*	**scaleMenu**	
MenuItem*	**scaleItems**	
MenuItem*	**documentItems**	
MenuItem*	**documentInfoItems**	

Private Members

DrawingArea*	**drawingArea**	*DrawingArea part of main window.*
Widget	**scrolledWindow**	*Motif ScrolledWindow widget.*
Widget	**horizontalScrollBar**	*Horiz. scrollbar of scrollwin.*
Widget	**verticalScrollBar**	*Vert. scrollbar of scrollwin.*
Widget	**scaleValue**	*Motif Label for scale value.*
Widget	**inlineEditToggle**	*Motif ToggleButton for inline edit.*
Widget	**autoResizeToggle**	*Motif ToggleButton for recomputing sizes.*
Widget	**arrowButtons**	*Motif Frame Widget for arrow buttons.*
int	**initDrawingWidth**	
int	**initDrawingHeight**	
int	**maxDrawingWidth**	
int	**maxDrawingHeight**	

static const int	**MIN_DRAWING_SIZE**	
int	**scrollWidth**	*width of the scrolled part in pixels.*
int	**scrollHeight**	*height of the scrolled part in pixels.*
void	**DetermineScrollSize** (ScreenType)	*Determine scrolled window size from X screen.*
string	**toolIntroMenuLabel**	

139

class **DumpWidgetTree**

recursively descend Widget tree and print children and their addresses.

Public Members

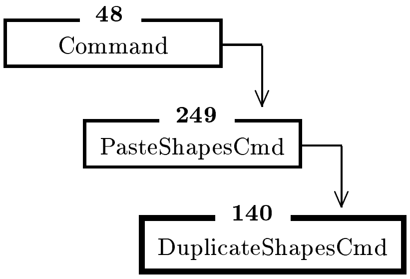
static void
Dump (Widget w)

140

class **DuplicateShapesCmd** : public PasteShapesCmd

command class for duplicating node shapes.

Inheritance



Public Members

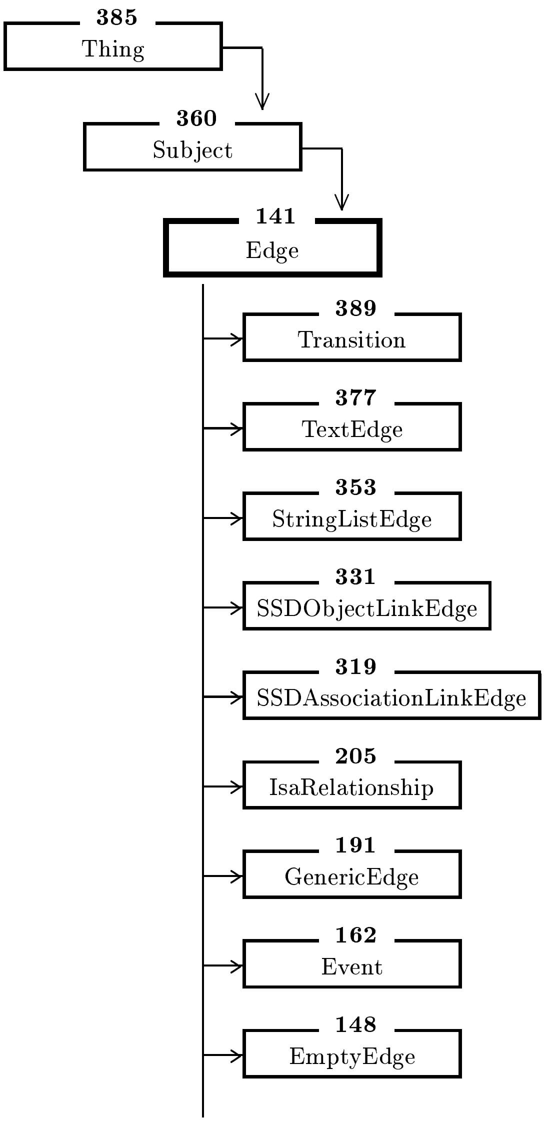
	DuplicateShapesCmd (ShapeView *v)
virtual	**~DuplicateShapesCmd** ()
void	**Abort** ()
void	**Execute** ()
void	**UnExecute** ()
const char*	**GetName** () const

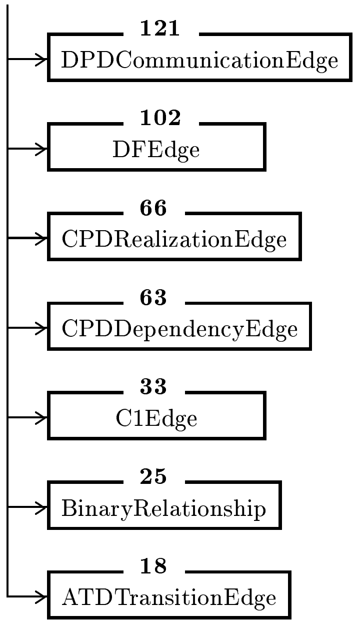
141

class **Edge** : public Subject

Graph edge class, base class of all edges. An edge connects two subjects.

Inheritance





Public Members

	Edge (Graph *g, Subject *n1, Subject *n2)	
virtual	**~Edge** ()	
Subject*	**GetSubject1** () const	*edge connects subject1 and subject2.*
Subject*	**GetSubject2** () const	*when edge is directed then subject1 is 'from' and subject2 'to'.*
void	**SetSubject1** (Subject *n)	
void	**SetSubject2** (Subject *n)	
bool	**IsDirected** () const	
bool	**InGraph** () const	*is edge part of the graph?*
bool	**IsEdge** () const	*yes of course it is*
bool	**PartOf** (Subject *subject) const	*is subject one of the two subjects that edge connects?*
void	**SetSubjects** (Subject *n1, Subject *n2, bool d = 0)	

set it all

void	**WriteMembers** (OutputFile *f)	*write attributes to file.*
bool	**ReadMembers** (InputFile *f, double format)	*read and check attributes from file.*
bool	**SetReferences** (AssocList *)	*set and check references to subjects after reading.*
bool	**CheckReferences** ()	*check whether subject references exist.*

Protected Members

void	**SetDirected** (bool b)

Private Members

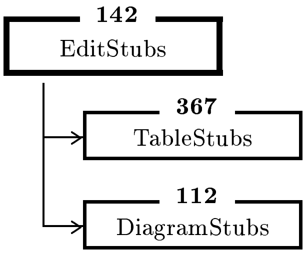
Subject*	**subject1**	*from subject.*
Subject*	**subject2**	*to subject.*
bool	**directed**	*is the edge directed?*
bool	**SetSubjectReference** (AssocList *al, Subject **subject, int which)	*part of SetReference.*

142

class **EditStubs**

static member functions as interface between callback and class.

Inheritance



Public Members

static void
HelpCB (Widget, XtPointer, XtPointer)
Callback for the help functions.

static void
LoadOKCB (Widget, XtPointer, XtPointer)
Callbacks for load and save.

static void
SaveAsOKCB (Widget, XtPointer, XtPointer)

static void
AnnotationOKCB (Widget, XtPointer, XtPointer)

static void
CancelFSCB (Widget, XtPointer, XtPointer)

static void
FindNextCB (Widget, XtPointer, XtPointer)
Callbacks for find/replace functions.

static void
FindAllCB (Widget, XtPointer, XtPointer)

static void

ReplaceFindCB (Widget, XtPointer, XtPointer)

static void
ReplaceNextCB (Widget, XtPointer, XtPointer)

static void
ReplaceAllCB (Widget, XtPointer, XtPointer)

static void
ChangeNameCB (Widget, XtPointer, XtPointer)
Callbacks for some editor properties.

static void
ChangeDirCB (Widget, XtPointer, XtPointer)

static void
SetModifiedCB (Widget, XtPointer, XtPointer)

static void
SetAutoResizeCB (Widget, XtPointer, XtPointer)

static void
SetInlineEditCB (Widget, XtPointer, XtPointer)

static void
NewCB (Widget, XtPointer, XtPointer)
Callbacks for file functions.

static void
LoadCB (Widget, XtPointer, XtPointer)

static void
AppendCB (Widget, XtPointer, XtPointer)

static void
SaveCB (Widget, XtPointer, XtPointer)

static void
SaveAsCB (Widget, XtPointer, XtPointer)

static void
SaveSelectionAsCB (Widget, XtPointer, XtPointer)

static void
QuitCB (Widget, XtPointer, XtPointer)

static void
FindCB (Widget, XtPointer, XtPointer)
Callbacks for search functions

static void

ReplaceCB (Widget, XtPointer, XtPointer)

static void
DocumentInfoCB (Widget, XtPointer, XtPointer)
Callbacks for document functions.

static void
DocumentSummaryCB (Widget, XtPointer, XtPointer)

static void
DocumentSourceCB (Widget, XtPointer, XtPointer)

static void
DocumentSourceOKCB (Widget, XtPointer, XtPointer)

static void
CheckDocumentCB (Widget, XtPointer, XtPointer)

static void
AnnotateDocumentCB (Widget, XtPointer, XtPointer)

static void
ModelCheckDocumentCB (Widget, XtPointer, XtPointer)

static void
MakeLargerCB (Widget, XtPointer, XtPointer)
Callbacks for scale functions.

static void
MakeSmallerCB (Widget, XtPointer, XtPointer)

static void
ScaleFactorCB (Widget, XtPointer, XtPointer)

static void
NormalScaleCB (Widget, XtPointer, XtPointer)

static void
WholeDrawingCB (Widget, XtPointer, XtPointer)

static void

ScaleFactorOKCB (Widget, XtPointer, XtPointer)

static void
MoveCB (Widget, XtPointer, XtPointer)
Callback for arrow buttons.

static void
SaveAsPSCB (Widget, XtPointer, XtPointer)
Callbacks for Printer functions.

static void
SaveAsEPSCB (Widget, XtPointer, XtPointer)

static void
SaveAsFigTexCB (Widget, XtPointer, XtPointer)

static void
SaveAsFigPSCB (Widget, XtPointer, XtPointer)

static void
SaveAsPNGCB (Widget, XtPointer, XtPointer)

static void
SetNameCB (Widget, XtPointer, XtPointer)

static void
SetPrintCommandCB (Widget, XtPointer, XtPointer)

static void
SetPreviewCommandCB (Widget, XtPointer, XtPointer)

static void
SetCopiesCB (Widget, XtPointer, XtPointer)

static void
ExportAsCB (Widget, XtPointer, XtPointer)
Callbacks for Printer functions.

static void
ShowPrintPreviewCB (Widget, XtPointer, XtPointer)

static void
PrintCB (Widget, XtPointer, XtPointer)

static void

ShowPrinterQueueCB (Widget, XtPointer, XtPointer)

static void
PrinterNameCB (Widget, XtPointer, XtPointer)

static void
PrinterCopiesCB (Widget, XtPointer, XtPointer)

static void
PrintCommandCB (Widget, XtPointer, XtPointer)

static void
PrinterQueueCommandCB (Widget, XtPointer, XtPointer)

static void
PrinterRemoveCommandCB (Widget, XtPointer, XtPointer)

static void
PreviewCommandCB (Widget, XtPointer, XtPointer)

static void
SetPrintColorsCB (Widget, XtPointer, XtPointer)

static void
SetTumbleCB (Widget, XtPointer, XtPointer)

static void
SetDuplexCB (Widget, XtPointer, XtPointer)

static void
BannerPageCB (Widget, XtPointer, XtPointer)

static void
ShowPageBoundaryCB (Widget, XtPointer, XtPointer)
Callbacks for Page functions.

static void
PageOrientationCB (Widget, XtPointer, XtPointer)

static void

PageSizeCB (Widget, XtPointer, XtPointer)

static void
ShowPageNumbersCB (Widget, XtPointer, XtPointer)

static void
ShowDocumentInfoCB (Widget, XtPointer, XtPointer)

Callbacks for Properties functions.

static void
DefaultLineWidthCB (Widget, XtPointer, XtPointer)

static void
DefaultLineWidthOKCB (Widget, XtPointer, XtPointer)

static void
DefaultLineWidthDefaultCB (Widget, XtPointer, XtPointer)

static void
UpdateLineWidthCB (Widget, XtPointer, XtPointer)

static void
UpdateLineWidthOKCB (Widget, XtPointer, XtPointer)

static void
UpdateLineWidthDefaultCB (Widget, XtPointer, XtPointer)

static void
UpdateLineColorCB (Widget, XtPointer, XtPointer)

static void
UpdateLineColorOKCB (Widget, XtPointer, XtPointer)

static void

UpdateLineColorDefaultCB (Widget, XtPointer, XtPointer)

static void
DefaultLineColorCB (Widget, XtPointer, XtPointer)

static void
DefaultLineColorOKCB (Widget, XtPointer, XtPointer)

static void
DefaultLineColorDefaultCB (Widget, XtPointer, XtPointer)

static void
UpdateTextColorCB (Widget, XtPointer, XtPointer)

static void
UpdateTextColorOKCB (Widget, XtPointer, XtPointer)

static void
UpdateTextColorDefaultCB (Widget, XtPointer, XtPointer)

static void
DefaultTextColorCB (Widget, XtPointer, XtPointer)

static void
DefaultTextColorOKCB (Widget, XtPointer, XtPointer)

static void
DefaultTextColorDefaultCB (Widget, XtPointer, XtPointer)

static void
UpdateFillColorCB (Widget, XtPointer, XtPointer)

static void

UpdateFillColorOKCB (Widget, XtPointer, XtPointer)

static void
UpdateFillColorDefaultCB (Widget, XtPointer, XtPointer)

static void
DefaultFillColorCB (Widget, XtPointer, XtPointer)

static void
DefaultFillColorOKCB (Widget, XtPointer, XtPointer)

static void
DefaultFillColorDefaultCB (Widget, XtPointer, XtPointer)

static void
DefaultLineStyleCB (Widget, XtPointer, XtPointer)

static void
DefaultLineStyleOKCB (Widget, XtPointer, XtPointer)

static void
DefaultLineStyleDefaultCB (Widget, XtPointer, XtPointer)

static void
UpdateLineStyleCB (Widget, XtPointer, XtPointer)

static void
UpdateLineStyleOKCB (Widget, XtPointer, XtPointer)

static void
UpdateLineStyleDefaultCB (Widget, XtPointer, XtPointer)

static void
DefaultFontCB (Widget, XtPointer, XtPointer)

static void

DefaultFontOKCB (Widget, XtPointer, XtPointer)

static void
DefaultFontDefaultCB (Widget, XtPointer, XtPointer)

static void
UpdateFontCB (Widget, XtPointer, XtPointer)

static void
UpdateFontOKCB (Widget, XtPointer, XtPointer)

static void
UpdateFontDefaultCB (Widget, XtPointer, XtPointer)

static void
DefaultTextAlignmentCB (Widget, XtPointer, XtPointer)

static void
DefaultTextAlignmentOKCB (Widget, XtPointer, XtPointer)

static void
DefaultTextAlignmentDefaultCB (Widget, XtPointer, XtPointer)

static void
UpdateTextAlignmentCB (Widget, XtPointer, XtPointer)

static void
UpdateTextAlignmentOKCB (Widget, XtPointer, XtPointer)

static void
UpdateTextAlignmentDefaultCB (Widget, XtPointer, XtPointer)

static void

UpdateFSFilesCB (Widget, XtPointer, XtPointer)

static void
UpdateTextCB (Widget, XtPointer, XtPointer)

static void
PointSnappingCB (Widget, XtPointer, XtPointer)
Callbacks for the grid functions.

static void
ShowGridCB (Widget, XtPointer, XtPointer)

static void
GridSizeCB (Widget, XtPointer, XtPointer)

static void
PointDistanceCB (Widget, XtPointer, XtPointer)

static void
GridSizeOKCB (Widget, XtPointer, XtPointer)

static void
PointDistanceOKCB (Widget, XtPointer, XtPointer)

static void
UndoCB (Widget, XtPointer, XtPointer)
view/edit functions.

static void
RedoCB (Widget, XtPointer, XtPointer)

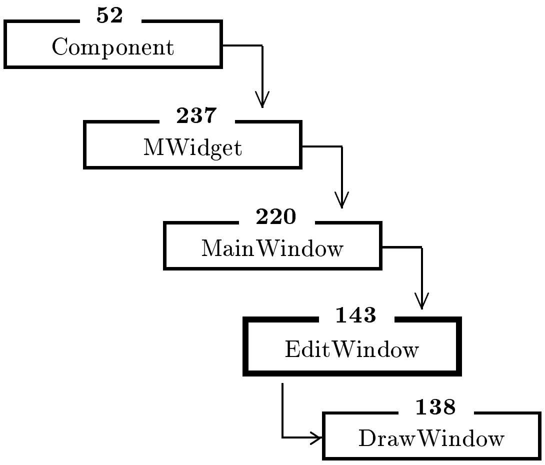
static void
RefreshCB (Widget, XtPointer, XtPointer)

143

class **EditWindow** : public MainWindow

(abstract) main window class common to all editors.

Inheritance



Public Members

	EditWindow (const char *name)	
virtual	**~EditWindow** ()	
const char*	**GetClassName** ()	
void	**Manage** ()	
enum	**ScreenType**	*most common screen sizes.*
void	**SetStatus** (const char *message)	*set text field of status area to message*
void	**SetStatus** (const string *message)	
void	**SetDocumentName** (const char *s)	*set text field of document name to s*
void	**SetDocumentName** (const string *s)	

void **SetDocumentType** (const char *s)
set text field of document type to s

void **SetDocumentType** (const string *s)

void **SetDocumentLevel** (const char *s)

void **SetDocumentLevel** (const string *s)

void **SetDirName** (const char *s)
set text field of directory name to s

void **SetDirName** (const string *s)

void **SetModified** (bool set)
set/reset modified toggle

virtual void **SetUndoName** (const char *s)
sets name of edit undo menu item.

virtual void **SetRedoName** (const char *s)
sets name of edit redo menu item.

virtual void **EnableUndo** (bool flag)
enables edit undo menu item.

virtual void **EnableRedo** (bool flag)
enables edit redo menu item.

virtual void **EnableDocumentSource** (bool)
enable show document source menu item.

bool **IsDoMap** ()

Menu* **GetFileMenu** ()

Menu* **GetEditMenu** ()

Menu* **GetPrintMenu** ()

Menu* **GetHelpMenu** ()

Menu* **GetViewMenu** ()

Menu* **GetSearchMenu** ()

Menu*	**GetPropertiesMenu** ()
MenuItem*	**GetPopupEditItems** ()
void	**SetPageSubMenuItem** (const char *text, int n) *set toggles in submenu of page menu.*
void	**SetPageMenuItem** (const char *text, bool b) *set toggle in page menu.*

Protected Members

Document*	**GetDocument** ()
void	**SetDocument** (Document *d)
Printer*	**GetPrinter** ()
Helper*	**GetHelper** ()
int	**GetArgsLeft** ()
void	**SetArgsLeft** (int n)
ScreenType	**GetScreenType** ()
bool	**IsDoLoad** ()
void	**SetDoLoad** (bool b)
void	**SetLoadFile** (const char *s)
Widget	**GetMenuBar** ()
void	**SetMenuBar** (Widget w)
Widget	**GetDocumentArea** ()
void	**SetDocumentArea** (Widget w)
Widget	**GetStatusArea** ()
void	**SetStatusArea** (Widget w)
Widget	**GetControlArea** ()
void	**SetControlArea** (Widget w)
Widget	**GetLogo** ()
void	**SetLogo** (Widget w)
Widget	**GetDirName** ()

void **SetDirName** (Widget w)

void **ParseUserOptions** (int argc, char **argv)
parse user options.

Widget **CreateWorkArea** (Widget parent, int argc, char **argv)
create the work area of a document window widget (a frame).

virtual Widget
CreateMenuBar (Widget parent)

virtual Widget
CreateLogo (Widget parent)

virtual Widget
CreateDocumentArea (Widget parent)

virtual Widget
CreateStatusArea (Widget parent)

virtual Widget
CreateControlArea (Widget parent)

Widget **CreateToggle** (Widget parent, const char *label, bool Init, XtCallbackProc fun, XtPointer clientData)

void **Finalize** ()

virtual void
InitMenuItems ()

Menu* **fileMenu**

Menu* **editMenu**

Menu* **printMenu**

Menu* **pageMenu**

Menu* **viewMenu**

Menu* **propertiesMenu**

Menu* **helpMenu**

Menu* **searchMenu**

MenuItem*
popupEditItems

MenuItem*

fileItems

MenuItem*
editItems

MenuItem*
viewItems

MenuItem*
bannerPageItems

MenuItem*
printItems

MenuItem*
searchItems

MenuItem*
propertiesItems

MenuItem*
printerOptionsItems

MenuItem*
helpItems

MenuItem*
pageOrientationItems

MenuItem*
pageSizeItems

MenuItem*
pageItems

virtual void
WriteUsage (const char *s)

Private Members

Printer*	**printer**	*The printing subsystem.*
Document*	**document**	*The document that is edited.*
Helper*	**helper**	*The helping subsystem.*
Widget	**menubar**	*Motif menu bar widget.*
Widget	**logo**	*Motif Label widget containing logo pixmap.*
Widget	**documentArea**	*Motif RowColumn Widget containg the following:*

Widget	**documentType**	*Motif TextField for document type.*
Widget	**documentName**	*Motif TextField for document name.*
Widget	**documentLevel**	
Widget	**modifiedToggle**	*Motif ToggleButton for modification.*
Widget	**controlArea**	*Motif RowColumn widget containing the following:*
Widget	**statusArea**	*Motif RowColumn Widget containing the following:*
Widget	**statusText**	*Motif TextField for status.*
Widget	**dirName**	*Motif TextField for directory name.*
ScreenType	**screenType**	*the type of X screen.*
string	**loadFile**	
string	**projDir**	
bool	**doLoad**	
int	**argsLeft**	
virtual void	**CreateDocument** ()	
void	**CreatePrinter** ()	
void	**CreateHelper** ()	
void	**DetermineScreen** (Screen *)	*Determine ScreenType from actual X screen width and height.*
static void	**AboutCB** (Widget, XtPointer, XtPointer)	
bool	**generatePS**	*for -toPS, -toEPS or -toFig options.*
bool	**generateEPS**	
bool	**generatePNG**	
bool	**generateFig**	
bool	**generateLatexFonts**	
string	**psFile**	

string	**pngFile**	
string	**epsFile**	
string	**figFile**	
Pixmap	**logoPixmap**	
Bitmap	**logoBitmap**	
bool	**doMap**	*Do map (show) mainwindow when it is realized*

144

template <class T> class **Element**

element of arbitrary type T.

Public Members

Element <T> *
next

Element <T> *
prev

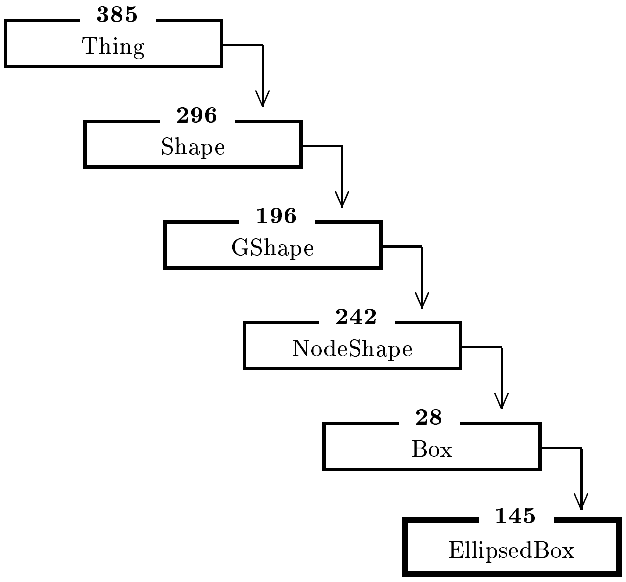
T **item**

145

class **EllipsedBox** : public Box

box class drawn as a ellipsed rectangle.

Inheritance



Public Members

	EllipsedBox (ShapeView *v, Grafport *g, int x, int y, int w = BOX_WIDTH, int h = BOX_HEIGHT)
	EllipsedBox (const NodeShape &c)
Shape*	**Clone** ()
int	**GetClassType** () const
void	**DrawOutLine** (const Point *center)
void	**DrawDuplicate** ()
Point	**GiveSnp** (Point to) const
Point	**GiveOrthoSnp** (Point to) const

Point **GiveSnp** (Point p, int order, int count, int &code)

Protected Members

void **SetSize** (int w, int h)
Additional check on size

void **DrawShape** () *draws a ellipsed rectangle.*

int **GetRadius** () const
return radius of rounded corner.

Private Members

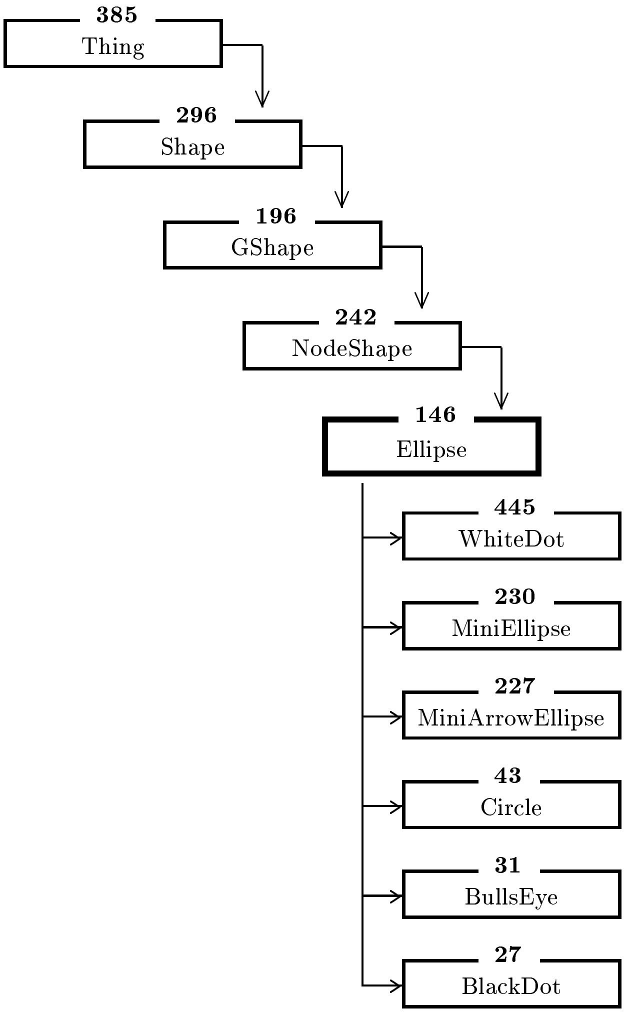
static const int
RD_FACTOR *radius factor of the rounded corner.*

146

class **Ellipse** : public NodeShape

ellipse shape class.

Inheritance



Public Members

static const int

ELLIPSE_WIDTH
default width.

static const int
ELLIPSE_HEIGHT
default height.

Ellipse (ShapeView *v, Grafport *g, int x, int y,
int w = ELLIPSE_WIDTH,
int h = ELLIPSE_HEIGHT)

Ellipse (const NodeShape &c)

Shape* **Clone** ()

void **DrawOutLine** (const Point *center)

void **DrawDuplicate** ()
the asterisk has a slightly different position.

Point **GiveSnp** (Point to) const

Point **GiveSnp** (Point p, int count, int order, int &code)

void **AdjustSize** () *size adjustment is a bit different.*

int **GetClassType** () const

static const int
ELLIPSE_HANDLES
default nr. of handles.

Protected Members

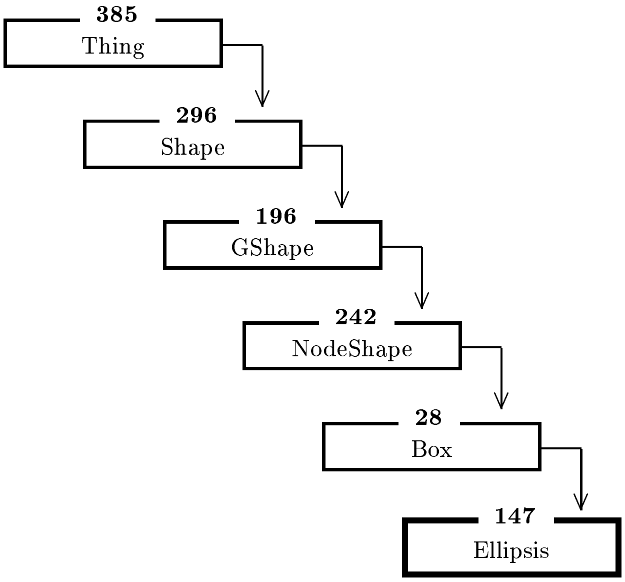
void **DrawShape** ()

147

class **Ellipsis** : public Box

invisible box class, having three black dots.

Inheritance



Public Members

static const int
DOT_WIDTH

static const int
DOT_HEIGHT

static const int
ELLIPSIS_WIDTH

static const int
ELLIPSIS_HEIGHT

Ellipsis (ShapeView *v, Grafport *g, int x, int y,
int w = ELLIPSIS_WIDTH,
int h = ELLIPSIS_HEIGHT)

Ellipsis (const NodeShape &c)

Shape*	**Clone** ()	
int	**GetClassType** () const	

Protected Members

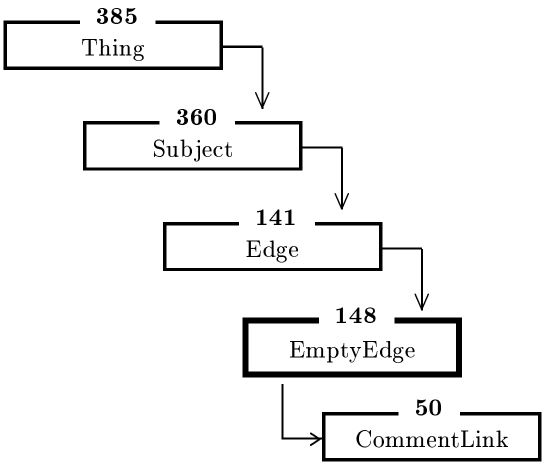
void	**DrawShape** ()	*draws no box only the three black dots.*

148

class **EmptyEdge** : public Edge

empty edge class, having fixed label ""

Inheritance



Public Members

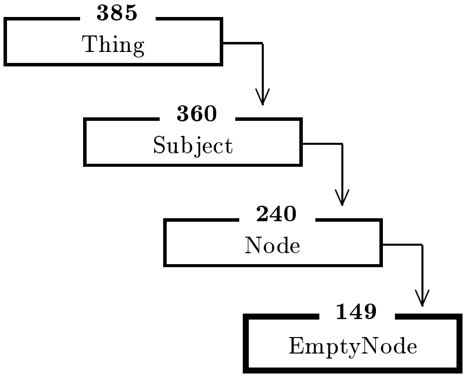
	EmptyEdge (Graph *g, Subject *n1, Subject *n2)
Subject*	**Clone** ()
Subject::NameErrType	**SetName** (const string *s)
int	**GetClassType** () const

149

class **EmptyNode** : public Node

a node with empty name and empty index.

Inheritance



Public Members

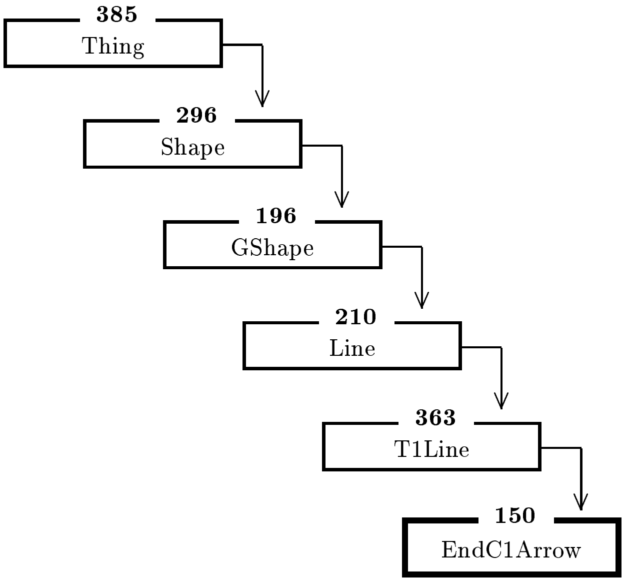
	EmptyNode (Graph *g)
int	**GetClassType** () const
Subject*	**Clone** ()
Subject::NameErrType	**SetName** (const string *s)
Subject::NameErrType	**SetIndex** (const string *s)

150

class **EndC1Arrow** : public T1Line

arrow having card constraint at 'to' side, name is not editable (SND).

Inheritance



Public Members

	EndC1Arrow (ShapeView *v, Grafport *g, GShape *n1, GShape *n2, List<Point *> *line, bool Curved)
	EndC1Arrow (ShapeView *v, Grafport *g, GShape *n1, GShape *n2, bool Curved)
Shape*	**Clone** ()
int	**GetClassType** () const
bool	**SetAssocSubject** (AssocList *)
void	**SetTextShape** ()

Private Members

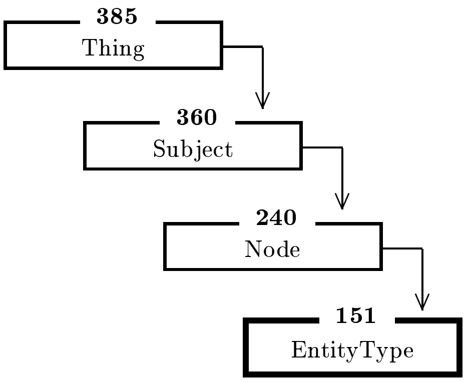
void **Initialize** ()

151

class **EntityType** : public Node

entity type (ER) node class.

Inheritance



Public Members

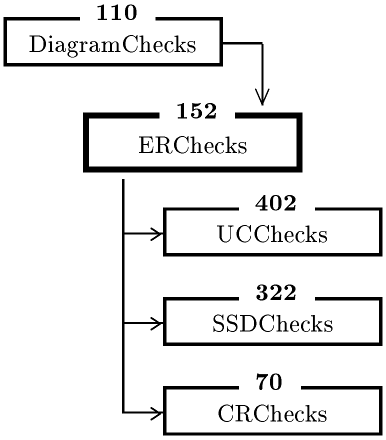
	EntityType (ERGraph *)
Subject*	**Clone** ()
Subject::NameErrType	**SetName** (const string *name)
int	**GetClassType** () const

152

class **ERChecks** : public DiagramChecks

implements checks for soft constraints on ER diagrams.

Inheritance



Public Members

ERChecks (Diagram *d, Graph *g)

virtual unsigned **CheckRelationshipCoherence** (string &chkbuf)
relationship nodes are correctly connected.

unsigned **CheckDoubleRelationships** (string &chkbuf)
name + component types are unique for relationship nodes.

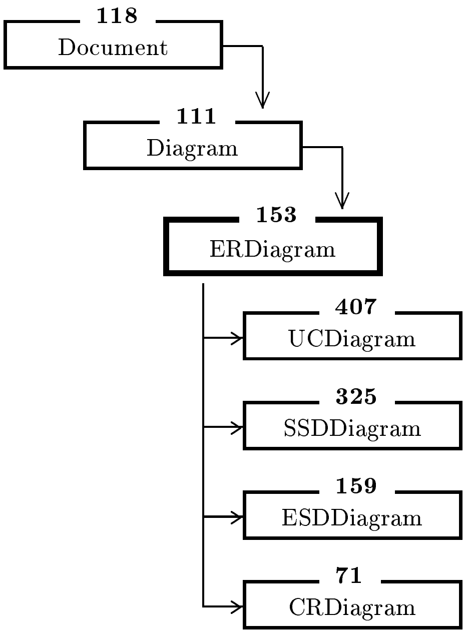
unsigned **CheckNamelessBinaryRelationships** (string &chkbuf)
relationship edges have an non-empty name or role name.

153

class **ERDiagram** : public Diagram

entity-relationship diagram class

Inheritance



Public Members

	ERDiagram (Config *, ERWindow *, ERViewer *, ERGraph *)
virtual	**~ERDiagram** ()
Node*	**CreateNode** ()
Edge*	**CreateEdge** (Subject *n1, Subject *n2)
NodeShape*	**CreateNodeShape** (Node *node, int x, int y)
Line*	**CreateLine** (Edge *edge, GShape *fromShape, GShape *toShape, List<Point *> *line)

void	**UpdateNodeType** (int n)
void	**UpdateEdgeType** (int n)
bool	**SetText** (TextShape *t, const string *s)
void	**CheckDocument** ()

Protected Members

Thing*	**CreateThing** (int classNr)
bool	**CheckIsaLoop** (Edge *edge)
bool	**CheckEdgeConstraints** (Subject *n1, Subject *n2)

Private Members

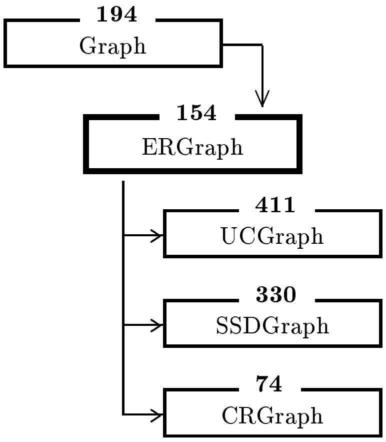
ERChecks*
erChecks

154

class **ERGraph** : public Graph

entity-relationship graph class

Inheritance



Public Members

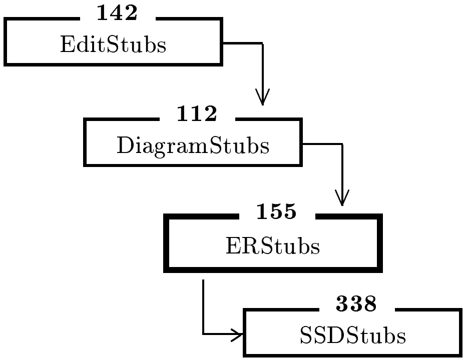
	ERGraph ()	
bool	**IsaLeaves** (Subject *s)	*returns if an is_a edge leaves this subject.*
bool	**IsaPathExists** (Subject *n1, Subject *n2)	*returns if there is an is_a path from n1 to n2.*
bool	**IsaPathExists** (Subject *n1, Subject *n2, List<Edge *> *path)	
Subject*	**GetGeneralization** (Subject *s)	*return the subject that is a generalization of this subject*
void	**InitConnections** ()	

155

class **ERStubs** : public DiagramStubs

Specific stubs for a ERD.

Inheritance



Public Members

static void
HideValueTypesCB (Widget w, XtPointer, XtPointer)

static void
ShowISACB (Widget w, XtPointer, XtPointer)

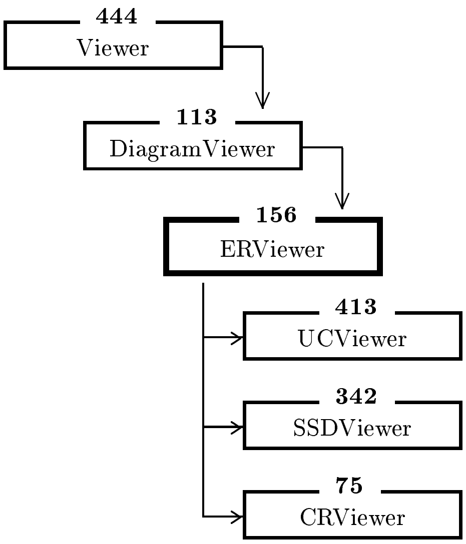
static void
UpdateReadDirectionCB (Widget, XtPointer, XtPointer)

156

class **ERViewer** : public DiagramViewer

entity-relationship viewer class.

Inheritance



Public Members

	ERViewer (Config *c, ERWindow *w)	
void	**Initialize** ()	
bool	**IsIsaOnly** ()	
bool	**IsValueTypesHidden** ()	
void	**SetHideValueTypes** (bool set)	*set/reset hided value types.*
void	**SetShowISA** (bool set)	*set/reset show only ISA.*
void	**UpdateReadDirection** (ReadDirection::Type dir)	*update read direction arrow selected object types.*

Private Members

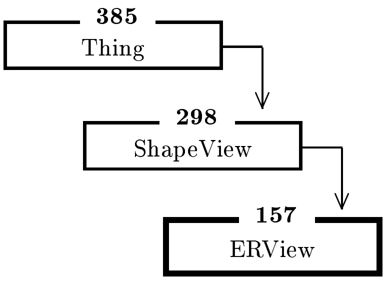
bool	**valueTypesHidden**	*value types hidden.*
bool	**isaOnly**	*isa hierarchy only is made visible.*

157

class **ERView** : public ShapeView

entity-relationship viewer class.

Inheritance



Public Members

	ERView (ERViewer *v)
void	**PutValueTypesTemp** ()
void	**GetValueTypesTemp** ()
void	**GetNotIsaTemp** ()
void	**PutNotIsaTemp** ()

Private Members

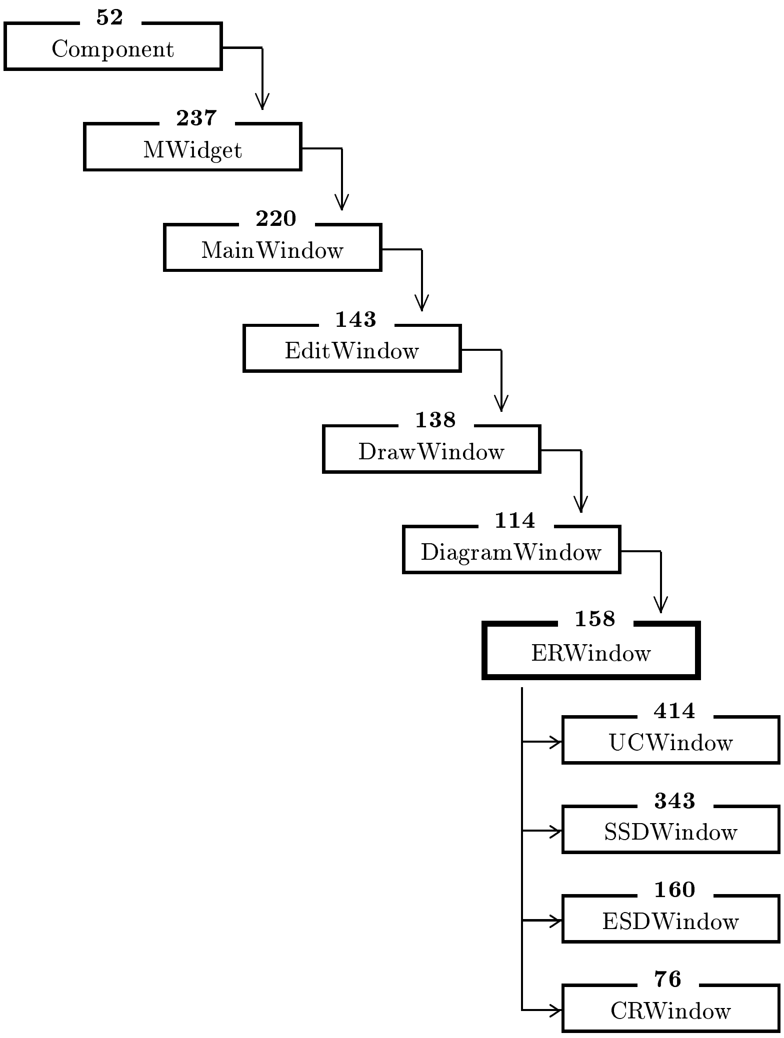
void	**DrawWholeLine** (Shape *shape)

158

class **ERWindow** : public DiagramWindow

entity-relationship edit window class.

Inheritance



Public Members

ERWindow (const char *name)

~ERWindow ()

Protected Members

void **InitMenuItems** ()

MenuItem*
ERReadDirectionItems

Private Members

int **GetTool** ()

DiagramViewer*
CreateViewer (Config *config)

Graph* **CreateGraph** ()

Diagram* **CreateDiagram** (Config *c, DiagramViewer *v, Graph *g)

void **DetermineIcons** ()

static const char*
ER_NODE_NAMES []

static const Bitmap
ER_NODE_ICONS []

static const char*
ER_EDGE_NAMES []

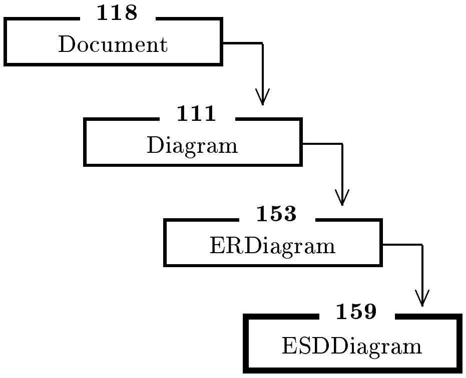
static const Bitmap
ER_EDGE_ICONS []

159

class **ESDDiagram** : public ERDiagram

UML static structure diagram class.

Inheritance



Public Members

	ESDDiagram (Config *, ESDWindow *, SSDViewer *, SSDGraph *)
virtual	**~ESDDiagram** ()
Node*	**CreateNode** ()
Edge*	**CreateEdge** (Subject *n1, Subject *n2)
NodeShape*	**CreateNodeShape** (Node *node, int x, int y)
Line*	**CreateLine** (Edge *edge, GShape *fromShape, GShape *toShape, List<Point *> *line)
void	**UpdateNodeType** (int n)
void	**UpdateEdgeType** (int n)
bool	**SetText** (TextShape *t, const string *s)
bool	**SetAttribute** (StringListNode *n, const string *s, unsigned nr)

Set Attribute in class obj at position nr.

bool **SetRoleName** (Subject *subj, const string *s, unsigned nr)
Set nr-th role name of subject. Check name syntax.

void **CheckDocument** ()
check the soft constraints on a ESD.

Protected Members

void **MakeErrorMessage** (StringListNode *node, StringListNode::StringErrorType result, const string *s, string *text)
Give error after failing to update attribute or operation.

Thing* **CreateThing** (int classNr)

SSDChecks* **GetSSDChecks** ()

static const int **MINI_ELLIPSE_WIDTH**

static const int **MINI_ELLIPSE_HEIGHT**

Private Members

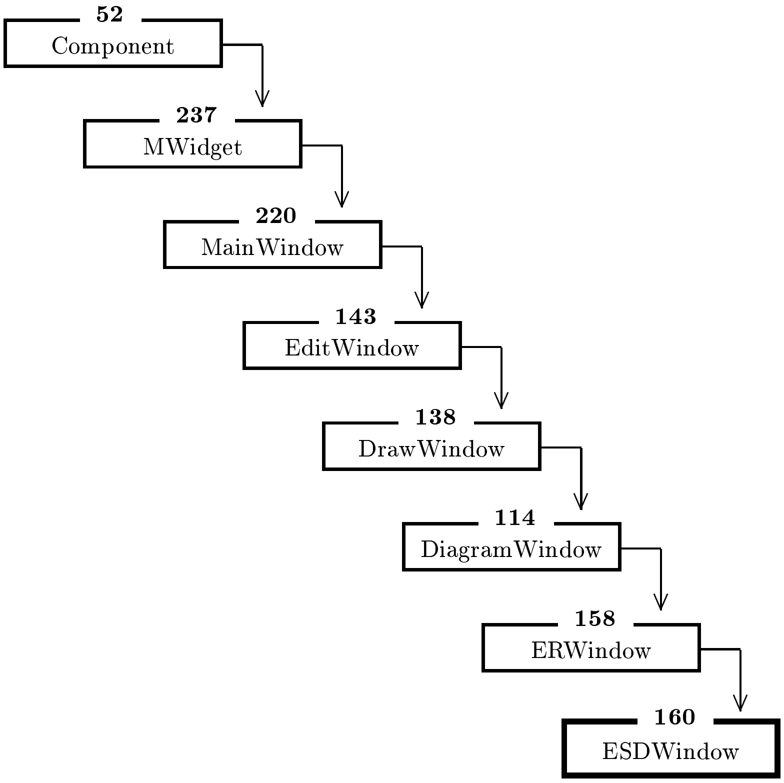
SSDChecks* **ssdChecks**

160

class **ESDWindow** : public ERWindow

class-relationship edit window class.

Inheritance



Public Members

	ESDWindow (const char *name)
virtual	**~ESDWindow** ()

Private Members

int	**GetTool** ()

DiagramViewer*
CreateViewer (Config *config)

Graph* **CreateGraph** ()

Diagram* **CreateDiagram** (Config *c, DiagramViewer *v, Graph *g)

void **DetermineIcons** ()

MenuItem*
diagramESDBoxItems

void **InitMenuItems** ()

static const char*
ESD_NODE_NAMES []

static const Bitmap
ESD_NODE_ICONS []

static const char*
ESD_EDGE_NAMES []

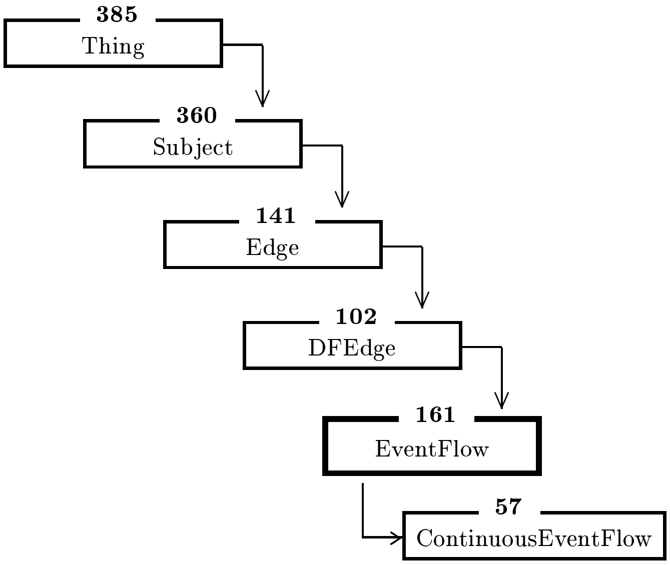
static const Bitmap
ESD_EDGE_ICONS []

161

class **EventFlow** : public DFEdge

(discrete) event flow (EFD) edge class.

Inheritance



Public Members

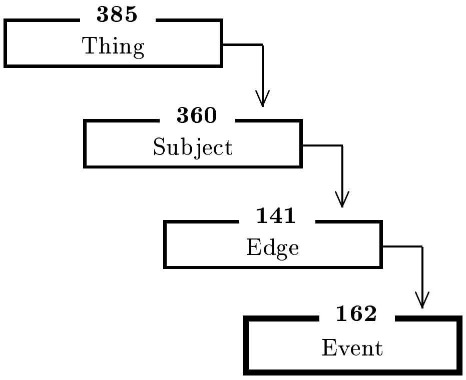
	EventFlow (DCFGraph *g, Subject *n1, Subject *n2)
int	**GetClassType** () const
Subject*	**Clone** ()
Subject::NameErrType	**SetName** (const string *name)

162

class **Event** : public Edge

event (process graph) edge class.

Inheritance



Public Members

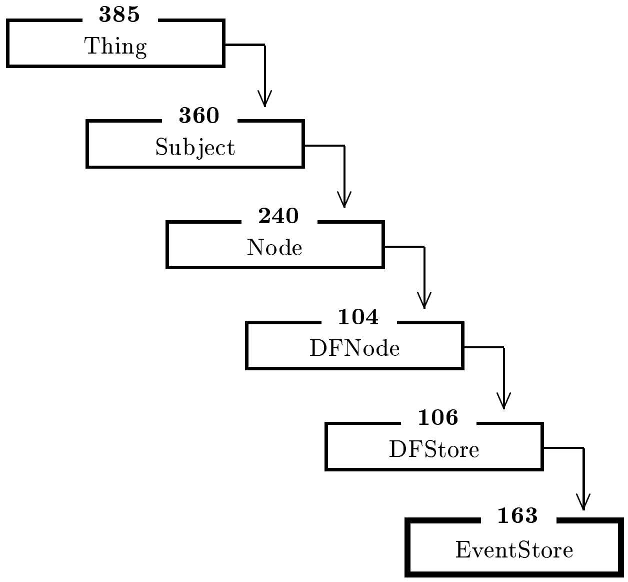
	Event (RPGraph *g, Subject *n1, Subject *n2)
int	**GetClassType** () const
Subject*	**Clone** ()
Subject::NameErrType	**SetName** (const string *name)

163

class **EventStore** : public DFStore

event store (EFD) node class.

Inheritance



Public Members

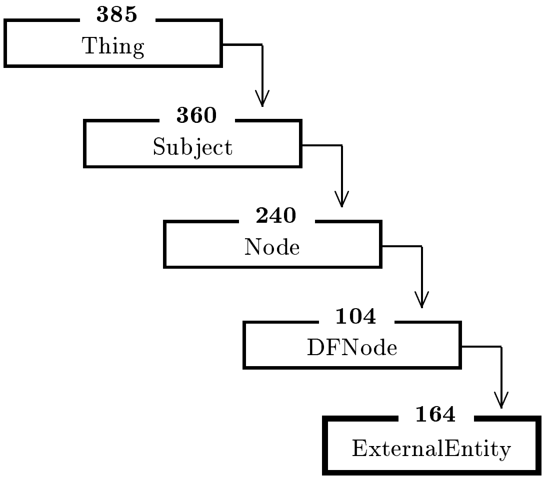
	EventStore (DCFGraph *g)
int	**GetClassType** () const
Subject*	**Clone** ()

164

class **ExternalEntity** : public DFNode

external entity (DF) node class.

Inheritance



Public Members

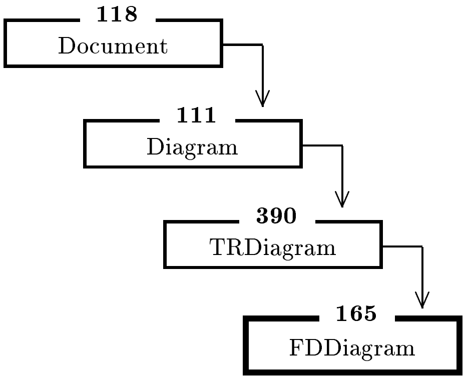
	ExternalEntity (DFGraph *g)
int	**GetClassType** () const
Subject*	**Clone** ()

165

class **FDDiagram** : public TRDiagram

function refinement tree class.

Inheritance



Public Members

FDDiagram (Config *, FDWindow *, FDViewer *, FDGraph *)

Node* **CreateNode** ()

Edge* **CreateEdge** (Subject *n1, Subject *n2)

NodeShape* **CreateNodeShape** (Node *node, int x, int y)

Line* **CreateLine** (Edge *edge, GShape *fromShape, GShape *toShape, List<Point *> *line)

void **UpdateNodeType** (int n)

void **UpdateEdgeType** (int n)

Protected Members

bool **CheckEdgeConstraints** (Subject *n1, Subject *n2)

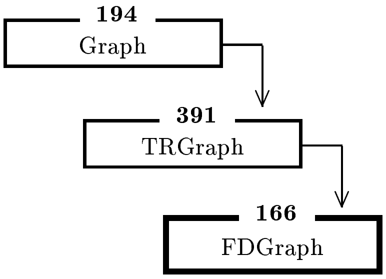
void	**CheckDocument** ()
Thing*	**CreateThing** (int classNr)

166

class **FDGraph** : public TRGraph

function refinement tree graph class.

Inheritance

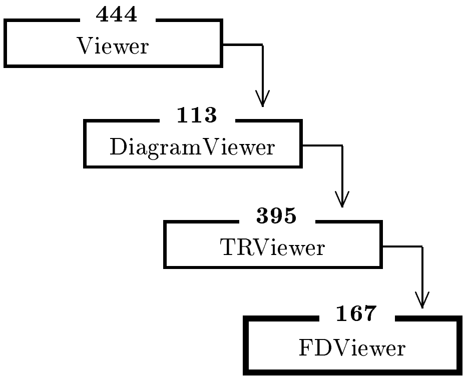


167

class **FDViewer** : public TRViewer

function refinement tree viewer class.

Inheritance



Public Members

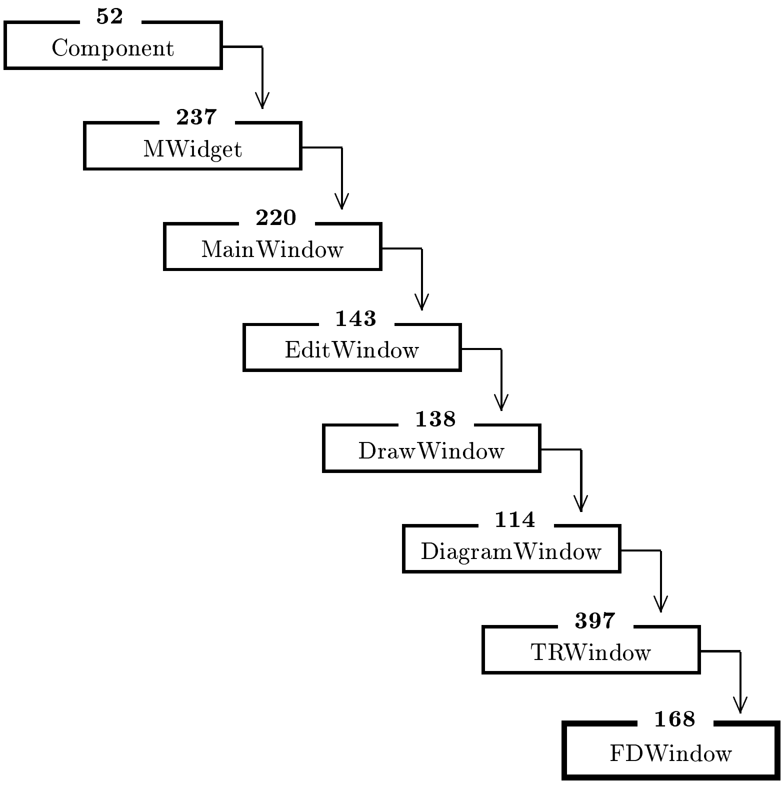
FDViewer (Config *c, FDWindow *w)

168

class **FDWindow** : public TRWindow

function refinement tree edit window class.

Inheritance



Public Members

FDWindow (const char *name)

Private Members

int **GetTool** ()

DiagramViewer*

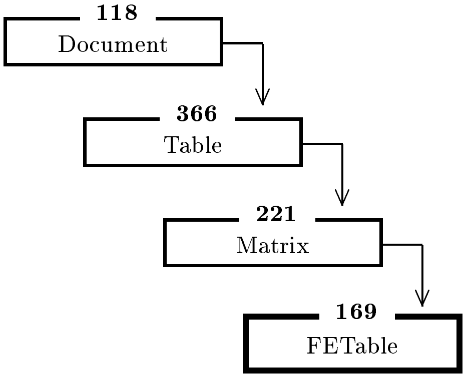
	CreateViewer (Config *c)
Graph*	**CreateGraph** ()
Diagram*	**CreateDiagram** (Config *c, DiagramViewer *v, Graph *g)
void	**DetermineIcons** ()
static const char*	**FD_NODE_NAMES** []
static const Bitmap	**FD_NODE_ICONS** []
static const char*	**FD_EDGE_NAMES** []
static const Bitmap	**FD_EDGE_ICONS** []

169

class **FETable** : public Matrix

function entity type table class.

Inheritance



Public Members

	FETable (Config *c, FEWindow *t, FEViewer *v)
void	**CheckDocument** ()

Private Members

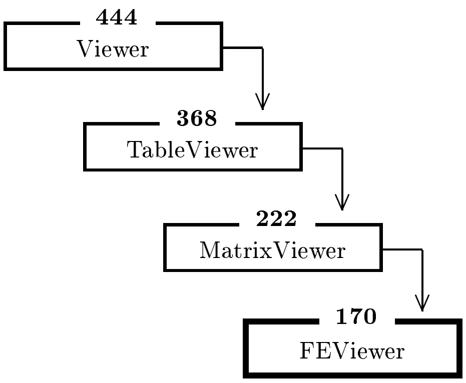
bool	**CheckTextString** (Cell *c, const string *text)
bool	**IsCRUD** (const string *text)
unsigned	**CountCEntries** (unsigned row)

170

class **FEViewer** : public MatrixViewer

function entity type table viewer class.

Inheritance



Public Members

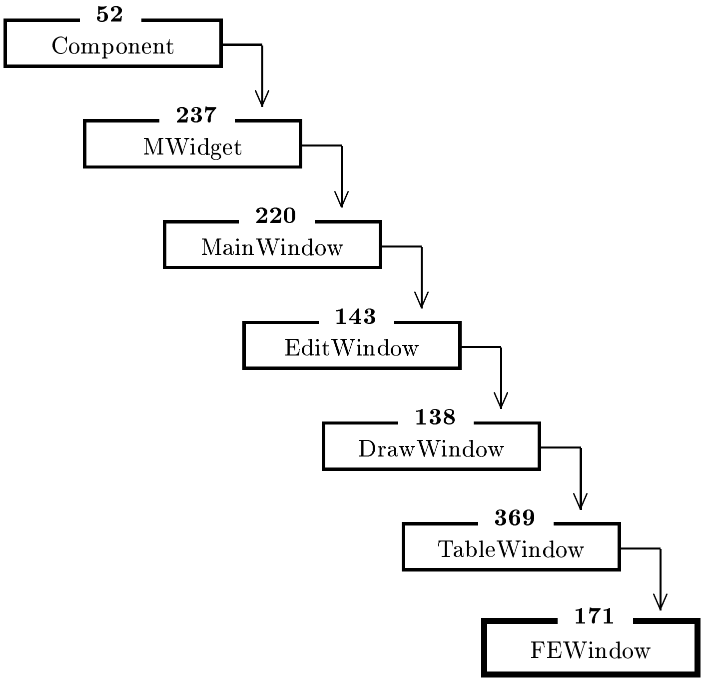
	FEViewer (Config *c, TableWindow *t)
void	**Initialize** ()

171

class **FEWindow** : public TableWindow

function entity type table window class.

Inheritance



Public Members

	FEWindow (const char *name)
TableViewer*	**CreateViewer** ()
Table*	**CreateTable** ()

Private Members

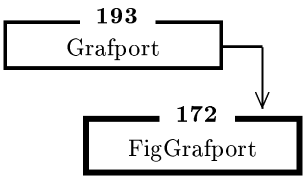
int	**GetTool** ()

172

class **FigGrafport** : public Grafport

Fig Grafport Class

Inheritance



Public Members

	FigGrafport (const char *fileName) *open grafport to filename, When filename = "", stdout is used.*
virtual	**~FigGrafport** ()
bool	**IsSucces** ()
void	**ClearArea** (int, int, int, int)
void	**SetLineWidth** (unsigned n)
void	**SetBackgroundColor** (const string *)
void	**SetBackgroundColor** (const char *)
void	**SetForegroundColor** (const string *)
void	**SetForegroundColor** (const char *)
void	**DrawRectangle** (int x, int y, int wd, int ht)
void	**FillRectangle** (int x, int y, int wd, int ht)
void	**DrawRoundedRectangle** (int x, int y, int wd, int ht, int rd)
void	**FillRoundedRectangle** (int x, int y, int wd, int ht, int rd)
void	**DrawEllipsedRectangle** (int x, int y, int wd, int ht, int rd)

void **FillEllipsedRectangle** (int x, int y, int wd, int ht, int rd)

void **DrawDisk** (int x, int y, int wd, int ht, int rd)

void **FillDisk** (int x, int y, int wd, int ht, int rd)

void **DrawUnzoomedRectangle** (int x, int y, int wd, int ht)

void **DrawStringLeft** (int x, int y, const char *str)

void **DrawStringCentered** (int x, int y, const char *str)

void **DrawStringRight** (int x, int y, const char *str)

void **DrawUnzoomedStringLeft** (int x, int y, const char *str)

void **DrawUnzoomedStringCentered** (int x, int y, const char *str)

void **DrawStringLeftUnderlined** (int x, int y, const char *str)

void **DrawStringCenteredUnderlined** (int x, int y, const char *str)

void **DrawStringRightUnderlined** (int x, int y, const char *str)

void **DrawLine** (int x1, int y1, int x2, int y2)

void **DrawLine** (const Point *p1, const Point *p2)

void **DrawCurve** (const Point *p1, const Point *p2, const Point *p3, const Point *p4)

void **DrawUnzoomedLine** (int x1, int y1, int x2, int y2)

void **DrawPoint** (int x, int y)

void **DrawPoint** (const Point *p)

void **DrawEllipse** (int x, int y, int wd, int ht)

void **FillEllipse** (int x, int y, int wd, int ht)

void	**DrawArc** (int x, int y, int wd, int ht, int arc1, int arc2)
void	**FillArc** (int x, int y, int wd, int ht, int arc1, int arc2)
void	**DrawPolygon** (const Point *points, int n)
void	**FillPolygon** (const Point *points, int n)
void	**Header** (bool landscape=False) *generate Fig header.*
void	**SetFont** (XFont *font) *set grafport font.*
void	**SetPageSize** (double x, double y)
void	**SetFactor** (double f)
void	**SetTexFonts** (bool b)
bool	**IsTexFonts** ()

Protected Members

static void	**MakeFigString** (string *str)

Private Members

FILE*	**fd**	*output file.*
static const double	**F**	*Factor to convert X coordinates to Fig coordinates.*
void	**InitColors** ()	*Load built-in Fig colors.*
int	**penColor**	*Xfig pen color number, set by SetForeground.*
int	**fillColor**	*Xfig fill color number, set by FilIXXXX.*
double	**width**	
double	**height**	
double	**factor**	
bool	**succes**	

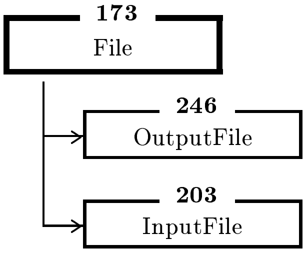
int	**psFont**
int	**texFont**
bool	**texFonts**
void	**DrawFigPolyLine** (List<Point *> *points, int sub_type, int radius = -1)
void	**DrawFigRectangle** (int x, int y, int wd, int ht, int sub_type, int radius = -1)
void	**DrawFigEllipse** (int x, int y, int rd_x, int rd_y, int sub_type)
void	**DrawFigText** (int x, int y, const char *text, int sub_type)
void	**SetFillColor** ()
void	**SetPSFont** ()
void	**SetTexFont** ()
void	**GetFigStringSize** (const char *str, double &ht, double &wd)

173

class **File**

Abstract file class.

Inheritance



Public Members

	File ()	
virtual	**~File** ()	
virtual void	**Open** (const string* file)	*open a file.*
virtual void	**Close** ()	*close current the file.*
virtual bool	**Good** ()	*is it a correct file.*
const string*	**GetFileName** ()	*return file name.*

Protected Members

bool	**IsOpened** ()	

Private Members

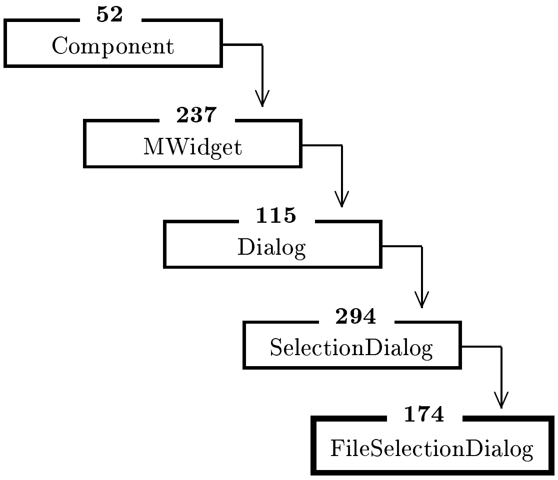
bool	**opened**	*The file is currently open.*
string	**fileName**	*The file that is opened.*

174

class **FileSelectionDialog** : public SelectionDialog

file selection dialog class.

Inheritance



Public Members

	FileSelectionDialog (Widget parent)
const char*	**GetClassName** ()
void	**SetExtension** (const char *extension)
void	**SetExtension** (const string *extension)
void	**SetDirectory** (const char *d)
void	**SetDirectory** (const string *d)
void	**SetDefaultFile** (const char *f)
void	**SetDefaultFile** (const string *f)
void	**CreateOptionMenu** (string *label, List<string *> *items)
void	**GetOption** (string *item)

void **Popup** ()

void **GetFileName** (char *name)
returns selected file name.

void **GetFileName** (string *name)

void **GetDirectory** (char *dir)
returns current directory

void **GetDirectory** (string *name)

const string*
GetDefaultFile () const

void **ManageOptionMenu** (bool b)

void **SetOptionChangedCallback** (XtCallbackProc fun, XtPointer cd)

void **SetOption** (const string *s)

Protected Members

void **CreateWidget** ()

void **Configure** ()

static void
OptionChangedCB (Widget, XtPointer, XtPointer)

static void
SetDefaultFileCB (Widget, XtPointer, XtPointer)

Private Members

Widget **pane**

Widget **menuFrame**

Widget **menu**

string **defaultFile**

175

class **FileUtils**

utilities with files.

Public Members

static bool
Load (Widget w, const char *fileName, string *text)

static bool
Save (Widget w, const char *fileName, const string *text)

static bool
PrintingOK (Widget w, const char *jobName)

static bool
Print (Widget w, const char *fileName, const char *jobName, const char *filter)

176

class **FillStyle**

different fill styles

Public Members

enum **Type**

static Type
String2Type (const string* s)
e.g. convert "Unfilled" to UNFILLED.

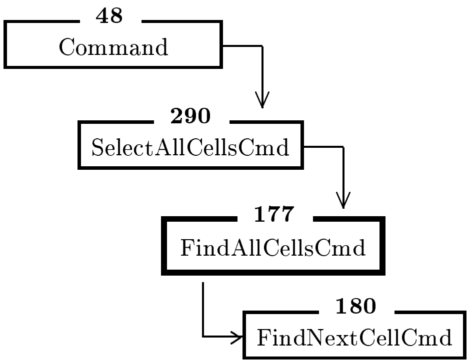
static void
Type2String (Type t, string* s)
e.g. convert UNFILLED to "Unfilled".

177

class **FindAllCellsCmd** : public SelectAllCellsCmd

select all cells containing a certain substring.

Inheritance



Public Members

	FindAllCellsCmd (Table *t, TableViewer *v, const string *s, bool c1, bool c2)
void	**Execute** ()
const char*	**GetName** () const

Protected Members

Cell*	**GetNextCell** ()

Private Members

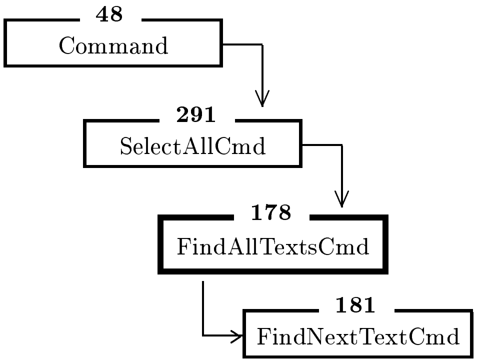
string	**findString**
bool	**caseSensitive**
bool	**substring**

178

class **FindAllTextsCmd** : public SelectAllCmd

select all shapes containing a certain substring as name.

Inheritance



Public Members

	FindAllTextsCmd (ShapeView *, const string *s, bool sens, bool substr, bool nameOnly)
void	**Execute** ()
const char*	**GetName** () const

Protected Members

GShape*	**GetNextShape** ()

Private Members

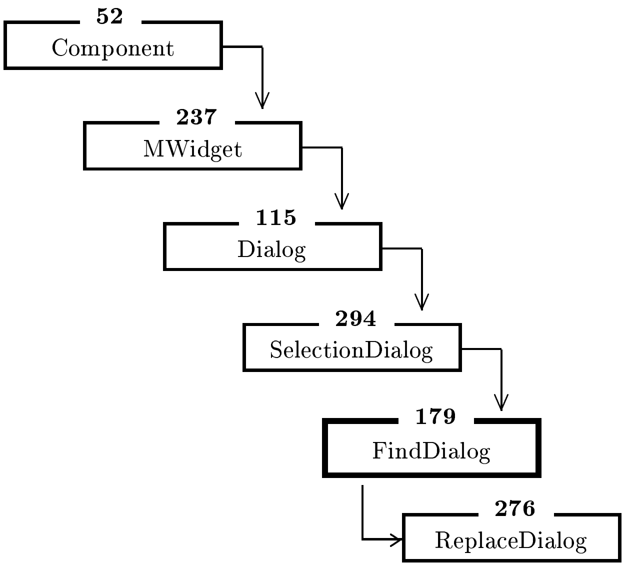
string	**findString**
bool	**caseSensitive**
bool	**substring**
bool	**nameOnly**

179

class **FindDialog** : public SelectionDialog

find dialog class.

Inheritance



Public Members

	FindDialog (Widget parent)
const char*	**GetClassName** ()
bool	**SensitiveOn** ()
bool	**SubstringOn** ()
bool	**NameOnlyOn** ()
void	**ManageSensitiveToggle** (bool b)
void	**ManageSubstringToggle** (bool b)
void	**ManageNameOnlyToggle** (bool b)

Protected Members

void **Configure** ()

void **CreateWidget** ()

Widget **GetForm** ()

Widget **GetToggleRC** ()

virtual void
AttachToggles ()

Private Members

Widget **form**

Widget **toggleRC**

Widget **substringToggle**

Widget **sensitiveToggle**

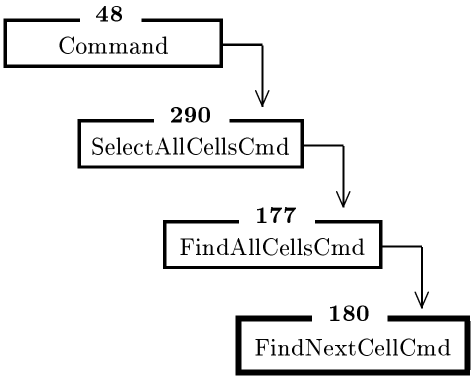
Widget **nameOnlyToggle**

180

class **FindNextCellCmd** : public FindAllCellsCmd

select next cell containing a certain substring in text.

Inheritance



Public Members

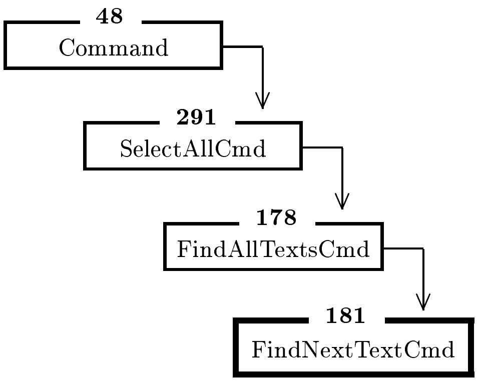
	FindNextCellCmd (Table *t, TableViewer *v, const string *s, bool c1, bool c2)
void	**Execute** ()
const char*	**GetName** () const

181

class **FindNextTextCmd** : public FindAllTextsCmd

select next shape containing a certain substring as name.

Inheritance



Public Members

FindNextTextCmd (ShapeView *v, const string *s, bool sens, bool substr, bool nameOnly)

void **Execute** ()

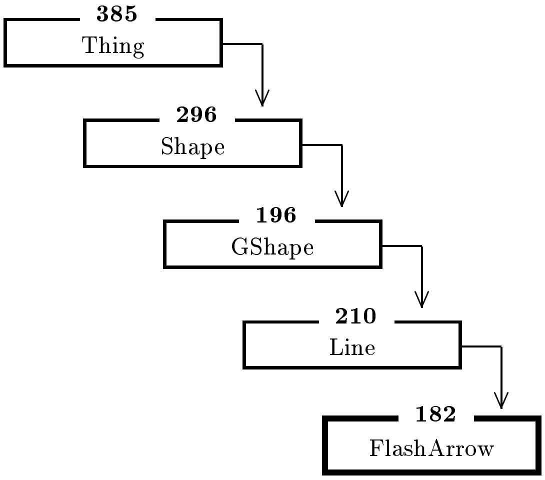
const char* **GetName** () const

182

class **FlashArrow** : public Line

An arrow with some kind of lightning flash.

Inheritance



Public Members

FlashArrow (ShapeView *s, Grafport *g, GShape *n1, GShape *n2, List<Point *> *line, bool Curved)

FlashArrow (ShapeView *v, Grafport *g, GShape *n1, GShape *n2, bool Curved)

Shape* **Clone** ()

int **GetClassType** () const

Protected Members

void **DrawLine** (const Point *ep1, const Point *ep2)

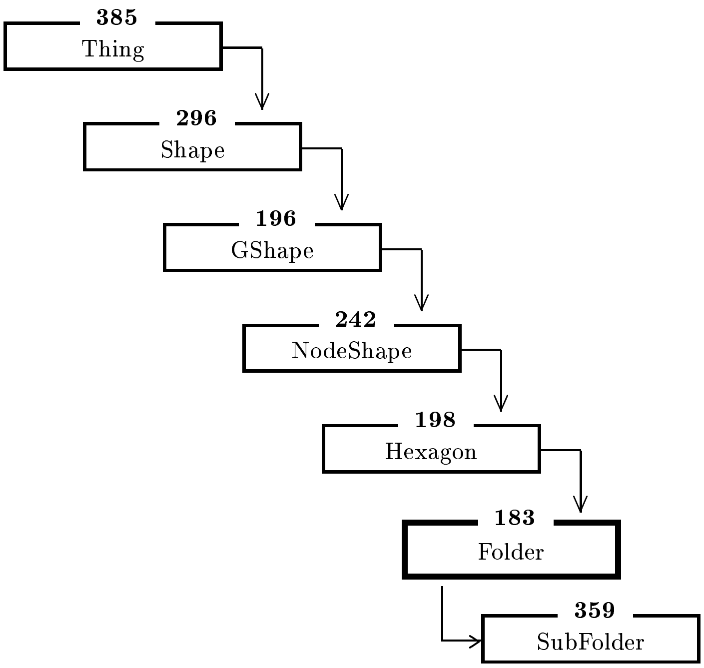
void **DrawBezier** (const Point *ep1, const Point *ep2) void **DrawLinePart** ()

183

class **Folder** : public Hexagon

Folder shape

Inheritance



Public Members

	Folder (ShapeView *v, Grafport *g, int x, int y)
	Folder (const NodeShape &)
Shape*	**Clone** ()

Protected Members

void	**DrawOutLine** (const Point* p)
void	**DrawShape** ()

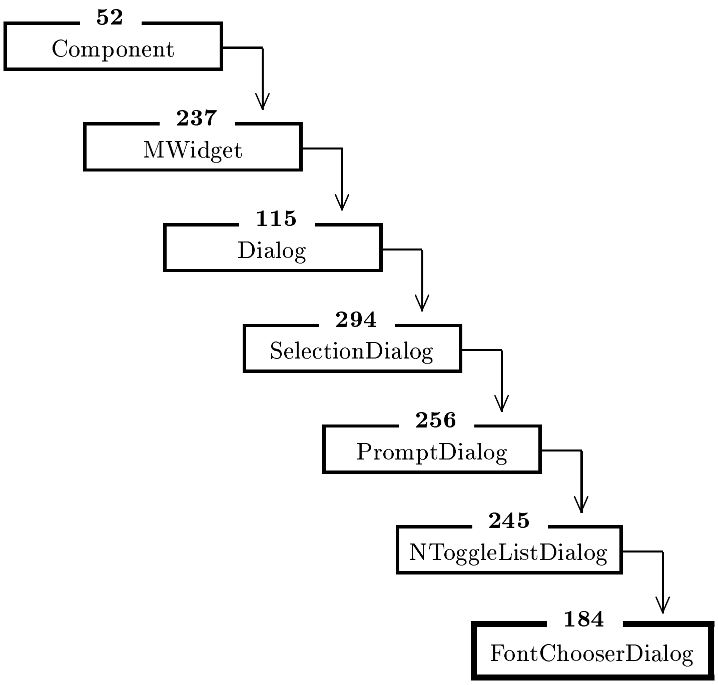
void	**SetPosition** (const Point *position, bool snap=True)
void	**SetSize** (int w, int h)
void	**CalcPoints** (int x, int y)
Point	**GiveSnp** (Point to) const
Point	**GiveSnp** (Point to, int order, int count, int& code)
int	**GetClassType** () const
void	**CalcPositionName** ()
void	**SetTextShape** ()

184

class **FontChooserDialog** : public NToggleListDialog

font chooser dialog class.

Inheritance



Public Members

	FontChooserDialog (Widget parent)
virtual	**~FontChooserDialog** ()
const char*	**GetClassName** ()
void	**ShowExample** ()
int	**GetFamily** ()
int	**GetStyle** ()
int	**GetSize** ()

bool **GetUnderlined** ()

void **SetFamily** (int)

void **SetStyle** (int)

void **SetSize** (int)

void **SetUnderlined** (bool)

Private Members

void **ChooseDefault** ()

void **Configure** ()

XFont* **GetFont** (int family, int style, int size)

XFont* **exampleFont**

List <XFont *> *
fontList *save fonts for later reuse.*

static void
ShowExampleCB (Widget, XtPointer, XtPointer)

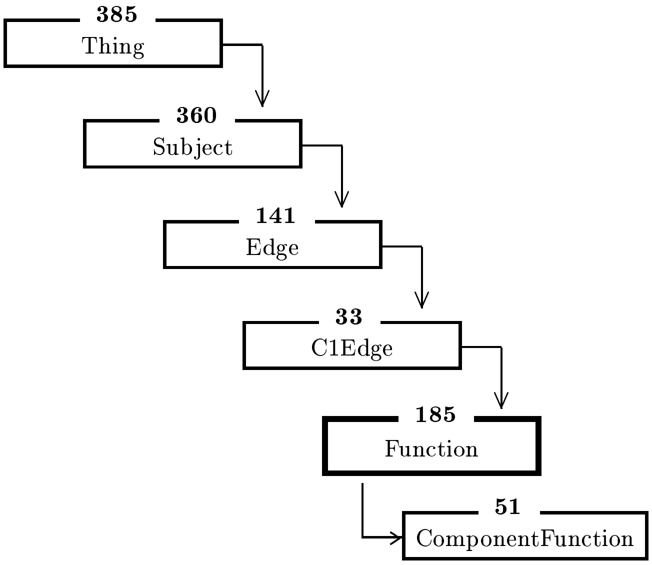
Widget **underlinedToggle**

185

class **Function** : public C1Edge

function (ER) edge class.

Inheritance



Public Members

	Function (ERGraph *g, Subject *n1, Subject *n2)
int	**GetClassType** () const
Subject*	**Clone** ()
bool	**SetConstraint** (const string *s)
Subject::NameErrType	**SetName** (const string *s)

Protected Members

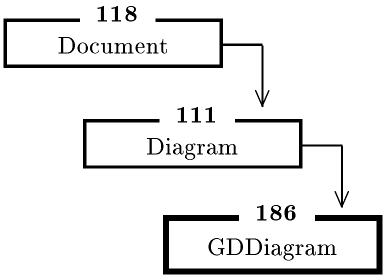
bool	**ParseCardConstraint** (const string *s)

186

class **GDDiagram** : public Diagram

generic diagram class.

Inheritance



Public Members

	GDDiagram (Config *, GDWindow *, GDViewer *, GDGraph *)
Node*	**CreateNode** ()
Edge*	**CreateEdge** (Subject *n1, Subject *n2)
NodeShape*	**CreateNodeShape** (Node *node, int x, int y)
Line*	**CreateLine** (Edge *edge, GShape *fromShape, GShape *toShape, List<Point *> *line)
void	**UpdateNodeType** (int n)
void	**UpdateEdgeType** (int n)
static const int	**DOT_WIDTH**

Protected Members

Thing*	**CreateThing** (int classNr)
bool	**HasIndexNode** (int code)

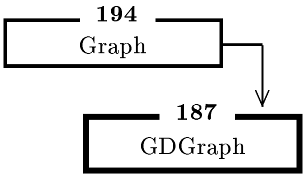
bool **HasIndexShape** (int code)

187

class **GDGraph** : public Graph

generic diagram graph class.

Inheritance



Public Members

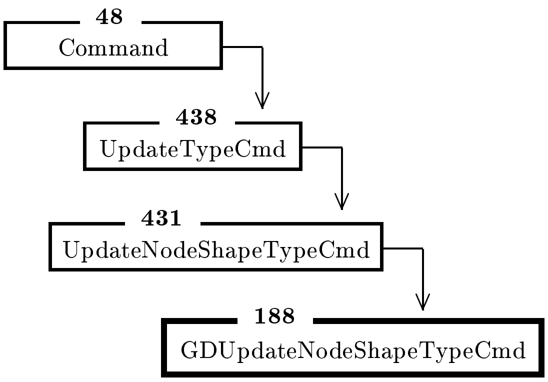
	GDGraph ()
void	**InitConnections** ()

188

class **GDUpdateNodeShapeTypeCmd** : public UpdateNodeShapeTypeCmd

command for changing node shape types in a TGD

Inheritance



Public Members

GDUpdateNodeShapeTypeCmd (ShapeView *v, List<int *> *fromTypes, int toType)

virtual Thing*
Transform (Thing *thing)

Private Members

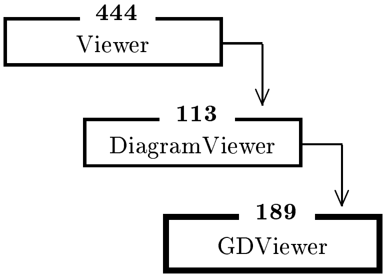
static bool
SolidShape (int type)

189

class **GDViewer** : public DiagramViewer

generic diagram viewer class.

Inheritance



Public Members

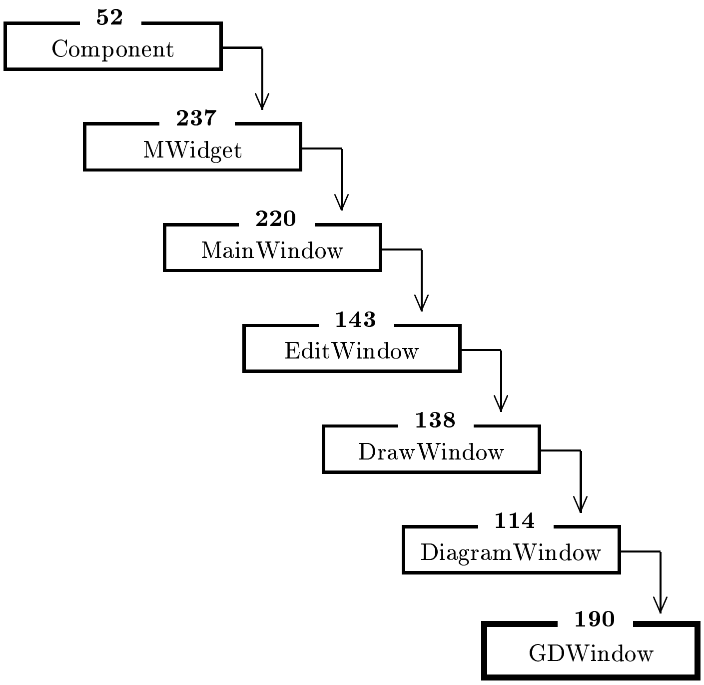
	GDViewer (Config *c, GDWindow *w)	
void	**UpdateNodeShapeType** (int)	*change node shape type of selected types.*

190

class **GDWindow** : public DiagramWindow

generic diagram edit window class.

Inheritance



Public Members

	GDWindow (const char *name)
bool	**ShowIndexToggle** ()

Private Members

void	**InitMenuItems** ()
int	**GetTool** ()
DiagramViewer*	

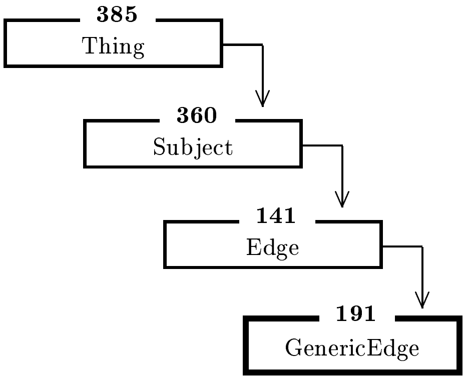
	CreateViewer (Config *c)
Graph*	**CreateGraph** ()
Diagram*	**CreateDiagram** (Config *c, DiagramViewer *v, Graph *g)
void	**DetermineIcons** ()
static const char*	**GD_NODE_NAMES** []
static const Bitmap	**GD_NODE_ICONS** []
static const char*	**GD_EDGE_NAMES** []
static const Bitmap	**GD_EDGE_ICONS** []

191

class **GenericEdge** : public Edge

generic edge class.

Inheritance



Public Members

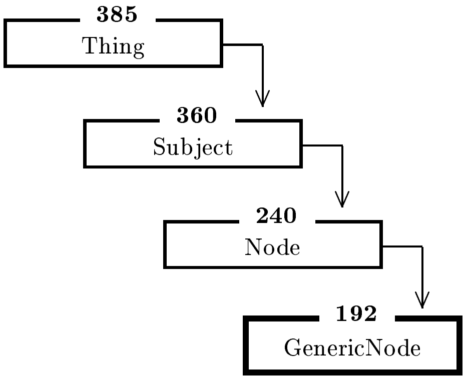
	GenericEdge (Graph *g, Subject *n1, Subject *n2)
Subject*	**Clone** ()
int	**GetClassType** () const

192

class **GenericNode** : public Node

generic node class.

Inheritance



Public Members

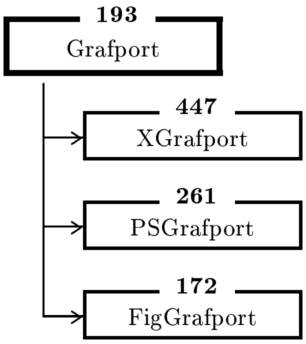
	GenericNode (Graph *g)
Subject*	**Clone** ()
int	**GetClassType** () const

193

class **Grafport**

(abstract) grafport class (primitive drawing functions).

Inheritance



Public Members

Grafport ()

virtual **~Grafport** ()

virtual bool **DrawsXor** ()

virtual void **ClearArea** (int x, int y, int wd, int ht)

virtual void **UpdateSize** (int, int)

virtual int **GetWidth** ()

virtual int **GetHeight** ()

virtual void **Redraw** (int, int, int, int, int, int)

virtual void **SetFont** (XFont *f) — *Make this the current font.*

XFont* **GetFont** () — *Get the current font.*

virtual void

SetBackgroundColor (const char *colorName)

virtual void
SetBackgroundColor (const string *colorName)

virtual void
SetForegroundColor (const char *colorName)

virtual void
SetForegroundColor (const string *colorName)

const string*
GetBackgroundColor ()

const string*
GetForegroundColor ()

List <Color *> *
GetColors ()

void **SetLineStyle** (LineStyle::Type s)

LineStyle::Type
GetLineStyle ()

void **SetFillStyle** (FillStyle::Type s)

FillStyle::Type
GetFillStyle ()

virtual void
SetLineWidth (unsigned n)

unsigned **GetLineWidth** ()

Zooming in/out.

void **SetZoomValue** (double newValue)
set zoom value.

double **GetZoomValue** ()

int **Zoom** (int x)

int **ZoomCorrect** (int x)

virtual void
DrawRectangle (int x, int y, int wd, int ht)

virtual void
FillRectangle (int x, int y, int wd, int ht)

virtual void
DrawRoundedRectangle (int x, int y, int wd, int ht, int rd)

virtual void

FillRoundedRectangle (int x, int y, int wd, int ht, int rd)

virtual void
DrawEllipsedRectangle (int x, int y, int wd, int ht, int rd)

virtual void
FillEllipsedRectangle (int x, int y, int wd, int ht, int rd)

void **DrawRectangles** (const Rectangle *rects, int n)
Draw list of n rectangles.

void **FillRectangles** (const Rectangle *rects, int n)

virtual void
DrawEditDashedRectangle (int, int, int, int)
Draw dashed rectangle (not zoomed, not printed)

virtual void
DrawUnzoomedRectangle (int, int, int, int)

virtual void
DrawStringLeft (int x, int y, const char *str)
Draw string with topleft at (x, y)

virtual void
DrawStringCentered (int x, int y, const char *str)
Draw string centered at (x, y)

virtual void
DrawStringRight (int x, int y, const char *str)
Draw string with topright at (x, y)

virtual void
DrawStringLeftUnderlined (int x, int y, const char *str)
Draw string with topleft underlined at (x, y)

virtual void

DrawStringCenteredUnderlined (int x, int y, const char *str)
Draw string centered underlined at (x, y)

virtual void
DrawStringRightUnderlined (int x, int y, const char *str)
Draw string with topright underlined at (x, y)

Draws n-line string centered (underlined?) at (x, y),

Point **DrawStringsCentered** (int x, int y, const string *str, bool ul, int cursor=0)
returns cursor position.

Point **DrawStringsLeft** (int topLeftx, int topLefty, const string *str, bool ul, int cursor=0)
draw n-line string left aligned, returns cursor position.

Point **DrawStringsRight** (int topRightx, int topRighty, const string *str, bool ul, int cursor=0)
draw n-line string right aligned, return cursor position.

unsigned **CursorPosition** (int x, int y, const string *str, int hitX, int hitY)
return cursor place closest to hitX, hitY.

virtual void
DrawUnzoomedStringLeft (int, int, const char *)

virtual void
DrawUnzoomedStringCentered (int, int, const char *)

virtual void
DrawLine (const Point *p1, const Point *p2)

virtual void

DrawLine (int x1, int y1, int x2, int y2)

virtual void
DrawCurve (const Point *p1, const Point *p2, const Point *p3, const Point *p4)

virtual void
DrawEditLine (int, int, int, int)
not zoomed not printed.

virtual void
DrawEditLine (const Point *, const Point *)
not zoomed not printed.

virtual void
DrawEditDottedGridLine (int, int, int, int)
not zoomed or printed. Uses special dot pattern.

virtual void
DrawEditDottedLine (int, int, int, int)

virtual void
DrawEditPoint (int, int)
not zoomed or printed.

virtual void
DrawEditPoint (const Point *)
not zoomed or printed.

virtual void
DrawUnzoomedLine (int, int, int, int)

virtual void
DrawPoint (int x, int y)

virtual void
DrawPoint (const Point *p)

virtual void
DrawEllipse (int x, int y, int wd, int ht)

virtual void
FillEllipse (int x, int y, int wd, int ht)

virtual void
DrawArc (int x, int y, int wd, int ht, int arc1, int arc2)

virtual void

FillArc (int x, int y, int wd, int ht, int arc1, int arc2)

virtual void
DrawDisk (int x, int y, int wd, int ht, int ellipse_ht)
disk is an ellipse, two vert. lines and a half ellipse

virtual void
FillDisk (int x, int y, int wd, int ht, int ellipse_ht)

virtual void
DrawPolygon (const Point *plist, int n)
draw lines between the first n points in plist.

virtual void
FillPolygon (const Point *plist, int n)

virtual void
DrawActiveCaret (int, int)
Draw active cursor sign.

virtual void
DrawDeactiveCaret (int, int)
Draw deactive cursor sign.

Protected Members

Point **CursorPosition** (int x, int y, const string *buf, int cursor)

Color* **LookupColor** (const string *colorName)

Color* **LookupColor** (const char *colorName)

void **AddColor** (const string *colorName, unsigned long pixel)

void **AddColor** (const char *colorName, unsigned long pixel)

void **AddColor** (Color *c)

virtual void

	InitColors ()	
void	**RecalcPolygon** (const Point *points, int nrPoints, Point *newPoints, int d)	

Private Members

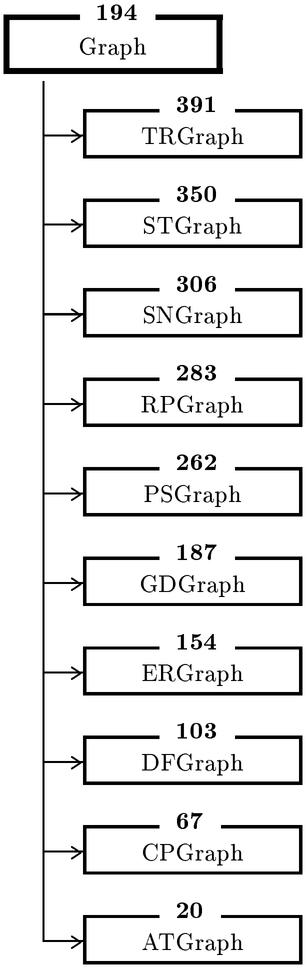
XFont*	**font**	*Current font.*
XFont*	**defaultFont**	*Default font.*
string	**backgroundColor**	*Current background color.*
string	**foregroundColor**	*Current foreground color.*
unsigned	**lineWidth**	*current line width.*
LineStyle::Type	**lineStyle**	*current line style.*
FillStyle::Type	**fillStyle**	*current fill style.*
double	**zoomValue**	*current zoom value.*
string	**buf**	
List <Color *> *	**colors**	*a cache for the colors.*

194

class **Graph**

(abstract) graph class.

Inheritance



Public Members

Graph ()

virtual	**~Graph** ()
void	**AddNode** (Node *node)
void	**AddEdge** (Edge *edge)
void	**RemoveNode** (Node *node)
void	**RemoveEdge** (Edge *edge)
bool	**HasNode** (Node *node)
bool	**HasEdge** (Edge *edge)
void	**ClearNodes** ()
void	**ClearEdges** ()
bool	**CheckConnection** (int stype1, int stype2, int edgetype) *checks if edge type may connect subject types 1 and 2.*
bool	**IsConnected** (Subject *s1, Subject *s2) *checks if there is an edge in the graph that connects s1 and s2.*
bool	**IsConnected** (Subject *s1, Subject *s2, const string *n) *same as above, but edge should also have name n.*
bool	**IsConnected** (Subject *s1, Subject *s2, int t) *here edge should also be of type t.*
bool	**IsConnected** (Subject *s1, Subject *s2, const string *n, int t) *combination of the two members above.*
int	**CompleteSubjects** (List<Subject *> *l) *add to l all edges which connect the subjects.*
int	**CompleteSubject** (List<Subject *> *l, Subject *s) *add to l all edges of which subject s is part.*
int	**CompleteSubject** (List<Subject *> *l, Subject *s1, Subject *s2) *add to l all edges which connect subjects s1 and s2.*
int	**CompleteEdges** (List<Subject *> *l)

add to l all subjects which are connected by edges already in l.

int **GetConnected** (List<Subject *> *l, Subject *s)
add to l the subjects that are directly connected to subject s.

int **GetNodes** (List<Subject *> *l)
add to l all nodes in the graph.

int **GetNodes** (List<Subject *> *l, int t)
add to l all nodes of type t in the graph.

int **GetNodes** (List<Subject *> *l, const string *n)
add to l all nodes having name n in the graph.

int **GetNodes** (List<Subject *> *l, const string *n, int t)
combination of two above members.

int **GetEdges** (List<Subject *> *l)
add to l all edges in the graph.

int **GetEdges** (List<Subject *> *l, int t)
add to l all edges of type t in the graph.

int **GetEdges** (List<Subject *> *l, const string *n)
add to l all edges having name n in the graph.

int **GetEdges** (List<Subject *> *l, const string *n, int t)
combination of two above members.

int **GetEdgesFrom** (List<Subject *> *l, Subject *s)
add to l all edges departing from subject s in the graph.

int **GetEdgesFrom** (List<Subject *> *l, Subject *s, int t)
add to l all edges of type t departing from subject s.

int **GetEdgesFrom** (List<Subject *> *l, Subject *s, const string *n) *add to l all edges having name n departing from subject s.*

int **GetEdgesFrom** (List<Subject *> *l, Subject *s, const string *n, int t)
combination of two above members.

int **GetEdgesTo** (List<Subject *> *l, Subject *s)
add to l all edges going to subject s.

int **GetEdgesTo** (List<Subject *> *l, Subject *s, int t)
add to l all edges of type t going to subject s.

int **GetEdgesTo** (List<Subject *> *l, Subject *s, const string *n)
add to l all edges having name n going to subject s.

int **GetEdgesTo** (List<Subject *> *l, Subject *s, const string *n, int t)
combination of the two above members.

int **GetEdges** (List<Subject *> *l, Subject *from, Subject *to)
add to l all edges between subjects from and to

int **GetEdges** (List<Subject *> *l, Subject *from, Subject *to, int t)
add to l all edges of type t between subjects from and to

int **GetEdges** (List<Subject *> *l, Subject *from, Subject *to, const string *n)
combination of the two above members.

int **GetEdges** (List<Subject *> *l, Subject *from, Subject *to, const string *n, int t)
combination of two above members.

int **CountNodes** () *return number of nodes in the graph*

int **CountNodes** (int t)

return number of nodes of type t in the graph

int **CountNodes** (const string *n)
return number of nodes having name n in the graph

int **CountNodes** (const string *n, int t)
return nr. of nodes of type t, having name n in the graph

int **CountEdges** () *return number of edges in the graph*

int **CountEdges** (int t)
return number of edges of type t in the graph

int **CountEdges** (const string *n)
return number of edges having name n in the graph

int **CountEdges** (const string *n, int t)
return nr. of edges of type t, having name n in the graph

int **CountEdgesFrom** (Subject *s)
return nr. of edges departing from subject s

int **CountEdgesFrom** (Subject *s, int t)
return nr. of edges of type t departing from subject s

int **CountEdgesFrom** (Subject *s, const string *n)
return number of edges having name n departing from subject s

int **CountEdgesFrom** (Subject *s, const string *n, int t)
return number of edges of type t, having name n, departing from subject s

int **CountEdgesTo** (Subject *s)
return nr. of edges going to subject s

int **CountEdgesTo** (Subject *s, int t)
return nr. of edges of type t going to subject s

int **CountEdgesTo** (Subject *s, const string *n) *return number of edges having name n going to subject s*

int **CountEdgesTo** (Subject *s, const string *n, int t)
return number of edges of type t, having name n going to subject s

int **CountEdges** (Subject *s1, Subject *s2)
return number of edges between subjects s1 and s2

int **CountEdges** (Subject *s1, Subject *s2, int t)
return number of edges of type t between subjects s1 and s2

int **CountEdges** (Subject *s1, Subject *s2, const string *n)
return number of edges having name n between subjects s1 and s2

int **CountEdges** (Subject *s1, Subject *s2, const string *n, int t)
return number of edges of type t, having name n between subjects s1 and s2

bool **PathExists** (Subject *s1, Subject *s2)
returns if there is some connected path of graph edges from s1 to s2.

bool **PathExists** (Subject *s1, Subject *s2, int t)
returns if there is some conn. path of graph edges of type t from s1 to s2.

bool **UndirectedPathExists** (Subject *s1, Subject *s2)
return if there's path from s1 to s2. irregardless edge direction.

void **WriteSubjects** (OutputFile *f)
Writes all nodes and edges to outputfile f.

virtual void **InitConnections** ()
initializes the table containing all allowed connections.

void **SetCounter** (int n)

some counter can be used

int **GetCounter** ()

void **SetIndexPrefix** (const char *s)
some string can be used as index prefix.

void **SetIndexPrefix** (const string *s)

const string*
GetIndexPrefix () const

void **GetIndex** (string *s)
index string := prefix + counter.

virtual void
GetNextIndex (string *index)
assign to index a new unique index

virtual int **CountIndexes** (const string *index)
return number of nodes in the graph having this index.

Protected Members

int* **nodeTypes** *Allowed 'connected' types in graph (depends on type of editor).*

int* **edgeTypes** *Allowed edges type in graph (depends on type of editor).*

enum *Max. number of different connected types and edge types.*

int **connections** [MAX_TYPES][MAX_TYPES][MAX_TYPES]
matrix to store what types can be connected by what edge types.

List <Node *> *
nodes *the graph nodes.*

List <Edge *> *
edges *the graph edges.*

bool **PathExists** (Subject *s1, Subject *s2, int edgetype, bool Directed)
used by other PathExists.

int **CountIndex** (const string *index, List<Subject *> *l) *return number of nodes in l having this index*

Private Members

string	**indexPrefix**	*current index prefix*
int	**counter**	*current counter*

195

class **Grid**

class for showing the grid and snapping points to discrete point positions.

Public Members

	Grid (Config *c, DrawWindow *d, Viewer *v)	
virtual	**~Grid** ()	
void	**SetSnap** (bool set)	*set snapping to continuous or discrete.*
void	**PointDistance** ()	*make popup for asking the point distance.*
void	**GridSize** ()	*make popup for asking the grid size.*
void	**SetShowGrid** (bool set)	*set showing of grid.*
bool	**IsShowGrid** () const	
void	**Snap** (Point *)	*Snap point to point-position.*
int	**Snap** (int)	*Snap int to point-position.*
void	**DrawGrid** ()	*draw grid.*
void	**SetPointDistance** (int i)	*set point distance to i.*
void	**SetGridSize** (int i)	*set grid size to i.*
DrawWindow*	**GetMainWindow** () const	
Config*	**GetConfig** () const	
ScaleDialog*	**GetScaleDialog** () const	

Private Members

DrawWindow*

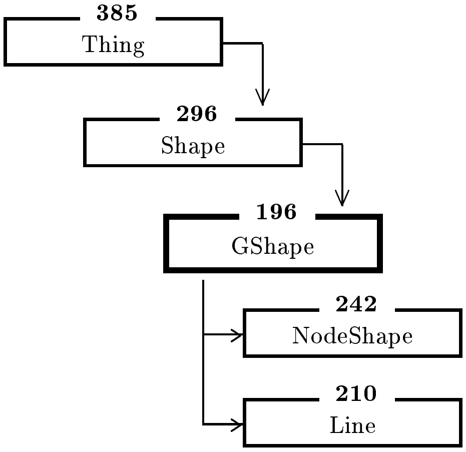
	mainwindow
Viewer*	**viewer**
Config*	**config**
ScaleDialog*	**scaleDialog**
int	**pointDistance**
bool	**snap**
bool	**showGrid**
int	**gridSize**
int	**defaultPointDistance**
int	**defaultGridSize**
static const int	**MAX_POINT_DISTANCE**
static const int	**MIN_POINT_DISTANCE**
static const int	**MAX_GRID_SIZE**
static const int	**MIN_GRID_SIZE**

196

class **GShape** : public Shape

(abstract) graphical shape class, superclass of box, ellipse, line etc.

Inheritance



Public Members

GShape (ShapeView *v, Grafport *g, int x, int y)
constructor, creates shape at position (x, y).

GShape (const GShape &)

virtual **~GShape** ()

void **Draw** ()

void **Undraw** ()

virtual void
DrawShape () *draw the shape.*

virtual void
UndrawShape ()

virtual void

DrawTextShapes ()

virtual void
UndrawTextShapes ()

virtual void
UpdateNameString (const string *s)
update string of name text shape and redraw.

void **SetNameString** (const string *s)

const string*
GetNameString ()

TextShape*
GetName () const

void **SetTextShape** ()

virtual void
SetSelect (bool s) *select this shape and its text shapes are selected.*

unsigned **GetLineWidth** () const

void **SetLineWidth** (unsigned n)

void **UpdateLineWidth** (unsigned n)

LineStyle::Type
GetLineStyle () const

void **SetLineStyle** (LineStyle::Type x)

void **UpdateLineStyle** (LineStyle::Type x)

const string*
GetTextColor () const

void **UpdateTextColor** (const string *c)

virtual TextShape*
HitTextShape (int x, int y)
Returns if (x, y) is in name, and if so returns name.

bool **HasTextShape** (TextShape *t) const *return if it has t as one of the text shapes.*

virtual int **GetLeftMost** () const
refinements of those shape members.

virtual int **GetTopMost** () const

virtual int **GetRightMost** () const

virtual int **GetBottomMost** () const

bool **IsFixedName** ()

void **SetFixedName** (bool b)

void **SetGrafport** (Grafport *g)
set grafport of itself and its text shapes.

void **SetView** (ShapeView *v)
set view of itself and its text shapes.

virtual void
UpdateFont (XFont *ft)
update font and redraw.

virtual void
UpdateAlignment (TextAlign::Type alignment)
update text alignments and redraw.

TextShape*
ChooseTextShape (int, int)
return a text shape near (x, y).

virtual void
NotifyTextMove (TextShape *)
react on movement of some text shape.

XFont* **GetFont** () const *return font of the textshapes (all textshapes have same font).*

TextAlign::Type
GetAlignment () const

bool **HasNameString** (const string *s, bool c1, bool c2) const *return if name text shape contains 's'.*

bool **HasString** (const string *s, bool c1, bool c2) const
return if some text shape contains 's'.

bool **HasString** (const string *s, bool cI, bool aS, List<TextShape *> *l)
put all textshapes containing 's' in the list l.

bool **CheckReferences** ()
Check also that name reference is ok.

Protected Members

virtual void **SetFont** (XFont *ft)
set fonts of text shapes.

virtual void **SetTextColor** (const string *c)
set color of text shapes.

void **WriteGraphicsPart** (OutputFile *ofile)
write line width and style to file.

bool **ReadGraphicsPart** (InputFile *ifile, double format)
read line width and style from file.

void **WriteTextPart** (OutputFile *ofile)
write font and alignment to file.

bool **ReadTextPart** (InputFile *ifile, double format)
read and parse font and alignment from file.

void **SetDrawAttributes** ()

virtual void **SetAlignment** (TextAlign::Type a)
set alignments of text shapes.

void **SnapPosition** () *snap at position and text shapes are moved with.*

void	**SnapTopleft** ()	*snap at topleft and text shapes are moved with.*

Private Members

TextShape* **name**		*the name text shape of the Gshape*
unsigned	**lineWidth**	*line width.*
LineStyle::Type **lineStyle**		*line style.*
bool	**filled**	*is gshape filled?*
bool	**fixedName**	*can name be edited?*

197

class **Helper**

class for showing popup help windows.

Public Members

	Helper (Config *c, MainWindow *w)
virtual	**~Helper** ()
enum	**HelpTopicType**
void	**Help** (HelpTopicType topic) *Gives help about basic operations.*
void	**SetStatus** (const char *msg)
void	**SetStatus** (const string *msg)

Private Members

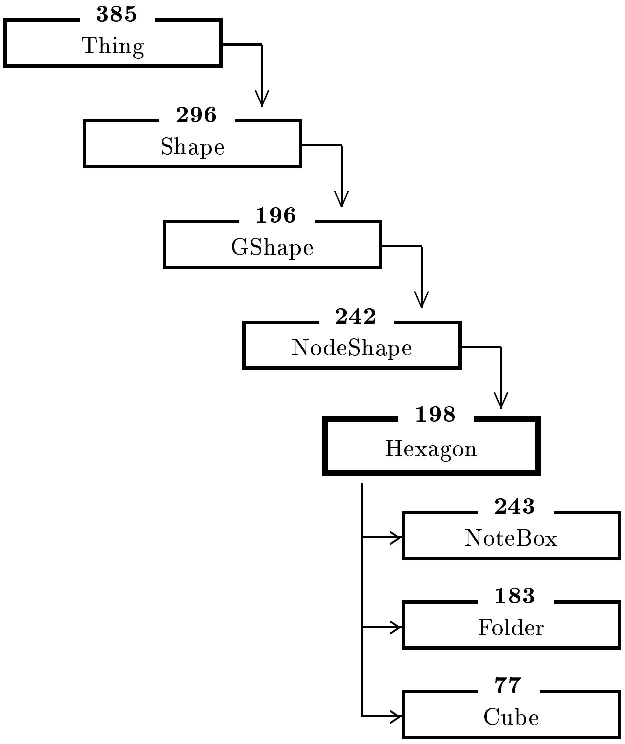
MainWindow*	**mainwindow**
TextViewDialog*	**helpDialog**
Config*	**config**
string	**text**

198

class **Hexagon** : public NodeShape

hexagon shape class.

Inheritance



Public Members

static const int
HEXAGON_WIDTH

static const int
HEXAGON_HEIGHT

Hexagon (ShapeView *v, Grafport *g, int x, int y,
int w = HEXAGON_WIDTH,
int h = HEXAGON_HEIGHT)

	Hexagon (const Hexagon &s)
	Hexagon (const NodeShape &s)
virtual	**~Hexagon** ()
Shape*	**Clone** ()
void	**DrawOutLine** (const Point *center) *draws dotted hexagon snapped to the grid.*
Point	**GiveSnp** (Point to) const
Point	**GiveSnp** (Point p, int order, int count, int &code)
bool	**ContainsPt** (int x, int y)
int	**GetClassType** () const
static const int	**HEXAGON_HANDLES**

Protected Members

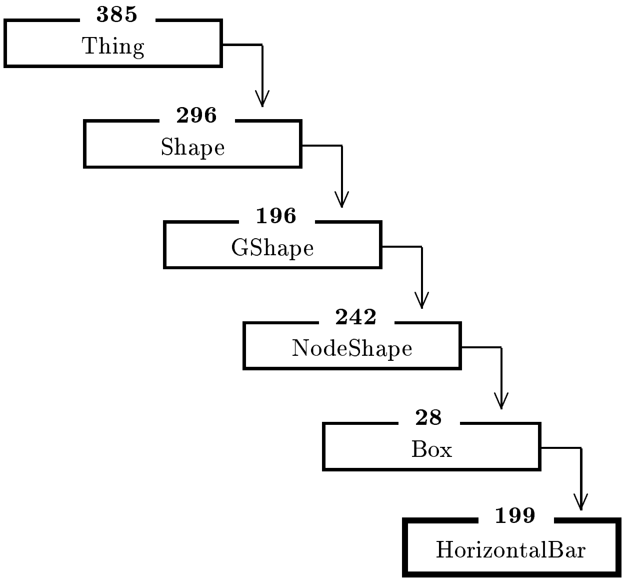
void	**DrawShape** () *draws a hexagon*
virtual void	**CalcPoints** (int x, int y)
const Point*	**GetPoints** () const
Point*	**points**

199

class **HorizontalBar** : public Box

box class with undrawn left and right sides.

Inheritance



Public Members

static const int
BAR_WIDTH

static const int
BAR_HEIGHT

HorizontalBar (ShapeView *v, Grafport *g, int x, int y, int w = BAR_WIDTH, int h = BAR_HEIGHT)

HorizontalBar (const NodeShape &c)

Shape* **Clone** ()

void **DrawOutLine** (const Point *center)

		draw bar-form dotted outline. snaps to the grid.
int	**GetClassType** () const	
static const int	**BAR_HANDLES**	

Protected Members

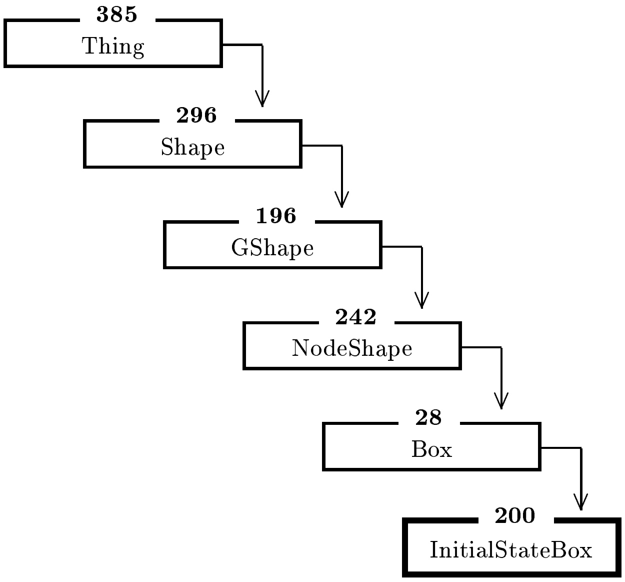
void	**DrawShape** ()	*draws a horizontal bar.*

200

class **InitialStateBox** : public Box

box having downwards pointing arrow on top and list of action shapes (STD).

Inheritance



Public Members

	InitialStateBox (ShapeView *v, Grafport *g, int x, int y, int w=BOX_WIDTH, int h = BOX_HEIGHT)
	InitialStateBox (const InitialStateBox &a)
virtual	**~InitialStateBox** ()
Shape*	**Clone** ()
int	**GetClassType** () const
void	**SetSize** (int w, int h)
void	**SetPosition** (const Point *p, bool b)

void **SetFont** (XFont *ft)

void **SetTextColor** (const string *color)

TextShape*
ChooseTextShape (int x, int y)

bool **HasTextShape** (TextShape *t) const

bool **ContainsPt** (int x, int y)

bool **HitHLine** (int x, int y)

const Point*
GetHLineFrom () const

const Point*
GetHLineTo () const

void **UpdateAnchor** (const Point *delta, const Point *position)

void **UpdateAction** (const string *s, unsigned n, bool update)
update or insert string of n-th attribute.

int **GetLeftMost** () const
take actions into account.

int **GetTopMost** () const

int **GetRightMost** () const

bool **SetAssocSubject** (AssocList *)

void **SetTextShape** ()

void **WriteMembers** (OutputFile *f)

bool **ReadMembers** (InputFile *f, double format)

void **SetGrafport** (Grafport *g)

void **SetView** (ShapeView *v)

bool **HasString** (const string *s, bool sens, bool sub) const

bool **HasString** (const string*, bool, bool, List<TextShape *> *)

Private Members

LineEnd	**end**
void	**DrawShape** ()
Point	**hlineFrom**
Point	**hlineTo**
Point	**anchorPoint**
int	**HLineLength** ()
bool	**HitVLine** (int x, int y)
void	**CalcHLinePart** ()
void	**CalcPositionHLine** ()
void	**CalcPositionActions** ()
void	**SetAction** (const string *s, unsigned n, bool update)

List <TextShape *> *
actions

static const int
ARROWLEN

static const int
HLINELEN *minimal length hline.*

static const int
HLINEXOFFSET

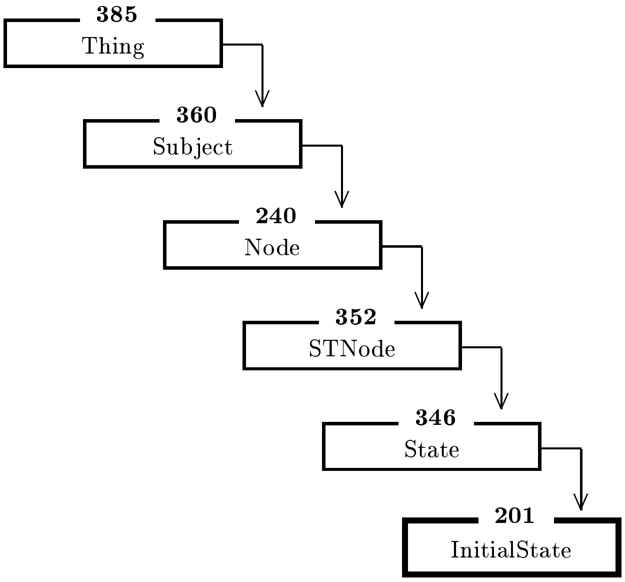
static const int
HLINEYOFFSET

201

class **InitialState** : public State

initial state (STD) node class.

Inheritance



Public Members

	InitialState (STGraph *g)
	InitialState (const InitialState &i)
virtual	**~InitialState** ()
int	**GetClassType** () const
Subject*	**Clone** ()
Subject::NameErrType	**SetAction** (const string *s, unsigned n, bool update)
const string*	**GetAction** (unsigned n)
unsigned	**NrActions** () const

bool	**HasAction** (const string *s)
void	**WriteMembers** (OutputFile *f)
bool	**ReadMembers** (InputFile *f, double format)

Private Members

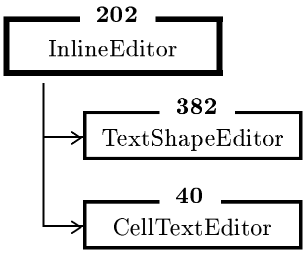
List <string *> *	**actions**
string	**controlProcess**

202

class **InlineEditor**

(abstract) text editor class.

Inheritance



Public Members

	InlineEditor (Viewer *v)	
virtual	**~InlineEditor** ()	
virtual void	**Stop** ()	*top inline editing, update text.*
virtual void	**Abort** ()	*top inline editing, do not update text.*
virtual void	**Draw** ()	
void	**Undraw** ()	
void	**AddChar** (char c)	*simple inline text edit functions.*
void	**BackSpace** ()	
void	**Delete** ()	
void	**CharLeft** ()	
void	**CharRight** ()	
void	**CharUp** ()	
void	**CharDown** ()	
void	**Home** ()	

void **End** ()

void **Clear** () *empty text*

void **SetCursor** (int x, int y)
set cursor to some position.

void **SetCursor** (int whichChar)
set cursor on n-th char.

void **StopWindow** (const string *)
stop window editing, update text.

void **CancelWindow** ()
cancel window editing.

TextEditDialog*
GetTextEditDialog () const

Protected Members

bool **IsInUse** () const

void **SetInUse** (bool b)

void **SetIgnoreWindow** (bool b)

void **SetText** (const string *t)

void **SetOldText** (const string *ot)

void **SetTextFont** (XFont *ft)

void **SetTextColor** (string *c)

void **SetAlignment** (TextAlign::Type a)

void **SetTextPosition** (Point *p)

const string*
GetText () const

const string*
GetOldText () const

Viewer* **GetViewer** () const

virtual bool
IsUnderlined () const

void **SetUnderlined** (bool b)

Private Members

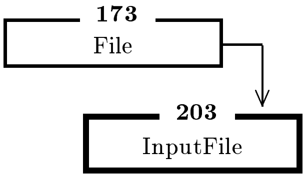
Point	**textPosition**	*the position of the text that is being edited.*
string	**text**	*multi-line text that is being edited.*
string	**oldtext**	*the text before editing.*
XFont*	**textFont**	*multi-line text that is being edited.*
string	**textColor**	
Point	**cursorPosition**	*the coordinates where the cursor is drawn.*
unsigned	**cursor**	*cursor is at for cursor-th char.*
bool	**inUse**	*editor is being used.*
bool	**ignoreWindow**	*set when inline editing is performed. window callbacks are ignored.*
TextAlign::Type	**alignment**	
Viewer*	**viewer**	
TextEditDialog*	**textEditDialog**	
virtual void	**UpdateText** ()	
bool	**underlined**	*is the text underlined*

203

class **InputFile** : public File

Input file class.

Inheritance



Public Members

	InputFile ()	
virtual	**~InputFile** ()	
void	**Open** (const string* file)	*open a file for reading.*
void	**Close** ()	*close current the file.*
bool	**Good** ()	*is it a correct file.*
bool	**Eof** ()	*is end of file reached?*
int	**LineNumber** ()	*return line number.*
int	**ReadChar** ()	*return next non-white-space char from input. Return -1 on eof.*
bool	**LookupWord** (const char* s)	
bool	**LookupString** (const char* s)	
bool	**ReadAttribute** (const string* name, string* value)	*a shorthand to lookup a name and read a value from the input file.*
bool	**ReadAttribute** (const char* name, string* value)	
bool	**ReadAttribute2** (const string* name, string* value1, string* value2)	

		another shorthand to look up a word and read two word values.
bool	**ReadAttribute2** (const char* name, string* value1, string* value2)	
bool	**ReadAttribute3** (const string* name, string* v1, string* v2, string* v3)	*another shorthand to look up a word and read three word values.*
bool	**ReadAttribute3** (const char* name, string* v1, string* v2, string* v3)	
bool	**ReadStringAttribute** (const string* name, string* value)	*another shorthand to look up a word and read a string value.*
bool	**ReadStringAttribute** (const char* name, string* value)	
bool	**ReadPair** (string* value1, string* value2)	

Protected Members

void	**SkipLine** ()	*skip rest of line.*

Private Members

ifstream*	**ifile**	*The current input file.*
int	**lineNr**	*the current ifile line count. Updated by the read functions.*

204

class **Intersect**

Functions for intersection point computation.

Public Members

static Point
BoxIntersects (Point from, Point to, int width, int height)
Return intersection from point 'from' to box at 'to' having width and height

static Point
BoxIntersects (Point position, Point to, int width, int height, int order, int count, int &code)
See above, intersecting line is order-th of count lines. code is not used.

static Point
EllipseIntersects (Point from, Point to, int width, int height)
Return intersect from point 'from' to ellipse at 'to' having wd and ht.

static Point
EllipseIntersects (Point position, Point to, int width, int height, int order, int count, int &code)
See above, intersecting line is order-th of count lines. code is not used.

static Point
DiamondIntersects (Point from, Point to, int width, int height)
Return intersect from point 'from' to diamond at 'to' having wd and ht

static Point

DiamondIntersects (Point position, Point to, int width, int height, int order, int count, int &code)
See above, intersecting line is order-th of count lines. code is not used.

static Point
TriangleIntersects (Point from, Point to, int width, int height)
Return intersect from point 'from' to triangle at 'to' having wd and ht

static Point
TriangleIntersects (Point position, Point to, int width, int height, int order, int count, int &code)
See above, intersecting line is order-th of count lines. code is not used.

static Point
NoteBoxIntersects (Point from, Point to, int width, int height)
Return intersect from point 'from' to notebox at 'to' having wd and ht

Protected Members

static Point
DiamondSnp (Point to, int dx, int dy, int x1, int y1, int x2, int y2)

static void
WhichLine (Point p, int w, int h, int dx, int dy, int &x1, int &y1, int &x2, int &y2)

static void
WhichLine2 (Point p, int w, int h, int dx, int dy, int deltaY, int &x1, int &y1, int &x2, int &y2)

static void

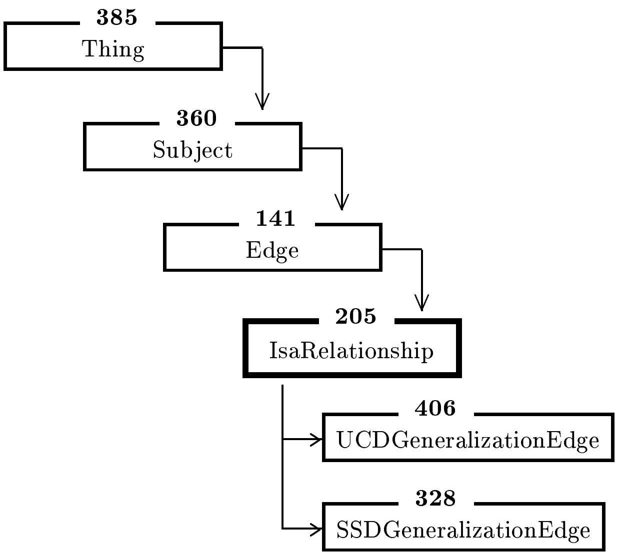
WhichLine3 (Point p, int w, int h, int dx, int dy, int deltaY, int &x1, int &y1, int &x2, int &y2)

205

class **IsaRelationship** : public Edge

isa-relationship (ER) edge class.

Inheritance



Public Members

	IsaRelationship (Graph *g, Subject *n1, Subject *n2)
int	**GetClassType** () const
Subject*	**Clone** ()
virtual Subject::NameErrType	**SetName** (const string *s) *returns WRONG_NAME.*

206

class **Key**

definitions of various character codes.

Public Members

static const int
BACKSPACE *definitions of character codes of non-printable ascii chars.*

static const int
TAB

static const int
VERTICAL_TAB

static const int
ESCAPE

static const int
FS

static const int
RS

static const int
DELETE

static const int
CTRL_B

static const int
CTRL_E

static const int
HOME *definitions of character codes of special keys (non-ascii).*

static const int
LEFT

static const int
UP

static const int
RIGHT

static const int
DOWN

static const int

PAGE_UP

static const int
PAGE_DOWN

static const int
END

static const int
INSERT

static bool
ArrowKey (int key)

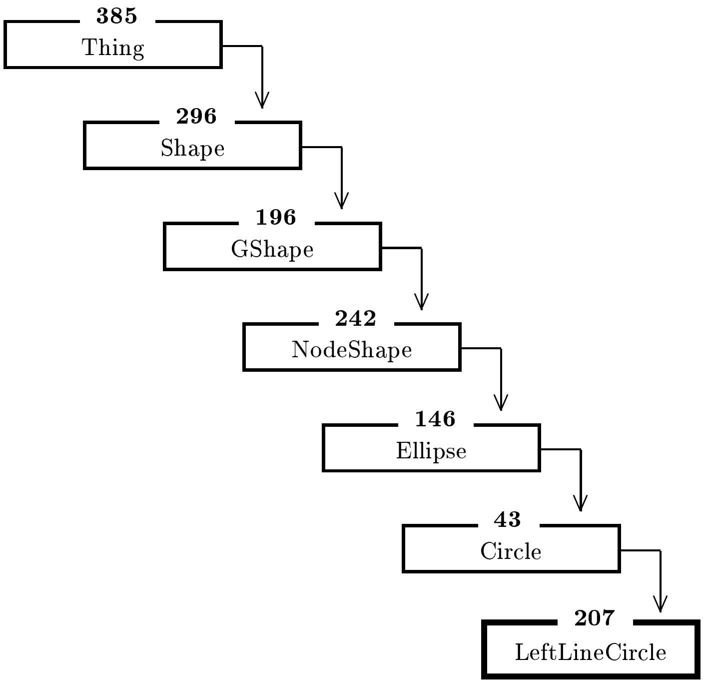
static bool
EditKey (int key)

207

class **LeftLineCircle** : public Circle

circle class having in the left part a vertical line

Inheritance



Public Members

LeftLineCircle (ShapeView *v, Grafport *g, int x, int y, int diameter = CIRCLE_WIDTH)

Shape* **Clone** ()

int **GetClassType** () const

Protected Members

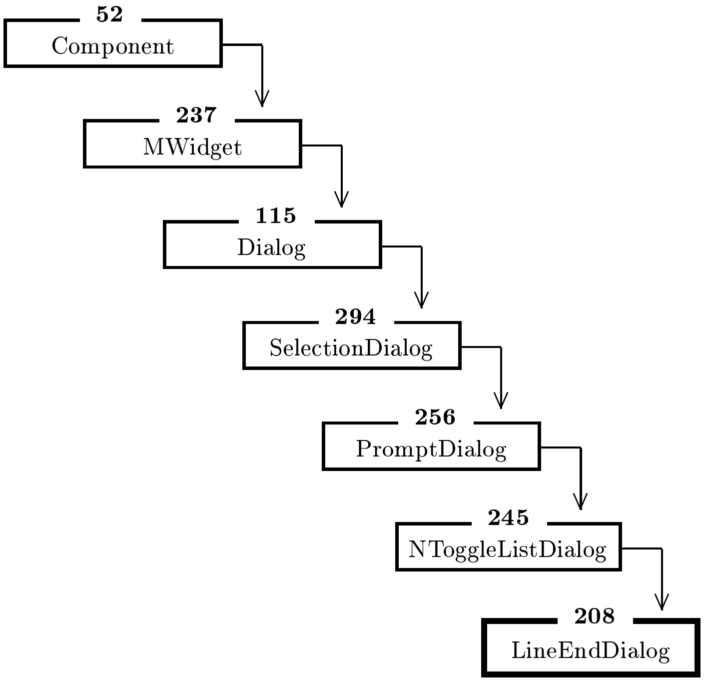
void **DrawShape** ()

208

class **LineEndDialog** : public NToggleListDialog

line end chooser dialog class.

Inheritance



Public Members

	LineEndDialog (Widget parent)
const char*	**GetClassName** ()

Private Members

void	**Configure** ()

209

class **LineEnd**

different line ending of a line

Public Members

enum **Type**
enum

LineEnd ()

LineEnd (Type t)

void **SetType** (Type t)

LineEnd::Type
GetType ()

void **SetLength** (int l)

int **GetLength** ()

void **SetWidth** (int w)

int **GetWidth** ()

static const int
CIRCLE_WIDTH
the default width of a circle line end.

static const int
OPEN_ARROW_LENGTH
default length of open arrow.

static const int
OPEN_ARROW_WIDTH
default width of open arrow.

static const int
FILLED_ARROW_LENGTH
default length of the sides of the arrow head

static const int
FILLED_ARROW_ANGLE
angle between each side and the line in degrees

static const double

FILLED_ARROW_INSET
percentage inset for base of arrow [0..1] (values +- 0.60 work well).

static const int
TRIANGLE_WIDTH
default width of a triangle

static const int
TRIANGLE_LENGTH
default length of a side of a triangle.

static const int
DIAMOND_WIDTH
default width of a diamond.

static const int
DIAMOND_LENGTH
default length of a side of a diamond.

static const TypeName
typeNames []

static Type
String2Type (const string *s)
e.g. convert "OpenArrow" to OPEN_ARROW.

static void
Type2String (Type t, string *s)
e.g. convert OPEN_ARROW to "OpenArrow".

void **Draw** (Grafport *g, const Point *from, const Point *to, Point *newEnd)
draw the given line ending in the grafport.

void static
CalcDiamond (const Point *from, const Point *to, int len, int wd, Point *points)

void static

CalcDirectionDiamond (const Point *from,
const Point *to,
const Point *apos,
int len, int wd,
Point *points)

static void
CalcFilledArrowHead (const Point *from,
const Point *to, int len,
Point *points)

Private Members

Type **type**

int **length**

int **width**

void **DrawArrowHead** (Grafport *g, const Point *from,
const Point *to, bool filled,
int count, Point *newEnd)

void **DrawOpenArrowHead** (Grafport *g,
const Point *from,
const Point *to,
Point *newEnd)

void **DrawFilledArrowHead** (Grafport *g,
const Point *from,
const Point *to,
Point *newEnd)

void **DrawExtraArrowHead** (Grafport *g,
const Point *from,
const Point *to,
bool filled,
Point *newEnd)

void **DrawCircle** (Grafport *g, const Point *from,
const Point *to, bool filled,
Point *newEnd)

void **DrawTriangle** (Grafport *g, const Point *from,
const Point *to, bool filled,
Point *newEnd)

void **DrawDiamond** (Grafport *g, const Point *from,
const Point *to, bool filled,
Point *newEnd)

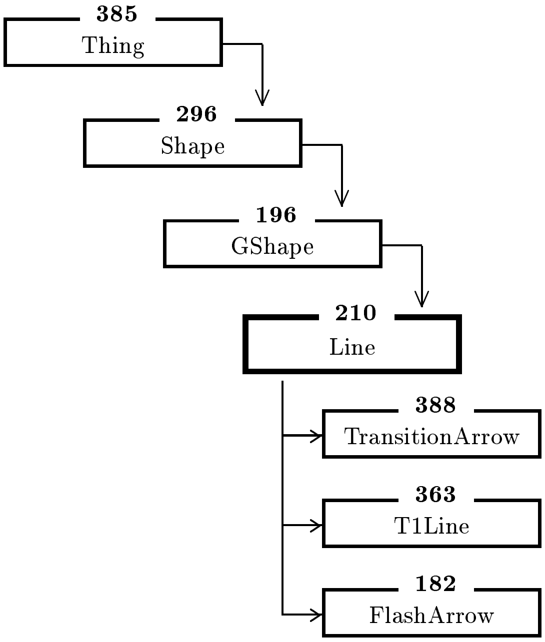
void **SetSizes** ()

210

class **Line** : public GShape

line shape class.

Inheritance



Public Members

	Line (ShapeView *v, Grafport *g, GShape *n1, GShape *n2, List<Point *> *line, bool Curved)
	Line (ShapeView *v, Grafport *g, GShape *n1, GShape *n2, bool Curved)
	Line (const Line &s)
Shape*	**Clone** ()
virtual	**~Line** ()
bool	**IsCurved** () const

void **UpdateCurve** (bool b)

void **UpdatePosition** (const Point *p, bool snap=True)
empty (line has no independent position).

void **SetPosition** (const Point *p, bool snap=True)
empty (line has no independent position).

void **DrawOutLine** (const Point *)
empty, because line has no outline.

bool **ContainsPt** (int x, int y)
returns if (x, y) is on one of the segments of line.

int **HitHandles** (int x, int y)
returns handle-nr if (x, y) hits a handle, otherwise -1.

bool **IsLine** () const *yep.*

unsigned int
GetCount () const
count and order are used to determine whether

unsigned int
GetOrder () const
more than one line connect the same pair of shapes.

void **SetOrder** (unsigned int i)

void **SetCount** (unsigned int i)

GShape* **GetFromShape** () const
fromShape and toShape are the connected shapes.

GShape* **GetToShape** () const

void **SetFromShape** (GShape *n)

void **SetToShape** (GShape *n)

unsigned int
NrPoints () const

List <Point *> *
GetLine () const *'line' is a list of points.*

bool **GetSegments** (Point pt, Point *pt1, Point *pt2) *get left and rigth line segments of point pt (if they exist).*

bool **GetSegment** (int n, Point *pt1, Point *pt2)
get line segment n (if it exists).

virtual bool
InTextArea (int x, int y)
returns if (x, y) is in text area (used for edit cursor).

void **SetEnd1** (LineEnd::Type t)

void **SetEnd2** (LineEnd::Type t)

void **UpdateEnd1** (LineEnd::Type t)

void **UpdateEnd2** (LineEnd::Type t)

LineEnd::Type
GetEnd1 ()

LineEnd::Type
GetEnd2 ()

int **GetLeftMost** () const

int **GetTopMost** () const

int **GetRightMost** () const

int **GetBottomMost** () const

void **SetToTop** () *set last point of line to top of toShape.*

void **SetFromTop** () *set first point of line to top of fromShape.*

void **SetToBottom** () *set last point of line to bottom of toShape.*

void **SetFromBottom** ()
set first point of line to bottom of fromShape.

void **UpdatePosition** ()
recalculates position and redraws the line.

virtual void
CalcPosition () *Define a position for the line (the center of the name of the line).*

virtual void

	CalcEndPoints ()	*Calculate begin and end-points (intersections with the shapes).*
void	**CalcDirection** ()	*Calculate direction arrow*
bool	**CalcEndPoint** (const Point *pt, unsigned which)	*ReCalculate when first or last point of line is moved to pt.*
void	**SetPoint** (const Point *pt, unsigned int n)	*Update point of given position n.*
void	**AddPoint** (const Point *pt, unsigned int n)	*Add a point after given position n.*
void	**RemovePoint** (unsigned int n)	*Remove the point at given position n.*
int	**GetClassType** () const	
void	**WriteMembers** (OutputFile *f)	*See shape. Reads/writes also the points.*
bool	**ReadMembers** (InputFile *f, double format)	*See shape. Reads/writes also the points.*
bool	**SetAssocSubject** (AssocList *)	*Set subject (edge) reference,*
bool	**SetReferences** (AssocList *)	*Set also fromShape and toShape references.*
bool	**CheckReferences** ()	*Check also fromShape and toShape shape references.*
void	**SetTextShape** ()	
bool	**GetLine** (int x, int y)	*return if segment has (x, y). if so, lineNumber set to segment nr.*
unsigned int	**GetLineNumber** ()	
void	**SetLineNumber** (unsigned int n)	
TextShape*		

ChooseTextShape (int x, int y)
choose text shape near (x, y).

virtual void
MoveRaw (const Point *delta)
move all points and text shapes with delta.

Point **GiveOrthoSnp** (Point to) const

Point **GiveSnp** (Point to) const

Point **GiveSnp** (Point p, int order, int count, int &code)

void **UpdateNameDirection** (ReadDirection::Type nd)

ReadDirection::Type
GetNameDirection () const

Protected Members

void **DrawShape** () *draws a line.*

void **CalcPositionName** ()
recalculate the position of the name text shape.

static const int
LINE_DIST *max distance (x, y) where point belongs to line.*

static const int
CL_DISTX *default x distance between textshapes to line.*

static const int
CL_DISTY *default y distance between textshapes to line.*

bool **ContainsPtLine** (int x, int y, const Point *from, const Point *to, int &cx, int &cy) const
returns (x, y) is near line from-to, cx and cy are points on line.

bool **BelowLine** (int x, int y, Point *from, Point *to)
return if (x, y) is down to line from->to.

bool **AboveLine** (int x, int y, Point *from, Point *to) *return if (x, y) is above line from->to.*

void **SetNameDirection** (ReadDirection::Type nd)

void **UndrawDirection** ()

void **DrawDirection** ()

void **NotifyTextMove** (TextShape *t)

Private Members

GShape*	**fromShape**	*The first connected shape.*
GShape*	**toShape**	*The other connected shape.*
List <Point *> *	**line**	*The points of the line (aka handles).*
bool	**curved**	*Is the line curved?*
LineEnd	**end1**	
LineEnd	**end2**	
Point*	**directionPoints**	
ReadDirection::Type	**nameDirection**	*possible read direction of name label.*
unsigned int	**lineNumber**	*The segment where the name text shape is located.*
unsigned int	**count**	*how many lines exist between fromShape and toShape.*
unsigned int	**order**	*the order in the case of multiple lines.*
int	**orientation**	*0=free, 1=horizontal, 2=vertical.*
virtual void	**DrawLine** (const Point *ep1, const Point *ep2)	
virtual void		

	DrawBezier (const Point *ep1, const Point *ep2)	*Draws a bezier curve through line's points.*
void	**DrawDuplicate** ()	*at the moment. lines cannot be duplicated.*
void	**DrawSelect** ()	*draws black handles at each point of line.*
void	**DrawGraySelect** ()	*draws gray handles.*
void	**SetHandles** (int)	*not relevant for a line.*
void	**Move** (const Point *delta)	*moves all points of line.*

211

class **LinePiece**

horizontal or vertical line (cell boundary).

Public Members

	LinePiece (Grafport *g, Point *from, Point *to, LineStyle::Type style, unsigned width)
void	**Draw** ()
void	**Undraw** ()
bool	**HitLine** (int x, int y) *return iff (x, y) is on line piece*
void	**SetLineStyle** (LineStyle::Type t)
LineStyle::Type	**GetLineStyle** ()
void	**UpdateLineStyle** (LineStyle::Type t)
void	**SetLineWidth** (unsigned n)
unsigned	**GetLineWidth** ()
void	**UpdateLineWidth** (unsigned n)
void	**SetBegin** (const Point *pt)
void	**SetEnd** (const Point *pt)
void	**SetPoints** (const Point *pt1, const Point *pt2)
void	**UpdatePoints** (const Point *pt1, const Point *pt2)
void	**UpdateBegin** (const Point *pt)
void	**UpdateEnd** (const Point *pt)
void	**SetGrafport** (Grafport *g)
void	**Write** (OutputFile *f)
bool	**Read** (InputFile *f, double format)
Point*	**GetBegin** ()
Point*	**GetEnd** ()
bool	**IsVisible** ()
void	**SetVisible** (bool b)

Private Members

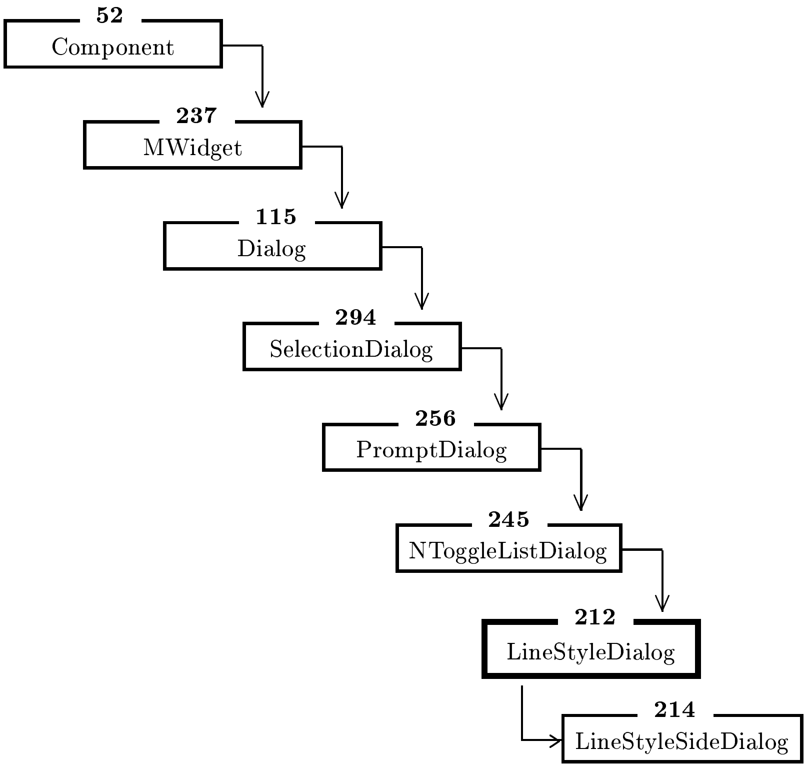
Point	**beginP**
Point	**endP**
bool	**visible**
LineStyle::Type	**lineStyle**
unsigned	**lineWidth**
Grafport*	**grafport**

212

class **LineStyleDialog** : public NToggleListDialog

line style chooser dialog class.

Inheritance



Public Members

LineStyleDialog (Widget parent)

const char*
GetClassName ()

Protected Members

virtual void

MakeLabels (List<string *> *labels)

virtual void
MakeItems (List<List<string *> *> *items)

virtual void
MakeBitmaps ()

virtual void
MakeLiteClues ()

void **Configure** ()

213

class **LineStyle**

different line styles

Public Members

enum **Type**

static Type
String2Type (const string* s)
e.g. convert "Solid" to SOLID.

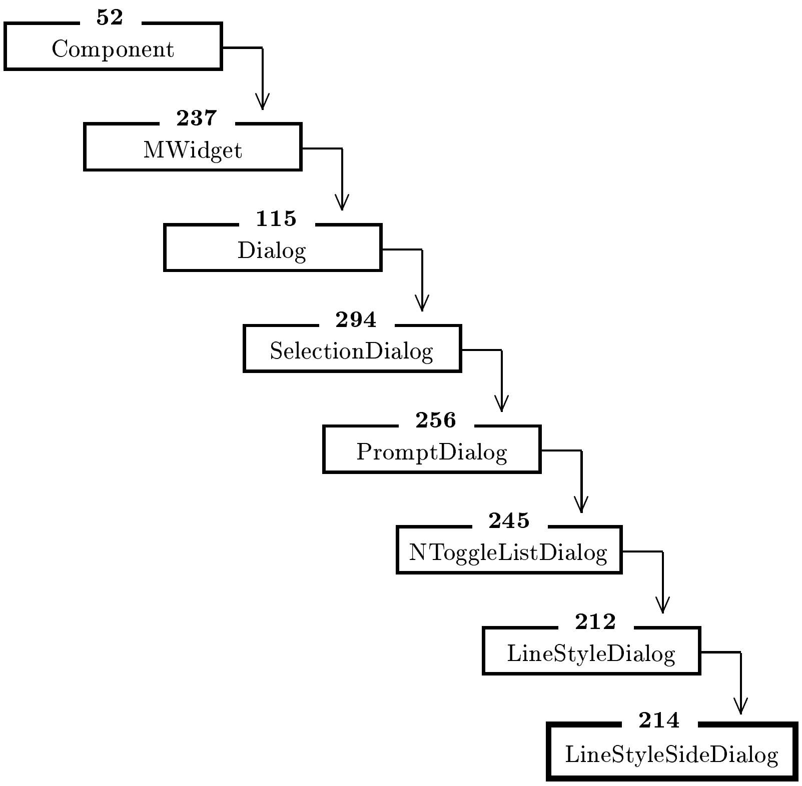
static void
Type2String (Type t, string* s)
e.g. convert SOLID to "Solid".

214

class **LineStyleSideDialog** : public LineStyleDialog

line style chooser dialog class with extra option to choose a side

Inheritance



Public Members

LineStyleSideDialog (Widget parent)

const char*
GetClassName ()

Protected Members

void **MakeLabels** (List<string *> *items)

void **MakeItems** (List<List<string *> *> *items)

void **MakeBitmaps** ()

void **MakeLiteClues** ()

Private Members

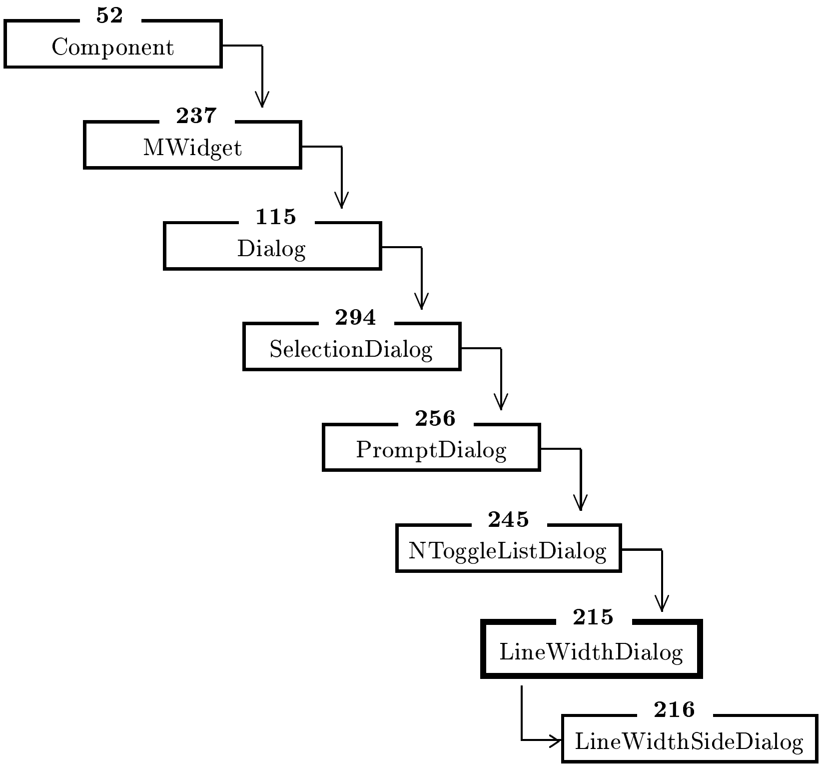
void **Configure** ()

215

class **LineWidthDialog** : public NToggleListDialog

line width chooser dialog class.

Inheritance



Public Members

LineWidthDialog (Widget parent)

const char*
GetClassName ()

Protected Members

virtual void

MakeLabels (List<string *> *labels)

virtual void
MakeItems (List<List<string *> *> *items)

virtual void
MakeBitmaps ()

virtual void
MakeLiteClues ()

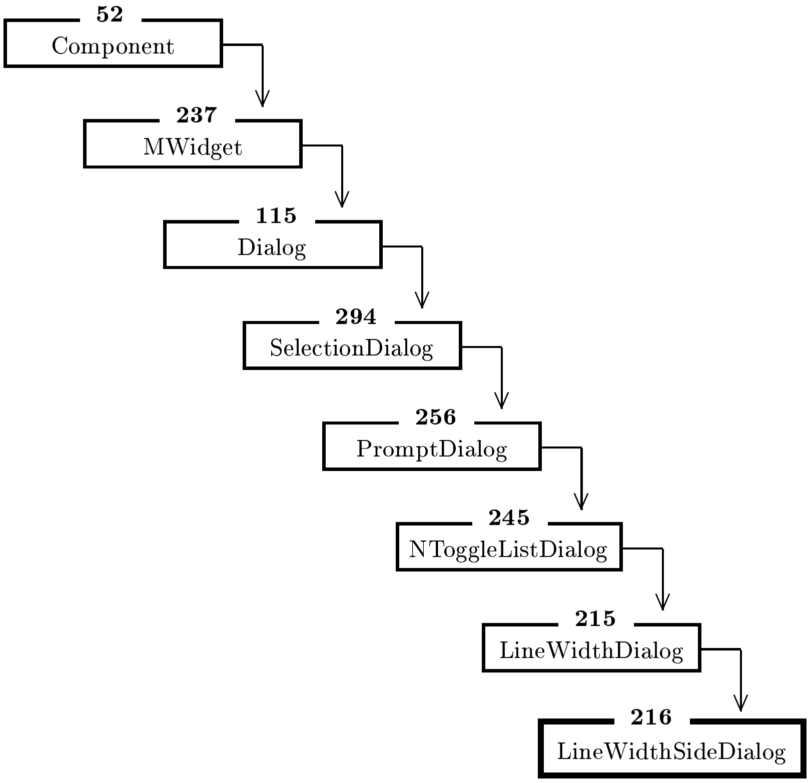
void **Configure** ()

216

class **LineWidthSideDialog** : public LineWidthDialog

line width chooser dialog class with extra option to choose a side

Inheritance



Public Members

LineWidthSideDialog (Widget parent)

const char*
GetClassName ()

Protected Members

void	**MakeLabels** (List<string *> *items)
void	**MakeItems** (List<List<string *> *> *items)
void	**MakeBitmaps** ()
void	**MakeLiteClues** ()

Private Members

void	**Configure** ()

217

class **Link**

holds all info relevant to one link.

Public Members

void	**Open** (const char* name, const char* param)	*open a link to another program named name, passing a param*
void	**Close** ()	*Close the link to a program that has terminated. Use Kill*
int	**Read** (char* s)	*read from the program started with Open. Returns a 0 if*
int	**InputWaiting** ()	*Returns the number of bytes waiting in the input buffer. If*
void	**WriteChar** (char c)	*write a char, without a newline, to the program.*
void	**Write** (const char* s)	*write a string to the program, with a newline.*
void	**Kill** ()	*kill the program and close the link. If the program has terminated*

Private Members

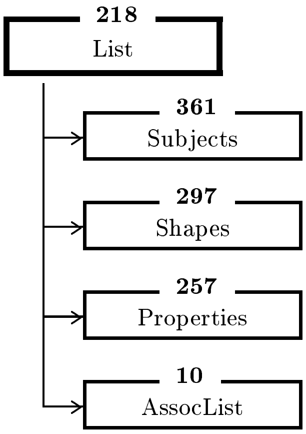
int	**pipefd1** [2]
int	**pipefd2** [2]
int	**pid**
FILE*	**fpin**
FILE*	**fpout**

218

template <class T> class **List**

lists of objects of arbitrary type T.

Inheritance



Public Members

	List ()	*create an empty list.*
virtual	**~List** ()	*delete entire list (but not the elements of the list).*
	List (const List<T> &l)	*create list, initialized by elements of l.*
List <T> &	**operator=** (const List<T> &l)	*copy list into an other, elements are copied too (e.g. l2 = l1).*
void	**add** (const T &e)	*add e to the end of the list.*
void	**insert** (const T &e, const unsigned i)	*insert e to i+1'th position of the list.*
int	**find** (const T &e) const	

		return index of first element equal to e, if not found, return -1.
bool	**isSet** () const	*return list is set*
int	**count** (const T &e) const	*count the number of times e occurs in the list*
bool	**contains** (List<T> *l) const	*return if list contains list l*
bool	**contains** (const T &e) const	*return if list contains e.*
void	**remove** (const T &e)	*remove all elements equal to e.*
void	**removei** (const unsigned i = 0)	*removes i+1'th element (0 <= i <= count()-1).*
bool	**removecur** ()	*removes current element and advances the current to the next one.*
void	**empty** ()	*removes all elements from list.*
virtual void	**clear** ()	*removes all elements + free elements (which should be pointers).*
T&	**operator**[] (const unsigned i) const	*returns ref. to i+1'th element, like array index (e.g. e = l[5]).*
T&	**elem** (const unsigned i) const	*same as [] (different syntax).*
unsigned	**count** () const	*returns number of elements in list.*
void	**replace** (const T &old, const T &fresh)	*replaces all elements equal to old by new.*
void	**sort** (int (*cmp)(T, T))	*sort all elements according to cmp function.*
void	**reverse** ()	*reverse all elements in list.*

int	**first** ()	*return if there is a first element + sets current to first element.*
int	**last** ()	*sets current to last element + returns if there is a last element.*
int	**next** ()	*sets current to next element + returns if there is a next element.*
int	**prev** ()	*sets current to prev. element + returns if there is a prev. element.*
T&	**cur** () const	*returns current element (Check first with 'done'!)*
bool	**done** () const	*returns if current is out of list.*
int	**setcur** (const T &)	*returns if T is in list + sets current to that element.*

Private Members

Element <T> *	**head**	*head of the list.*
Element <T> *	**tail**	*tail of the list.*
Element <T> *	**current**	*current element.*
unsigned	**length**	*length of the list.*

219

class **string**

infinite length string class.

Public Members

static const string

EMPTY *the empty string*

string () *create empty string.*

string (const string &s) *copy constructor. usage: string x = string*

string (const char* s) *copy constructor. usage: string x = "abc"*

string (char c) *copy constructor. usage: string x = 'a'*

string (int n) *copy constructor (decimals). usage: string x = 12;*

string (unsigned n) *copy constructor (decimals). usage: string x = 12;*

string (double d) *copy constructor (doubles). usage: string x = 1.2;*

virtual **~string** () *delete string.*

string& **operator=** (const string &s) *copy operator a la strcpy.*

string& **operator=** (const char* s) *copy operator a la strcpy.*

string& **operator=** (char c) *copy operator a la strcpy.*

string& **operator=** (int n) *copy operator a la strcpy. s = n in decimal notation.*

string& **operator=** (unsigned n) *copy operator a la strcpy. s = n in decimal notation.*

string& **operator=** (double d)

copy operator a la strcpy. s = d in floating point notation.

string& **operator+=** (const string &s)
concat operator a la strcat.

string& **operator+=** (const char* s)
concat operator a la strcat.

string& **operator+=** (int n)
concat. n in decimal notation is added.

string& **operator+=** (unsigned n)
concat. n in decimal notation is added.

string& **operator+=** (double d)
concat. d in floating point notation is added.

string& **operator+=** (char c)
concat operator a la strcat.

friend string
operator+ (const char* s1, const string &s2)
constructive concat operator.

friend string
operator+ (const string &s1, const char* s2)
constructive concat operator.

string **operator+** (const string &s)
constructive concat operator.

string **operator+** (const char* s)
constructive concat operator.

void **add** (char c) *add c at end of string.*

void **insert** (char c, int pos)
add character at n-th position of string.

int **contains** (const string &s) const
return 1 if target is in string. 0 for failure.

int **contains** (const char* s) const
return 1 if target is in string. 0 for failure.

int **contains** (char c) const

		return 1 if target is in string. 0 for failure.
int	**startsWith** (const string &s) const	*return 1 if string starts with target, 0 for failure.*
int	**startsWith** (const char* s) const	*return 1 if string starts with target, 0 for failure.*
int	**endsWith** (const string &s) const	*return 1 if string ends with target, 0 for failure.*
int	**endsWith** (const char* s) const	*return 1 if string ends with target, 0 for failure.*
int	**setSuffix** (const string &s, char delim = '.')	*adds suffix s. return 1 when suffix is changed, 0 when only added.*
int	**getSuffix** (string &s, char delim = '.') const	*puts suffix in s. when there is no suffix returns 0, 1 otherwise.*
char&	**operator**[] (int i)	*selection of char by index a la array.*
const char&	**operator**[] (int i) const	*selection of char by index a la array.*
void	**remove** ()	*remove last character of string.*
void	**remove** (int pos)	*remove pos-th character of string.*
int	**replace** (char old, char nw)	*replace each occurence of old char by nw char .*
int	**replace** (const string &pat, const string &repl, int casesens=1)	*global substitution.*
int	**replaceFirst** (const string &pat, const string &repl, int casesens=1)	

		first occurrence substitution.
void	**addPlural** ()	*add to word an english plural ending, like horse->horses, sky->skies.*
friend string	**upcase** (const string &s)	*move all chars to upper case*
friend string	**downcase** (const string &s)	*move all chars to lower case*
friend string	**stripblanks** (const string &s)	*strip all spurious blank chars.*
void	**upcase** ()	*in-place version of the above.*
void	**downcase** ()	
void	**stripblanks** ()	
unsigned long	**toulong** () const	
long	**tolong** () const	
double	**todouble** () const	
int	**toint** () const	
unsigned	**length** () const	*number of characters in string.*
unsigned	**lines** () const	*the number of lines in string when printed: newlines+1.*
unsigned	**width** () const	*return nr. of chars of line with most chars in string.*
int	**whichline** (int i, int &pos) const	*return which line has i-th char. pos becomes position in line.*
int	**nthInline** (int line, int n) const	*return position of n-th char in line-th line.*
unsigned	**letters** () const	*return number of letters (a-z, A-Z) in string.*
void	**setPrecision** (unsigned p)	

set precision when string is interpreted as double.

friend ostream&
operator<< (ostream &o, const string &s)
write string to output stream.

friend istream&
operator>> (istream &i, string &s)
read string from input stream.

friend int **operator==** (const string &x, const char* s)
comparison operator returns 1 if equal

friend int **operator==** (const string &x, const string &y)
comparison operator returns 1 if equal.

friend int **operator%=** (const string &x, const char* s)
comparison operator that ignores case

friend int **operator%=** (const string &x, const string &y)
comparison operator that ignores case

friend int **operator!=** (const string &x, const char* s)
comparison operator a la !equal.

friend int **operator!=** (const string &x, const string &y)
comparison operator a la !equal.

friend int **compare** (const string &x, const char* s)
comparison operator a la strcmp

friend int **compare** (const string &x, const string &y)
comparison operator a la strcmp

const char*
getstr () const *return a const pointer to the data to do dirty stuff.*

Protected Members

int **doReplace** (const string &pat, const string &repl, int global, int casesens=1)

Private Members

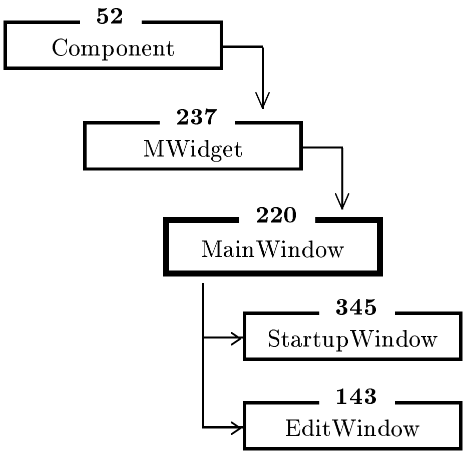
char*	**str**	*pointer to data.*
int	**len**	*the length of the data.*
int	**precision**	*the precision used for floats and doubles.*
static const int	**CHUNK**	*length in which data is allocated.*

220

class **MainWindow** : public MWidget

class supporting the main (toplevel) window.

Inheritance



Public Members

MainWindow (const char *name)
Constructor requires only a name

virtual **~MainWindow** ()

const char*
GetClassName ()

virtual void
Initialize (int argc, char** argv, Config *config)
create toplevel widget, and other class members.

virtual void
Finalize () *initialization to do when window is already managed.*

virtual void
Manage () *popup the window*

virtual void

Unmanage () *pop down the window*

virtual void
Iconify () *iconify the window*

void **ManageUnmapped** ()

void **Close** () *close the window.*

void **SetTitle** (const char *title)
Set window title.

void **SetTitle** (const string *title)

void **SetIconName** (const char *name)
Set name of window icon.

void **SetIconPixmap** (Pixmap px)
Set pixmap of window icon.

void **SetCursor** (int cursorNum)

void **SetSizeCursor** (int cursorNum)

virtual int **GetTool** ()

virtual void
SetStatus (const char *)

virtual void
SetStatus (const string *)

PrinterOptions*
GetPrinterOptions ()

Protected Members

virtual Widget
CreateWorkArea (Widget, int argc, char** argv)
Derived classes define this to create the appl-specific work area.

virtual void
ParseUserOptions (int argc, char** argv)
Derived classes use this functions to parse argv.

virtual void
CreateComponents (Widget window)
Functions to create the constituting parts of the main window.

MouseCursor*

	GetCursor ()	
Config*	**GetConfig** ()	

Private Members

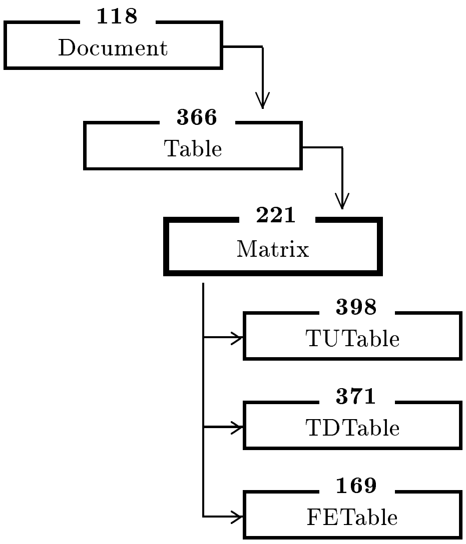
Widget	**main**	*The MainWindow widget.*
MouseCursor*	**cursor**	
Widget	**workarea**	
Config*	**config**	*the configuration object.*
void	**WidgetDestroyed** ()	

221

class **Matrix** : public Table

a table with some constraints on row and column zero.

Inheritance



Public Members

	Matrix (Config *c, TableWindow *t, TableViewer *v)
void	**CheckDocument** ()
bool	**UpdateText** (Cell *c, const string *text)

Protected Members

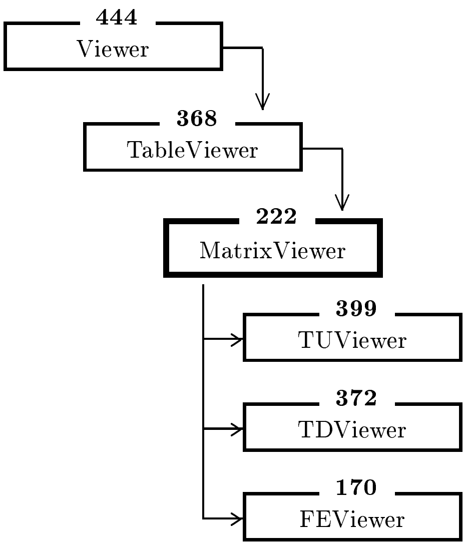
virtual bool	**CheckTextString** (Cell *c, const string *text)
unsigned	**CheckEmptyColumnEntries** (unsigned row)
unsigned	**CheckEmptyRowEntries** (unsigned column)

222

class **MatrixViewer** : public TableViewer

a table in which column and row 0 are different in layout.

Inheritance



Public Members

	MatrixViewer (Config *c, TableWindow *t)
void	**Initialize** ()
void	**AddRow** (CellRow *c)
void	**AddColumn** (CellColumn *c)
void	**InsertRow** (CellRow *c)
void	**InsertColumn** (CellColumn *c)

Private Members

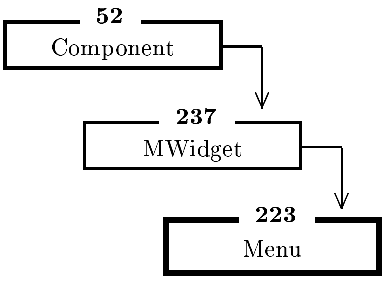
void	**SetNthLine** (CellVector *v, int n, LineStyle::Type t)

223

class **Menu** : public MWidget

Class for menu manipulation.

Inheritance



Public Members

enum **MenuType**

Menu (Widget parent, MenuType menuType, const char *name, char mnemonic, MenuItem *items)

const char* **GetClassName** ()

Widget **GetNthMenuItem** (int n)

Widget **GetMenuItem** (const char *label)

static Widget **GetChild** (Widget menu, MenuType type, const char *label)

static Widget **GetNthChild** (Widget menu, MenuType type, int n)

static void

SetChildToggleState (Widget menu,
MenuType type,
const char *label,
bool state,
bool notify=False)

static void
SetNthChildToggleState (Widget menu,
MenuType type, int n,
bool state,
bool notify=False)

static void
SetChildSensitive (Widget menu,
MenuType type,
const char *label, bool state)

static void
SetNthChildSensitive (Widget menu,
MenuType type, int n,
bool state)

Private Members

MenuType **menuType**

Widget **BuildMenu** (Widget parent, MenuType type,
const char *label, char mnem,
MenuItem *items)

224

class **MenuItem**

data structure that comprises all the important aspects of a menu item.

Public Members

enum	**ItemType**	
const char*	**label**	*the label for the item*
Bitmap	**bitmap**	
bool	**showBitmap**	
ItemType	**itemType**	*pushbutton, submenu, checkbox, radiobox, separator ..*
char	**mnemonic**	*mnemonic; 0 character if none*
const char*	**accelerator**	*accelerator; 0 if none*
const char*	**accelText**	*to be converted to compound string*
XtCallbackProc	**callback**	*routine to call; 0 if none*
XtPointer	**callbackData**	*client data for callback()*
XtPointer	**userData**	*user data for callback.*
MenuItem*	**subitems**	*pullright menu items, if not: 0*
int	**sensitive**	*0: insensitive, 1: sensitive*
	MenuItem ()	
	MenuItem (const char *l, ItemType i)	
	MenuItem (const char *l, ItemType i, char m, const char *a, const char *t, XtCallbackProc c, XtPointer d, XtPointer u, MenuItem *si, int s)	

MenuItem (const char *l, const Bitmap *b, ItemType i, char m, const char *a, const char *t, XtCallbackProc c, XtPointer d, XtPointer u, MenuItem *si, int s)

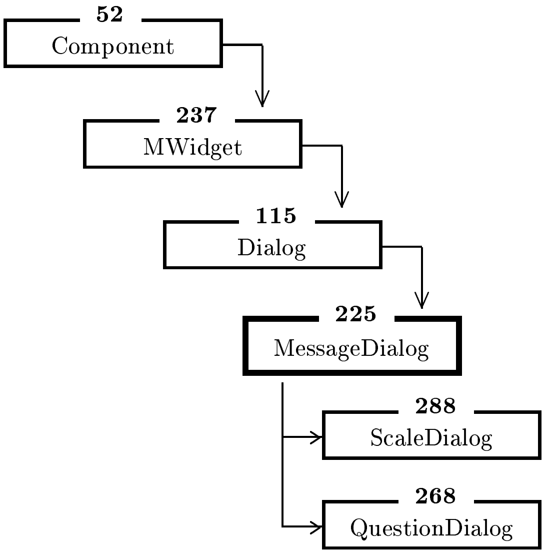
static const MenuItem **NUL**

static const MenuItem **SEP**

225 class **MessageDialog** : public Dialog

message dialog class.

Inheritance



Public Members

enum **DialogType**

MessageDialog (Widget parent)

MessageDialog (Widget parent, DialogType t)

const char* **GetClassName** ()

void **Popup** ()

virtual void **Show** (const char *title, const char *msg)

virtual void **Show** (const char *title, const string *msg)

void **SetMessageString** (const string *s) void **SetMessageString** (const char *s)

Protected Members

void **CreateWidget** ()

void **Configure** ()

Private Members

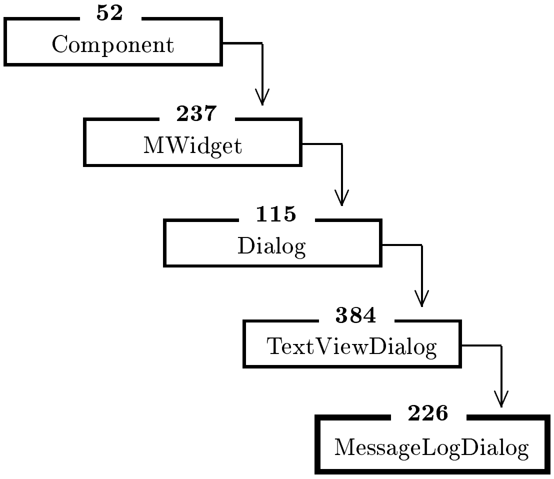
DialogType **dialogType**

226

class **MessageLogDialog** : public TextViewDialog

Dialog to display a message log.

Inheritance



Public Members

	MessageLogDialog (Widget parent)
const char*	**GetClassName** ()
void	**InsertText** (const char *txt)
void	**InsertText** (const string *txt)

Protected Members

void	**Configure** ()

Private Members

XmTextPosition

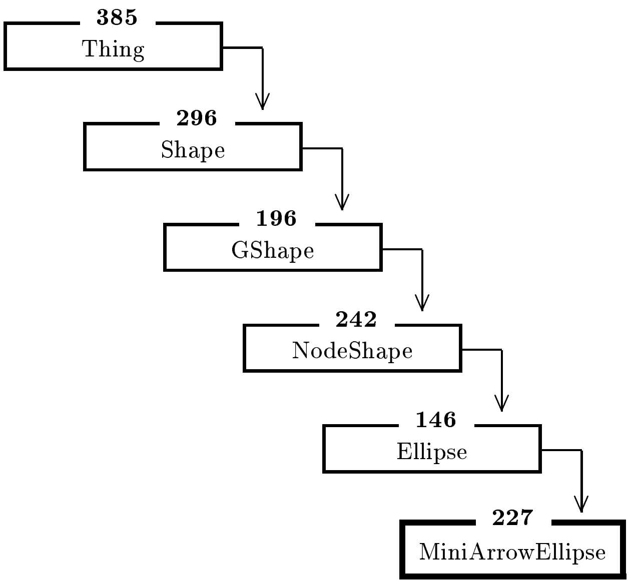
	wprPosition	
unsigned	**count**	
char	**timeAt** [32]	*shows time in error message.*

227

class **MiniArrowEllipse** : public Ellipse

small circle having small arrow and label on top.

Inheritance



Public Members

static const int
ARROWLEN

static const int
CIRCLE_WIDTH

MiniArrowEllipse (ShapeView *v, Grafport *g,
int x, int y,
int w = CIRCLE_WIDTH,
int h = CIRCLE_WIDTH)

MiniArrowEllipse (const MiniArrowEllipse &c)

MiniArrowEllipse (const NodeShape &c)

void **SetPosition** (const Point *p, bool snap=True)

void	**SetAlignment** (TextAlign::Type alignment)
void	**SetFont** (XFont *font)
void	**SetName** (const string *s)
TextShape*	**ChooseTextShape** (int x, int y)
Shape*	**Clone** ()
int	**GetClassType** () const
void	**SetTextShape** ()
Point	**GiveSnp** (Point to) const
Point	**GiveSnp** (Point to, int order, int count, int &code)
bool	**HasString** (const string *s, bool sens, bool sub) const
bool	**HasNameString** (const string *, bool, bool) const
bool	**HasString** (const string*, bool, bool, List<TextShape *> *)

Private Members

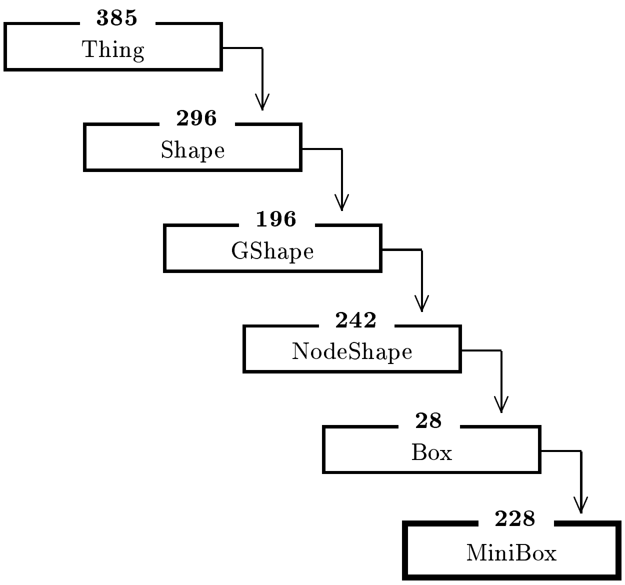
LineEnd	**end**
void	**DrawShape** ()
void	**CalcPositionName** ()
int	**textHeight**
void	**Init** ()
bool	**HasMinSize** ()

228

class **MiniBox** : public Box

small box class, having fixed min. width and height.

Inheritance



Public Members

	MiniBox (ShapeView *v, Grafport *g, int x, int y, int w = MIN_WIDTH, int h = MIN_HEIGHT)
	MiniBox (const NodeShape &c)
Shape*	**Clone** ()
int	**GetClassType** () const

Private Members

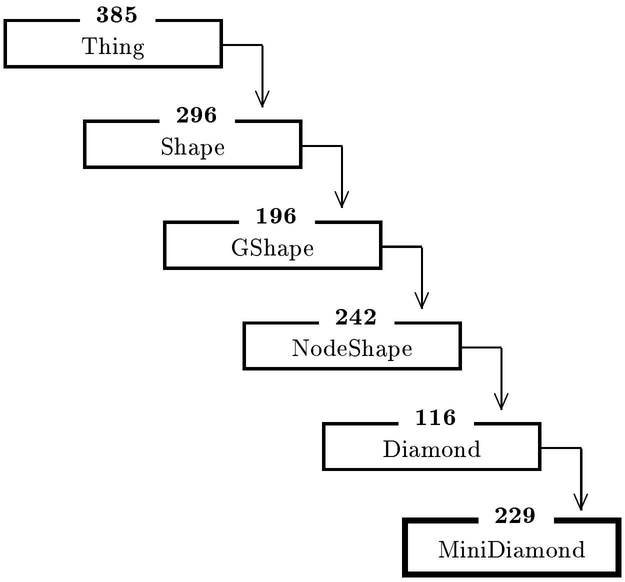
void	**Init** ()
bool	**HasMinSize** ()

229

class **MiniDiamond** : public Diamond

mini-diamond shape class.

Inheritance



Public Members

	MiniDiamond (ShapeView *v, Grafport *g, int x, int y, int w = MIN_WIDTH, int h = MIN_HEIGHT)
	MiniDiamond (const NodeShape &c)
Shape*	**Clone** ()
int	**GetClassType** () const

Private Members

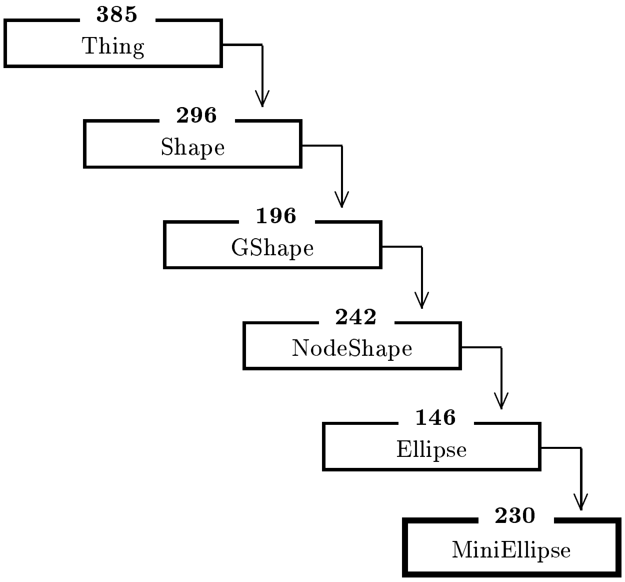
void	**Init** ()
bool	**HasMinSize** ()

230

class **MiniEllipse** : public Ellipse

mini-ellipse shape class.

Inheritance



Public Members

	MiniEllipse (ShapeView *v, Grafport *g, int x, int y, int w = MIN_WIDTH, int h = MIN_HEIGHT)
	MiniEllipse (const MiniEllipse &c)
	MiniEllipse (const NodeShape &c)
Shape*	**Clone** ()
enum	**NamePositionType**
void	**SetNamePosition** (NamePositionType t)
int	**GetClassType** () const
void	**SetPosition** (const Point *p, bool snap)

void **SetName** (const string *s)

void **SetFont** (XFont *f)

void **SetAlignment** (TextAlign::Type t)

void **SetSize** (int w, int h)

void **SetTextShape** ()

Private Members

void **Init** ()

bool **HasMinSize** ()

void **CalcPositionName** ()

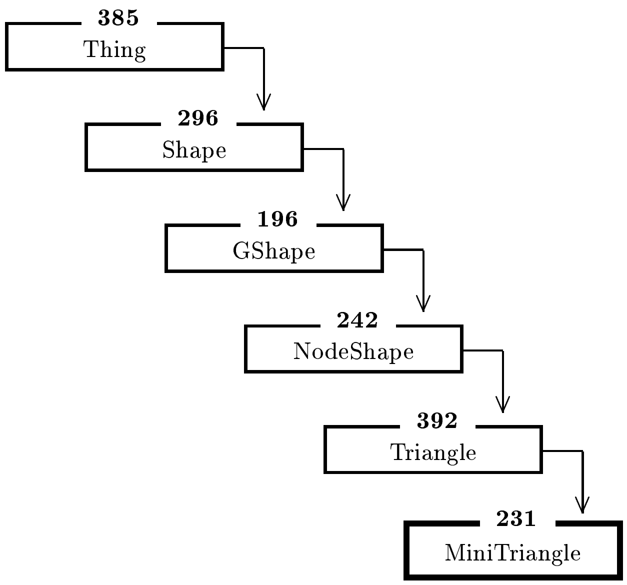
NamePositionType **namePosition**

231

class **MiniTriangle** : public Triangle

mini-triangle shape class.

Inheritance



Public Members

	MiniTriangle (ShapeView *v, Grafport *g, int x, int y, int w = MIN_WIDTH, int h = MIN_HEIGHT)
	MiniTriangle (const NodeShape &c)
Shape*	**Clone** ()
int	**GetClassType** () const

Private Members

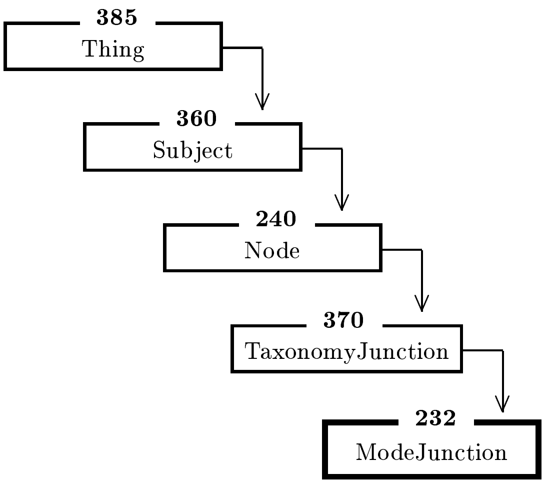
void	**Init** ()
bool	**HasMinSize** ()

232

class **ModeJunction** : public TaxonomyJunction

mode junction (CRD) node class.

Inheritance



Public Members

	ModeJunction (CRGraph *g)
Subject*	**Clone** ()
int	**GetClassType** () const

233

class **MouseCursor**

functions to set mouse cursor to different shapes.

Public Members

	MouseCursor (Widget w)	
static const int	**WATCH**	*cursor shapes.*
static const int	**LEFT_PTR**	
static const int	**PLUS**	
static const int	**TERM**	
static const int	**FLEUR**	
static const int	**H_DOUBLE_ARROW**	
static const int	**V_DOUBLE_ARROW**	
void	**Set** (int cursorNum)	*change the cursor to cursorNum (see /usr/include/X11/cursorfont.h).*
void	**SetSize** (int place)	*change the cursor to a resize cursor depending on the place.*

Private Members

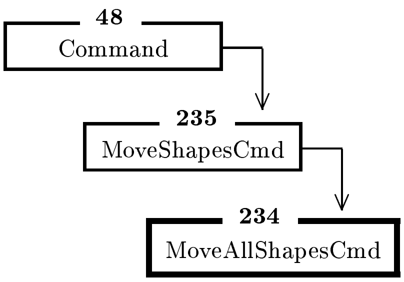
bool	**normalPtr**
Widget	**widget**

234

class **MoveAllShapesCmd** : public MoveShapesCmd

class for moving shapes in certain direction (move type).

Inheritance



Public Members

MoveAllShapesCmd (ShapeView *v, Document::MoveType t)

void **UnExecute** ()

void **Execute** ()

const char* **GetName** () const

Private Members

Document::MoveType **moveType**

static const int **MOVE_DISTANCE**

Point **delta**

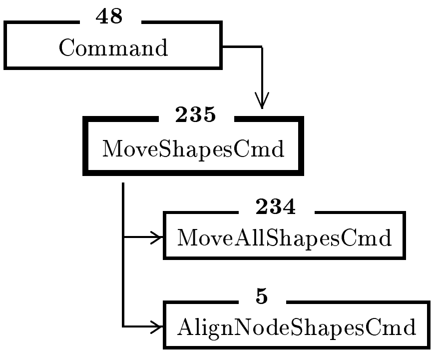
void **UpdateMoveShapes** (List<GShape *> *s, const Point *delta)

235

class **MoveShapesCmd** : public Command

abstract command class for moving shapes

Inheritance



Public Members

MoveShapesCmd (ShapeView *v, List<GShape *> *s)

virtual **~MoveShapesCmd** ()

virtual void
Execute ()

void **UnExecute** ()

const char*
GetName () const

Protected Members

List <GShape *> *
GetShapes () const

ShapeView*
GetView () const

Private Members

List <Point *> *
oldPositions

List <GShape *> *
shapes

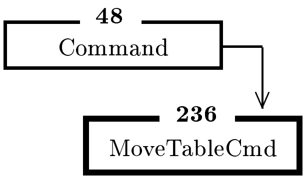
ShapeView*
view

236

class **MoveTableCmd** : public Command

move entire table in mainwindow.

Inheritance



Public Members

	MoveTableCmd (Table *t, TableViewer *v, Document::MoveType m)
void	**Execute** ()
void	**UnExecute** ()
const char*	**GetName** () const

Private Members

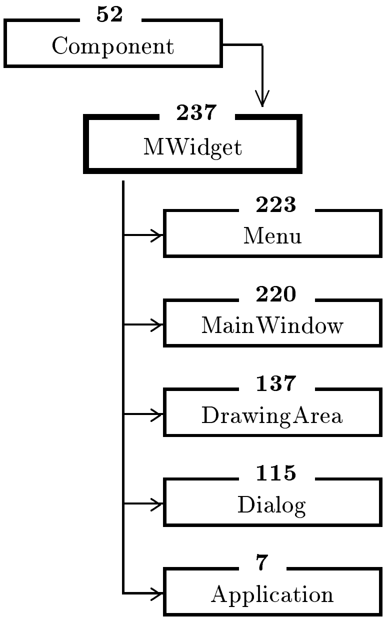
Document::MoveType	**moveType**
Point	**oldPosition**
static const int	**MOVE_DISTANCE**

237

class **MWidget** : public Component

base class for wrapping all Motif widgets.

Inheritance



Public Members

	MWidget (const char *name)	
virtual	**~MWidget** ()	
virtual void	**Manage** ()	
virtual const char*	**GetClassName** ()	
bool	**IsRealized** ()	*Is the widget realized ?*
void	**SetColormap** (Colormap cmap)	

Set the colormap of the window to cmap.

void **SetDestroyCallback** (XtCallbackProc fun, XtPointer clientData)
Add a widget destroyed callback.

void **AddLiteClue** (Widget w, const char *text)

Protected Members

Widget **GetLiteClue** ()

void **CreateLiteClue** ()

void **InstallDestroyHandler** ()
Easy hook for derived classes

Private Members

virtual void **WidgetDestroyed** ()
Called by widgetDestroyedCallback() if base widget is destroyed

void **SetDefaultResources** (const Widget, const char** resources)
Loads component's default resources into database

void **GetResources** (const XtResourceList, const int)
Retrieve resources for this clsss from the resource manager

static void **WidgetDestroyedCB** (Widget, XtPointer, XtPointer)
Interface between XmNdestroyCallback and this class

Widget **liteClue** *Widget for popping up online help messages.*

238

class **NodeAlign**

different node aligment options

Public Members

enum **Type**

static Type
String2Type (const string *s)
e.g. convert "Top" to TOP.

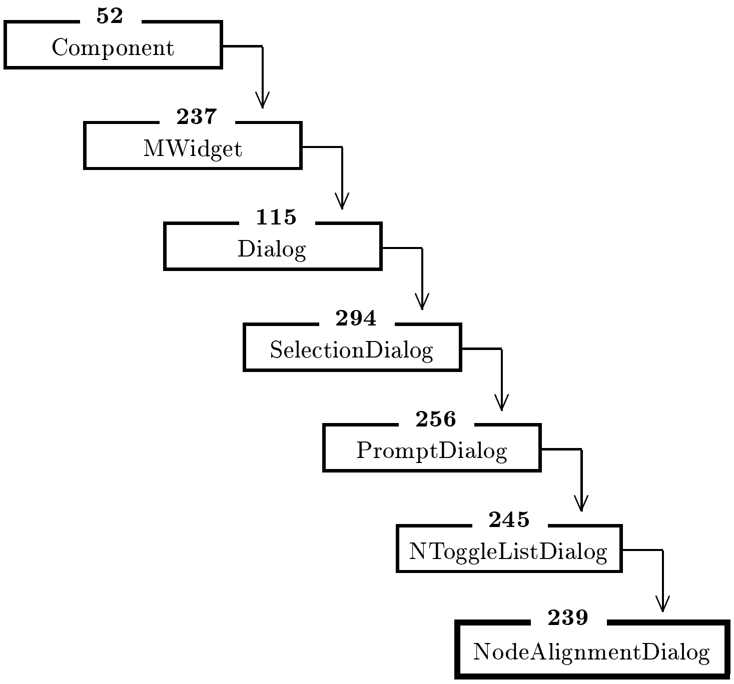
static void
Type2String (Type t, string *s)
e.g. convert TOP to "Top".

239

class **NodeAlignmentDialog** : public NToggleListDialog

node alignment chooser dialog class.

Inheritance



Public Members

	NodeAlignmentDialog (Widget parent)
const char*	**GetClassName** ()

Private Members

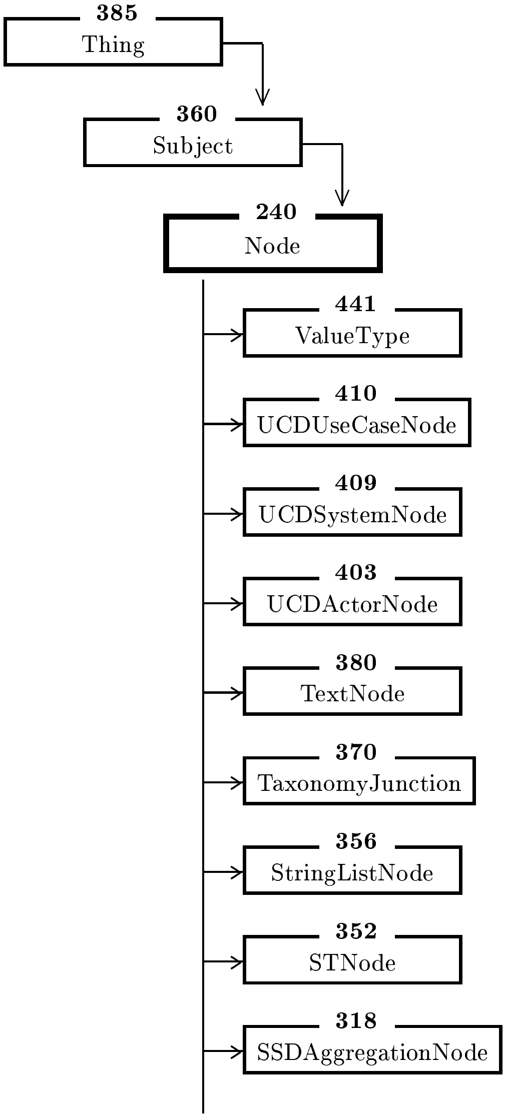
void	**Configure** ()

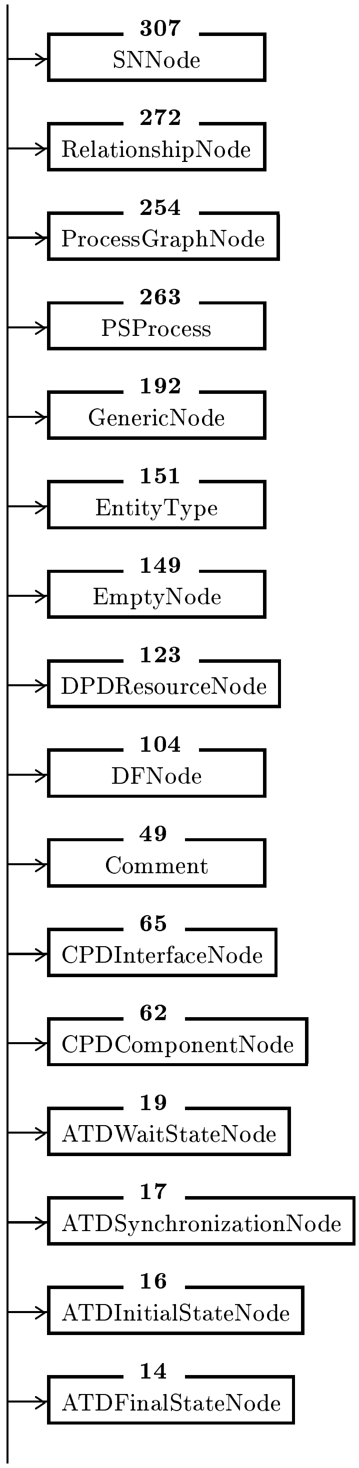
240

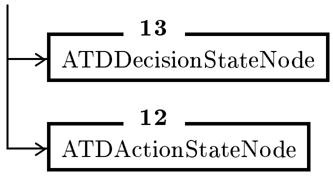
class **Node** : public Subject

Node class (base class for all specific node classes).

Inheritance




307
SNNode
272
RelationshipNode
254
ProcessGraphNode
263
PSProcess
192
GenericNode
151
EntityType
149
EmptyNode
123
DPDResourceNode
104
DFNode
49
Comment
65
CPDInterfaceNode
62
CPDComponentNode
19
ATDWaitStateNode
17
ATDSynchronizationNode
16
ATDInitialStateNode
14
ATDFinalStateNode



Public Members

	Node (Graph *g)	
bool	**IsEdge** () const	*a node is not an edge*
bool	**InGraph** () const	*is node in the graph?*
const string*	**GetIndex** () const	
virtual Subject::NameErrType	**SetIndex** (const string *s)	*set and check index*
void	**WriteMembers** (OutputFile *f)	
bool	**ReadMembers** (InputFile *f, double format)	
void	**GetParentIndex** (string *pindex)	*put in pindex the index of the parent node.*

Private Members

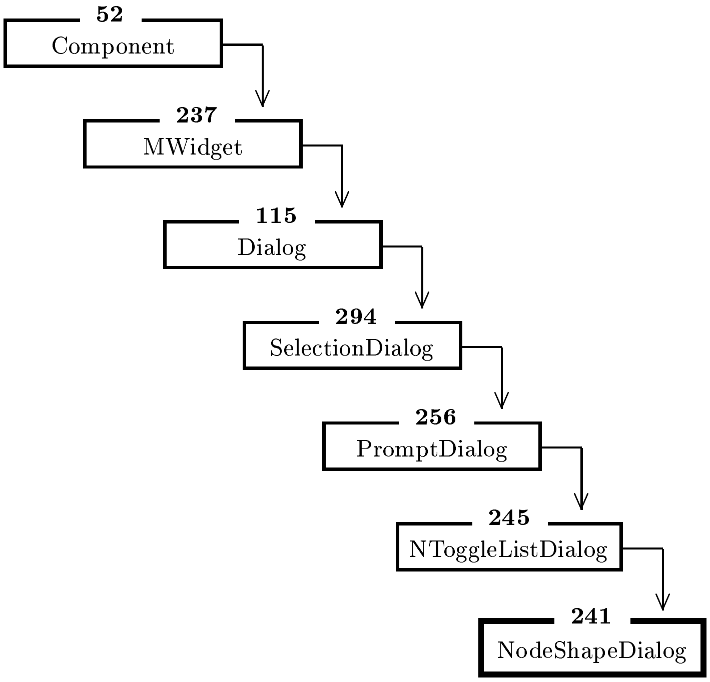
string	**index**	*an index label (not necessarily used).*

241

class **NodeShapeDialog** : public NToggleListDialog

node shape chooser dialog class.

Inheritance



Public Members

NodeShapeDialog (Widget parent)

const char*
GetClassName ()

Private Members

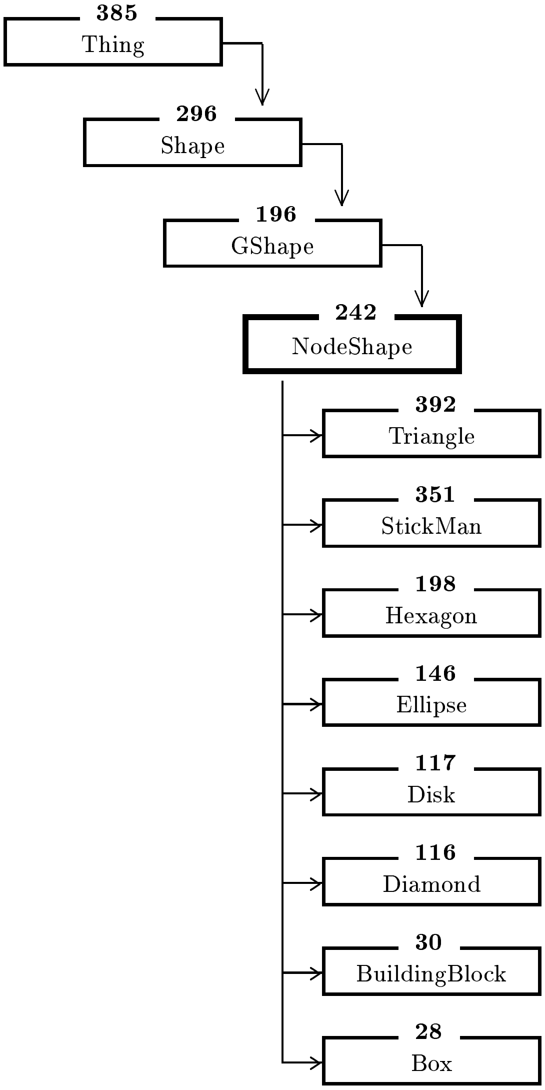
void **Configure** ()

242

class **NodeShape** : public GShape

(abstract) graphical node shape class, superclass of box, ellipse etc.

Inheritance



Public Members

	NodeShape (ShapeView *v, Grafport *g, int x, int y, int w=MIN_WIDTH, int h=MIN_HEIGHT)
	NodeShape (const NodeShape &c)
virtual	**~NodeShape** ()
void	**UpdateIndexLabel** (const string *s) *update string of label.*
void	**SetIndexLabel** (const string *s)
void	**DrawDuplicate** () *draw duplicate asterisk.*
void	**DrawMultiplicity** () *draw multiplicity number.*
FillStyle::Type	**GetFillStyle** () const
void	**SetFillStyle** (FillStyle::Type f)
void	**UpdateFillStyle** (FillStyle::Type f)
const string*	**GetFillColor** () const
void	**SetFillColor** (const string *c)
void	**UpdateFillColor** (const string *c)
void	**SetPosition** (const Point *p, bool snap=True) *adjusts also the label to a new place*
void	**SetSize** (int w, int h) *adjusts also the label to a new place*
void	**SetGrafport** (Grafport *g)
void	**SetView** (ShapeView *v)
void	**SetFont** (XFont *ft)
void	**SetTextColor** (const string *color)
bool	**InTextArea** (int x, int y) *Returns if (x, y) is in text-area.*
int	**HitHandles** (int, int)
TextShape*	

	ChooseTextShape (int x, int y)	
void	**AdjustSize** ()	*update size to make it fit the text shapes.*
bool	**SetAssocSubject** (AssocList *al)	
void	**WriteMembers** (OutputFile *ofile)	*write it to file.*
bool	**ReadMembers** (InputFile *ifile, double format)	*read and parse it from file.*
void	**SetTextShape** ()	
TextShape*	**GetIndexLabel** () const	
bool	**IsFixedIndexLabel** ()	
void	**SetFixedIndexLabel** (bool b)	
void	**SetResizable** (bool b)	
bool	**IsResizable** ()	
void	**ForceSize** (int w, int ht)	
bool	**HasTextShape** (TextShape *t) const	
bool	**HasString** (const string *s, bool sens, bool sub) const	
bool	**HasString** (const string*, bool, bool, List<TextShape *> *)	

Protected Members

void	**DrawTextShapes** ()	*draw name + index text shape.*
virtual void	**CalcPositionIndexLabel** ()	

Private Members

FillStyle::Type	**fillStyle**
string	**fillColor**
TextShape*	

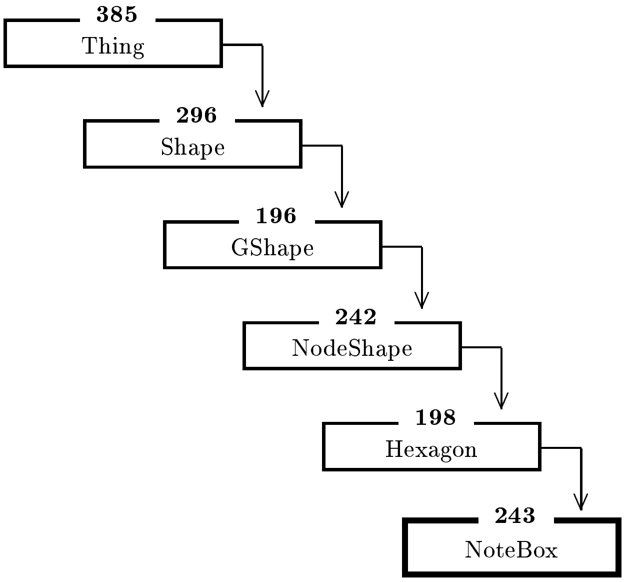
	indexLabel
bool	**fixedIndexLabel**
bool	**resizable**

243

class **NoteBox** : public Hexagon

UML note box

Inheritance



Public Members

	NoteBox (ShapeView *v, Grafport *g, int x, int y)
	NoteBox (const NodeShape &)
Shape*	**Clone** ()

Protected Members

void	**DrawOutLine** (const Point* p)
void	**SetSize** (int w, int h)
void	**DrawShape** ()

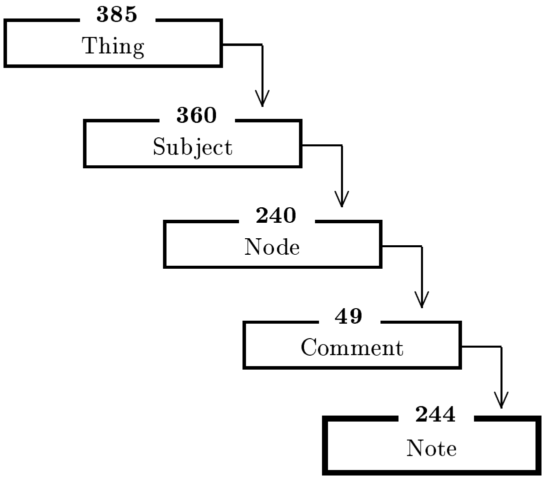
void	**CalcPoints** (int x, int y)
Point	**GiveSnp** (Point to) const
Point	**GiveSnp** (Point to, int order, int count, int& code)
int	**GetClassType** () const

244

class **Note** : public Comment

Note comment node class.

Inheritance



Public Members

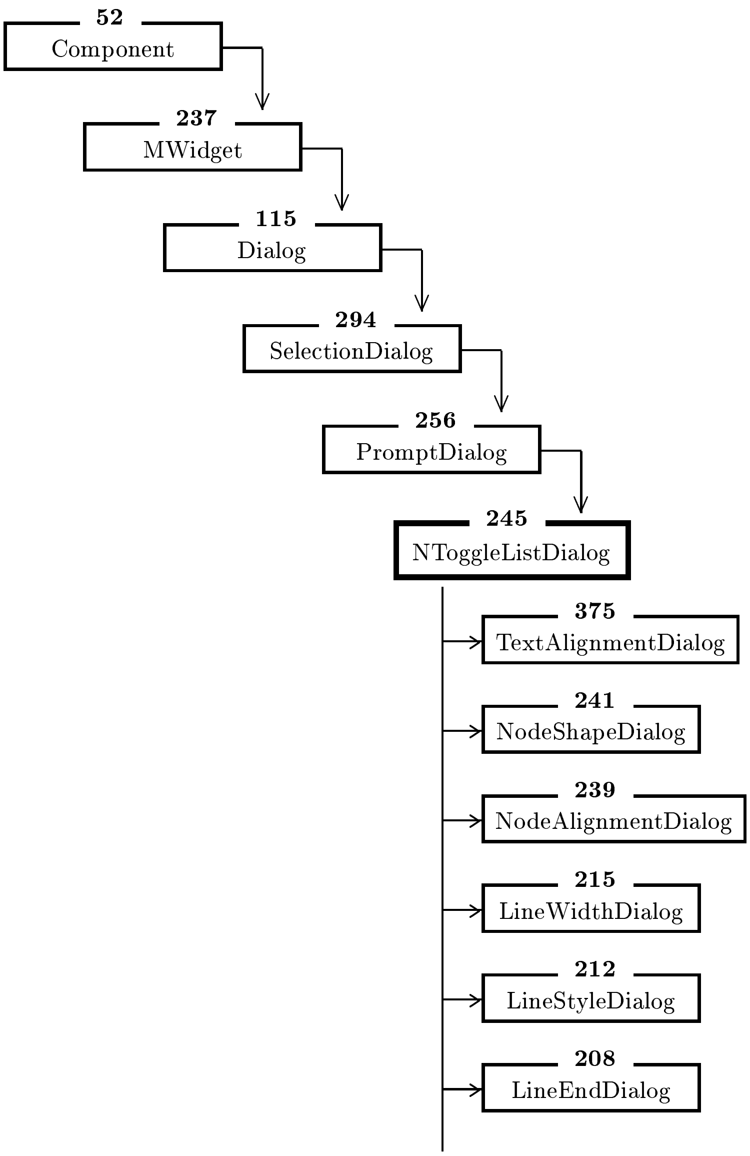
	Note (Graph *g)
Subject*	**Clone** ()
int	**GetClassType** () const

245

class **NToggleListDialog** : public PromptDialog

dialog class that shows a number of lists of radio buttons.

Inheritance



184
FontChooserDialog

Public Members

	NToggleListDialog (Widget parent, bool prompt)
virtual	**~NToggleListDialog** ()
const char*	**GetClassName** ()
void	**SetToggleChangedCallback** (unsigned n, XtCallbackProc fun, XtPointer cd)
void	**CreateToggleLists** (List<string *> *labels, List<List<string *> *> *items, List<string *> *options = 0)
void	**SetItem** (unsigned n, unsigned i, const string *text)
void	**SetItemBitmap** (unsigned n, unsigned i, Bitmap *bitmap)
void	**GetItem** (unsigned n, unsigned i, string *text)
void	**SetValue** (unsigned n, unsigned i)
void	**SetOptionValue** (unsigned n, bool b)
void	**SetValueOfText** (unsigned n, const string *text)
int	**GetValue** (unsigned n)
int	**GetOptionValue** (unsigned n)
bool	**IsPromptVisible** ()
bool	**IsOptionsVisible** ()
void	**SetPromptVisible** (bool b)
void	**SetOptionsVisible** (bool b)

Protected Members

void	**CreateWidget** ()
void	**Configure** ()

List <List<Widget> *> *
GetToggleList ()

Private Members

List <List<Widget> *>
toggleList

List <Widget>
optionList

bool **promptVisible**

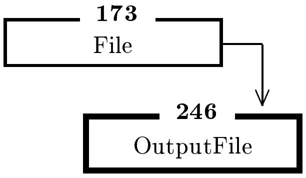
bool **optionsVisible**

246

class **OutputFile** : public File

Output file class.

Inheritance



Public Members

	OutputFile ()
virtual	**~OutputFile** ()
void	**Open** (const string* file)
void	**Close** ()
bool	**Good** ()
OutputFile&	**operator<<** (const char*)
OutputFile&	**operator<<** (char)
OutputFile&	**operator<<** (unsigned char)
OutputFile&	**operator<<** (short)
OutputFile&	**operator<<** (unsigned short)
OutputFile&	**operator<<** (int)
OutputFile&	**operator<<** (unsigned int)
OutputFile&	**operator<<** (long)
OutputFile&	

operator<< (unsigned long)

OutputFile&
operator<< (float)

OutputFile&
operator<< (double)

OutputFile&
operator<< (string)

OutputFile&
operator<< (Point)

Private Members

ofstream* **ofile** *The current output file.*

247

class **PageSize**

different page sizes

Public Members

enum **Type**

static Type
String2Type (const string* s)
e.g. convert "Legal" to LEGAL.

static Type
Substring2Type (const string* s)
e.g. convert "Legal (100x200)" to LEGAL.

static void
Type2String (Type t, string* s)
e.g. convert LEGAL to "Legal".

static int **GetWidth** (Type t)
get page width in pixels

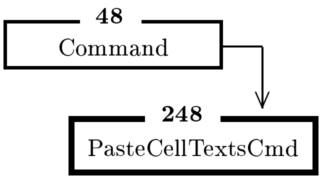
static int **GetHeight** (Type t)
get page height in pixels

248

class **PasteCellTextsCmd** : public Command

paste cell texts from cut-buffer command.

Inheritance



Public Members

	PasteCellTextsCmd (Table *t, TableViewer *v)
virtual	**~PasteCellTextsCmd** ()
void	**Execute** ()
void	**UnExecute** ()
void	**Abort** ()
void	**TrackMouse** (TrackType aPhase, Point *anchorPoint, Point *previousPoint, Point *nextPoint)
const char*	**GetName** () const

Private Members

List <CellText *> *	**cellTexts**
Point	**delta**
Point	**topLeft**
Point	**anchor**
int	**width**
int	**height**

TableViewer*
tviewer

CellTextBuffer*
buffer

List <Cell *> *
pastedCells

List <string *> *
oldStrings

void **DrawOutLine** (const Point *p)

void **MakeBox** ()

void **MoveCellTexts** (const Point *delta)

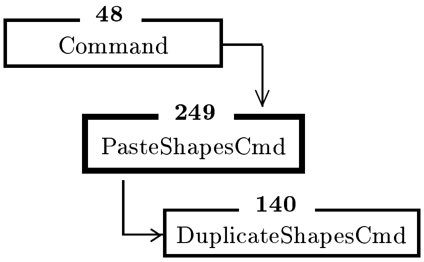
static const int
MIN_BOX_SIZE

249

class **PasteShapesCmd** : public Command

command class for pasting shapes.

Inheritance



Public Members

	PasteShapesCmd (ShapeView *v)
	PasteShapesCmd (ShapeView *v, bool)
virtual	**~PasteShapesCmd** ()
void	**MakeBox** ()
void	**Execute** ()
void	**UnExecute** ()
void	**Abort** ()
void	**TrackMouse** (TrackType aPhase, Point *anchorPoint, Point *previousPoint, Point *nextPoint)
void	**DrawOutLine** (const Point *center)
const char*	**GetName** () const

Protected Members

ShapeView*

GetView () const

List <Subject *> *
GetSubjects () const

void **SetSubjects** (List<Subject *> *s)

List <GShape *> *
GetShapes ()

void **SetShapes** (List<GShape *> *s)

const Point*
GetDelta ()

Private Members

List <Subject *> *
subjects

List <GShape *> *
shapes

Point **delta**

Point **center**

Point **anchor**

int **width**

int **height**

ShapeView*
view

Diagram* **diagram**

250

class **Point**

2-dimensional integer point ADT

Public Members

static const Point
ZERO

Point (int a = 0, int b = 0)
Construct and initialize point.

Point (const Point &p)
Copy constructor.

void **Set** (int a, int b) *Set point to (a, b).*

friend Point
operator+ (const Point p1, const Point p2)
Add points.

friend Point
operator- (const Point p1, const Point p2)
Subtract points.

friend Point
operator* (const Point p1, const Point p2)
Multiply points.

friend Point
operator/ (const Point p1, const Point p2)
Devide points.

Point **operator-** () *Subtract point from point.*

friend bool
operator== (const Point &x, const Point &y)
Return whether points are equal.

friend bool
operator!= (const Point &x, const Point &y)
Return whether points are unequal.

void **operator+=** (const Point p1)
Translate the point.

void **operator-=** (const Point p1)

		Translate the point.
void	**operator*=** (const Point p1)	*Translate the point.*
void	**operator/=** (const Point p1)	*Translate the point.*
void	**operator*** (int c)	*Multiply by constant.*
void	**operator*** (double c)	*Multiply by constant.*
friend ostream&	**operator<<** (ostream &o, const Point p)	*Write string representation to ostream.*
friend istream&	**operator>>** (istream &i, Point &p)	*Read string representation from istream.*
int	**x**	*the x-coordinate.*
int	**y**	*the y-coordinate.*

251

class **Printer**

class for printing and postscript generation.

Public Members

	Printer (Config *c, Document *d)	*constructor.*
virtual	**~Printer** ()	
enum	**OutputType**	
void	**Print** ()	*print document as postscript to current printer.*
void	**ShowPrintPreview** ()	*show print preview in separate window.*
void	**ExportAs** ()	*show popup dialog for exporting to file.*
void	**ShowPrinterQueue** ()	*show dialog containing printer queue incl.'lprm' option.*
void	**Name** ()	*popup dialog for new printer name.*
void	**Copies** ()	*popup dialog for new number of copies.*
void	**PrintCommand** ()	*popup dialog for setting printer command.*
void	**PrinterQueueCommand** ()	*popup dialog for setting printer queue command.*
void	**PrinterRemoveCommand** ()	*popup dialog for setting printer remove command.*
void	**PreviewCommand** ()	*popup dialog for setting preview command.*
void	**SetName** (const string *name)	

void **SetCopies** (int copies)

void **SetPrintCommand** (const string *command)

void **SetPageOrientation** (int num)
set page orientation to portrait or landscape.

void **SetPageNumbering** (bool b)
set page numbering on/off.

int **GetPageOrientation** ()

void **SetShowPageBoundary** (bool set)
set showing of page boundary.

void **ShowInfo** (int type)

void **SetPageSize** (PageSize::Type ps)
set pagesize to num.

PageSize::Type
GetPageSize () *return page-size (A4 etc).*

void **GetPageSize** (double &width, double &height)
return page size (pixels).

void **GetSizePages** (Point &topLeft, Point &bottomRight)
get area of used pages.

void **SetBannerPage** (int num)
set bannerpage to num.

int **GetBannerPage** ()
return page-size (DEF_BANNER etc).

bool **SavePostScript** (const string *file, bool TcmBanner)
actually save as postscript to the file.

bool **SaveEPS** (const string *file)
actually save as encapsulated postscript to the file.

bool **SaveFig** (const string *file, bool latexfonts=False)
actually save as Fig format to the file.

bool **SavePNG** (const string *file)

		actually save as PNG format to the file.
void	**DoSaveAsPS** (const string *file)	
void	**DoSaveAsEPS** (const string *file)	
void	**DoSaveAsFigTex** (const string *file)	
void	**DoSaveAsFigPS** (const string *file)	
void	**DoSaveAsPNG** (const string *file)	
void	**UpdateFSFiles** ()	
void	**SetPreviewCommand** (const string *newCommand)	
void	**SetDefaultFont** (int f, int s, int p)	
void	**SetPrintColors** (bool b)	
void	**SetDuplex** (bool b)	
void	**SetTumble** (bool b)	
void	**Draw** ()	*draw page boundaries, page numbers, header and footer.*
void	**Undraw** ()	
FileSelectionDialog*	**GetFileSelectionDialog** ()	
PromptDialog*	**GetPromptDialog** ()	
ScaleDialog*	**GetScaleDialog** ()	
Config*	**GetConfig** ()	
void	**SetStatus** (const string *msg)	
void	**SetStatus** (const char *msg)	
void	**SetViewer** (Viewer *v)	
void	**Save** (OutputFile *f)	*Save page attributes to file.*
bool	**Load** (InputFile *f, double format)	*Load page attributes from file.*
bool	**Check** (InputFile *f, double format)	

Check but not load page attributes from file.

Private Members

EditWindow*	**mainwindow**	
Viewer*	**viewer**	
Document*	**document**	
Config*	**config**	
PrinterOptions*	**printerOptions**	
PrinterQueueDialog*	**printerQueue**	
FileSelectionDialog*	**fileSelectionDialog**	
PromptDialog*	**promptDialog**	
ScaleDialog*	**scaleDialog**	
PageSize::Type	**pageSize**	*A4, A3, Letter etc.*
int	**pageOrientation**	*Portrait, landscape etc.*
int	**bannerPage**	*Default, None or TCM_Banner*
string	**previewCommand**	*Unix command to preview postscript file.*
string	**fig2devCommand**	*Unix command to generate diverse graphics files.*
string	**psFilterCommand**	*Unix command to filter postscript file.*
string	**docName**	*the name of the document that is going to be printed/saved.*
bool	**pageNumbering**	*page numbering on/off*

bool	**showPageBoundary**	*is page boundary visible.*
static const double	**PS_FACTOR**	*factor of screen pixels and postscript dots.*
bool	**header**	*show document header above each page.*
bool	**footer**	*show document footer beneath each page.*
bool	**printColors**	*print colors in PostScript or b/w only.*
bool	**duplex**	*print each page on a distinct sheet.*
bool	**tumble**	*print each backside page flipped relative to front side page.*
void	**DoSaveAs** (const string *file, OutputType type)	
void	**MakePSFilter** (string *psfilter)	
void	**GetPages** (int &nX, int &nY)	*count nr of pages used in x (nX) and y (nY) direction.*
void	**DrawPageBoundary** ()	*draw lines to reflect page boundaries.*
void	**DrawPageNumbering** ()	*draw all page numbers.*
void	**DrawPageNumber** (int i, int j, int nX, int nY, int w, int h)	*draw page nr at page (i, j) of size (w, h) (nX, nY pages total).*
void	**DrawHeaders** ()	*draw document info in page headers.*
void	**DrawHeader** (int i, int j, int w, int h)	*draw document header on page (i, j) of size (w, h).*
void	**DrawFooters** ()	*draw document info in page footers.*
void	**DrawFooter** (int i, int j, int w, int h)	

draw document info on page (i, j) of size (w, h).

void **DrawInfo** (int x, int y, int wd)
draw document info strings from position (x, y), info width is wd.

XFont* **GetFont** (int family, int style, int size)
change font of page numbers and headers to this font.

void **MakeTmpFile** (string *tmp)
put in tmp name of temperorary file.

int **ScaleCorrect** (int x)

XFont* **defaultFont** *font for page numbers and headers.*

List <XFont *> *
fontList *save fonts for later reuse.*

static const int
INFO_LINES *nr. of lines of text in header/footer info.*

252

class **PrinterOptions**

Printer options class.

Public Members

	PrinterOptions ()
void	**SetPrinterName** (const string* newName)
const string*	**GetPrinterName** ()
void	**SetPrintCopies** (unsigned i)
unsigned	**GetPrintCopies** ()
void	**SetPrintBannerPage** (int i)
unsigned	**GetPrintBannerPage** ()
void	**SetPrintCommand** (const string* newCommand)
const string*	**GetPrintCommand** ()
void	**SetTmpFile** (const string* s)
const string*	**GetTmpFile** ()
const string*	**GetTextFilterCommand** ()
void	**SetTextFilterCommand** (const string* newCommand)
static const int	**MIN_PRINT_COPIES**
static const int	**MAX_PRINT_COPIES**

Private Members

string	**printerName**	*name of current printer.*
unsigned	**printCopies**	*number of copies that are made.*
int	**printBannerPage**	

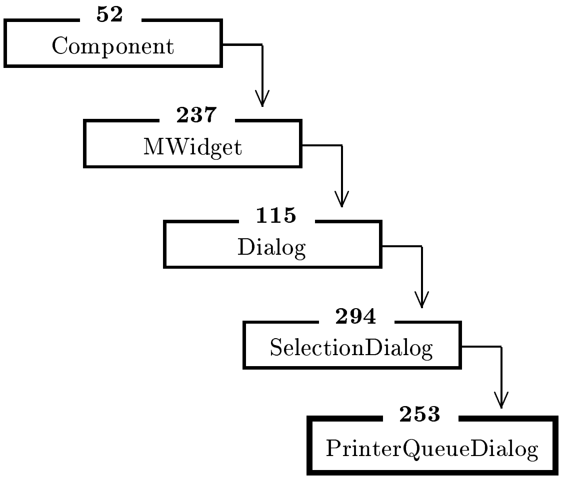
		printer banner page
string	**printCommand**	*Unix command to be used to print document.*
string	**textFilterCommand**	
string	**tmpFile**	

253

class **PrinterQueueDialog** : public SelectionDialog

class for showing printer queue.

Inheritance



Public Members

	PrinterQueueDialog (Widget parent, MainWindow *)
const char*	**GetClassName** ()
void	**Update** ()
void	**Remove** (const char *str)
void	**SetPrinterName** (const string *s)
void	**PrinterQueueCommand** () *show popup dialog to set Unix printer queue command.*
void	**PrinterRemoveCommand** () *... printer remove command.*
void	**SetPrinterQueueCommand** (const string *c)

void **SetPrinterRemoveCommand** (const string *c)

void **SetDefaultPrinterQueueCommand** (const string *c)

void **SetDefaultPrinterRemoveCommand** (const string *c)

PromptDialog* **GetPromptDialog** ()

Protected Members

void **CreateWidget** ()

void **Configure** ()

static void **RemoveCB** (Widget, XtPointer, XtPointer)

static void **UpdateCB** (Widget, XtPointer, XtPointer)

static void **SetPrinterQueueCommandCB** (Widget, XtPointer, XtPointer)

static void **SetPrinterRemoveCommandCB** (Widget, XtPointer, XtPointer)

Private Members

string **queueCommand**

string **removeCommand**

string **defaultQueueCommand**

string **defaultRemoveCommand**

MainWindow* **mainwindow**

PrinterOptions* **printerOptions**

PromptDialog*

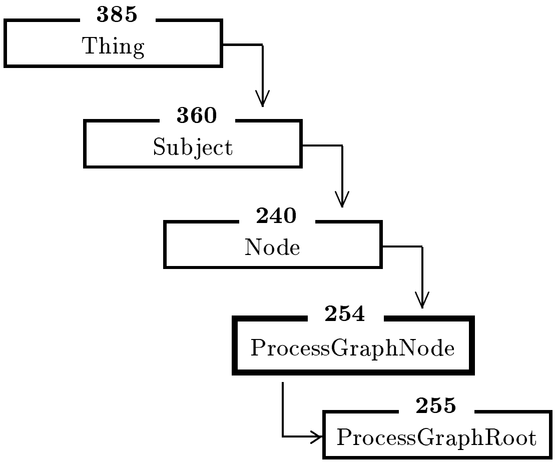
promptDialog

254

class **ProcessGraphNode** : public Node

process graph node class.

Inheritance



Public Members

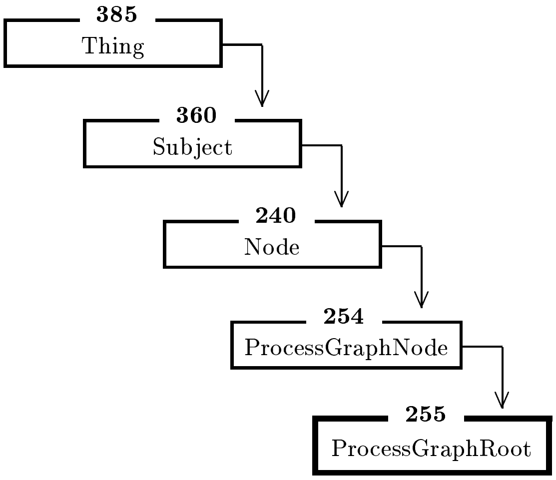
	ProcessGraphNode (RPGraph *)
int	**GetClassType** () const
Subject*	**Clone** ()
Subject::NameErrType	**SetName** (const string *name)

255

class **ProcessGraphRoot** : public ProcessGraphNode

process graph root node class.

Inheritance



Public Members

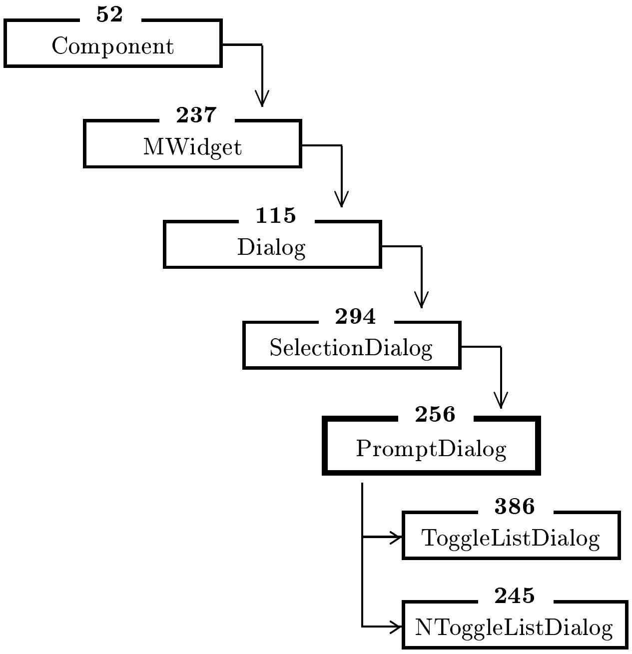
	ProcessGraphRoot (RPGraph *g)
Subject*	**Clone** ()
int	**GetClassType** () const
Subject::NameErrType	**SetName** (const string *n)
bool	**ReadMembers** (InputFile *f, double format)

256

class **PromptDialog** : public SelectionDialog

prompt dialog class.

Inheritance



Public Members

PromptDialog (Widget parent)

const char*
GetClassName ()

Protected Members

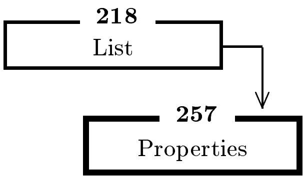
void **CreateWidget** ()

void **Configure** ()

257

class **Properties** : public List<Property>

Inheritance



Public Members

void **Save** (OutputFile* f)

void **Save** (OutputFile* f, const string* header)

void **Load** (InputFile* f)

const string*
GetValue (const string* key)

const string*
GetValue (const char* key)

258

class **Property**

A key-value pair.

Public Members

string **key**

string **value**

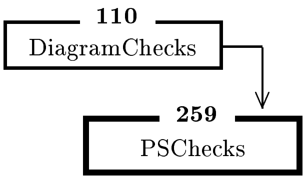
friend bool **operator==** (const Property &x, const Property &y)

259

class **PSChecks** : public DiagramChecks

implements generic checks for soft constraints on PS diagrams.

Inheritance



Public Members

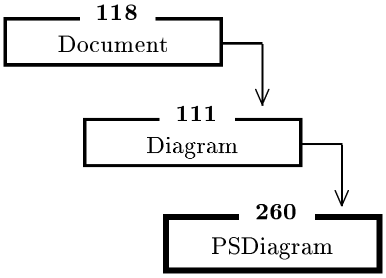
	PSChecks (Diagram *d, Graph *g)
unsigned	**CheckRootOperator** (PSProcess *root, string &chkbuf)
unsigned	**CheckChildOperators** (PSProcess *parent, List<Subject *> *children, string &chkbuf)
unsigned	**CheckChildPositions** (DiagramViewer *dviewer, PSProcess *parent, List<Subject *> *children, string &chkbuf)
unsigned	**CheckDoubleProcessNames** (string &chkbuf)

260

class **PSDiagram** : public Diagram

process structure diagram class.

Inheritance



Public Members

PSDiagram (Config *, PSWindow *, PSViewer *, PSGraph *)

virtual **~PSDiagram** ()

Node* **CreateNode** ()

Edge* **CreateEdge** (Subject *n1, Subject *n2)

NodeShape* **CreateNodeShape** (Node *node, int x, int y)

Line* **CreateLine** (Edge *edge, GShape *fromShape, GShape *toShape, List<Point *> *line)

void **UpdateNodeType** (int n)

void **UpdateEdgeType** (int n)

bool **SetOperator** (PSProcess *proc, const string *op)
set operator of process. Check if that is OK.

unsigned **TraverseTree** (bool c)

traverses nodes and checks all operators. Assign sequence numbers.

void **CheckDocument** ()

bool **SetText** (TextShape *t, const string *s)

Protected Members

bool **CheckEdgeConstraints** (Subject *s1, Subject *s2)

Thing* **CreateThing** (int classNr)

void **SaveEntries** ()

unsigned **TraverseChildren** (PSProcess *parent, List<Subject *> *children, bool c)

unsigned **FindRoot** (List<Subject *> *nodes, PSProcess **root)

void **FindChildren** (PSProcess *parent, List<Subject *> *nodes, List<Subject *> *children)

Edge* **FindLeftMostEdge** (PSProcess *parent, List<Subject *> *childEdges)

Private Members

PSChecks* **psChecks**

unsigned **treeSequenceNr**

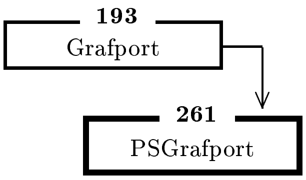
unsigned **actionSequenceNr**

261

class **PSGrafport** : public Grafport

PostScript(tm) grafport class.

Inheritance



Public Members

	PSGrafport (const char *fileName) *open grafport to filename, When filename = "", stdout is used.*
virtual	**~PSGrafport** ()
bool	**IsSucces** ()
void	**ClearArea** (int, int, int, int)
void	**SetLineWidth** (unsigned n)
void	**SetBackgroundColor** (const string *)
void	**SetBackgroundColor** (const char *)
void	**SetForegroundColor** (const string *)
void	**SetForegroundColor** (const char *)
void	**DrawRectangle** (int x, int y, int wd, int ht)
void	**FillRectangle** (int x, int y, int wd, int ht)
void	**DrawRoundedRectangle** (int x, int y, int wd, int ht, int rd)
void	**FillRoundedRectangle** (int x, int y, int wd, int ht, int rd)
void	**DrawEllipsedRectangle** (int x, int y, int wd, int ht, int rd)

void	**FillEllipsedRectangle** (int x, int y, int wd, int ht, int rd)
void	**DrawDisk** (int x, int y, int wd, int ht, int ellipse_ht)
void	**FillDisk** (int x, int y, int wd, int ht, int ellipse_ht)
void	**DrawUnzoomedRectangle** (int x, int y, int wd, int ht)
void	**DrawStringLeft** (int x, int y, const char *str)
void	**DrawStringCentered** (int x, int y, const char *str)
void	**DrawStringRight** (int x, int y, const char *str)
void	**DrawUnzoomedStringLeft** (int x, int y, const char *str)
void	**DrawUnzoomedStringCentered** (int x, int y, const char *str)
void	**DrawStringLeftUnderlined** (int x, int y, const char *str)
void	**DrawStringRightUnderlined** (int x, int y, const char *str)
void	**DrawStringCenteredUnderlined** (int x, int y, const char *str)
void	**DrawLine** (int x1, int y1, int x2, int y2)
void	**DrawLine** (const Point *p1, const Point *p2)
void	**DrawCurve** (const Point *p1, const Point *p2, const Point *p3, const Point *p4)
void	**DrawUnzoomedLine** (int x1, int y1, int x2, int y2)
void	**DrawPoint** (int x, int y)
void	**DrawPoint** (const Point *p)

void	**DrawEllipse** (int x, int y, int wd, int ht)
void	**FillEllipse** (int x, int y, int wd, int ht)
void	**DrawArc** (int x, int y, int wd, int ht, int arc1, int arc2)
void	**FillArc** (int x, int y, int wd, int ht, int arc1, int arc2)
void	**DrawPolygon** (const Point *points, int n)
void	**FillPolygon** (const Point *points, int n)
void	**Header** (const char *title, const char *creator) *generate postscript header.*
void	**BeginDrawProc** ()
void	**EndDrawProc** ()
void	**CallDrawProc** ()
void	**Banner** (const char *title) *concat banner from file with name of document.*
void	**BeginPage** (int i, int j, int nx, int tot)
void	**EndPage** ()
void	**Trailer** () *generate postscript trailer.*
void	**SetFont** (XFont *font) *set grafport font.*
void	**EPSHeader** (const char *title, const char *creator, int llx, int lly, int urx, int ury) *header for encapsulated postscript + bounding box*
void	**EPSProlog** (int y, int dy)
void	**SetPageSize** (double x, double y)
void	**SetFactor** (double f)
void	**SetISOLatin1Encoding** (bool b)
void	**SetShowColors** (bool b)
static const char	**BANNER** [] *postscript file containing front page for printer.*
static const double	

XOFF

static const double
YOFF

Protected Members

static void
MakePSString (string *str)

static void
FixEscapeChar (string *str, char c)

static void
FixNonAscii (string *str)

void **FontProlog** ()

void **InstallIsoLatin1Font** (const char *psFont)

void **PageSetup** ()

void **DrawPSRectangle** (int x, int y, int wd, int ht)

void **DrawPSRoundedRectangle** (int x, int y, int wd, int ht, int rd)

void **DrawPSEllipsedRectangle** (int x, int y, int wd, int ht, int rd)

void **DrawPSArc** (int x, int y, int wd, int ht, int ang1, int ang2)

void **DrawPSEllipse** (int x, int y, int wd, int ht)

void **DrawPSDisk** (int x, int y, int wd, int ht, int ellipse_ht)

void **DrawPSLine** (int x1, int y1, int x2, int y2)

void **DrawPSPolygon** (const Point *point, int nrPoints)

Private Members

FILE*	**fd**	*postscript output file.*
double	**width**	
double	**height**	
double	**factor**	

bool **succes**

bool **isoLatin1Encoding**

bool **showColors**

static const double
DEFAULT_LINE_WIDTH
default PostScript line width.

List <string>
definedFonts

void **SetPSDashes** ()

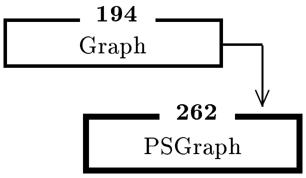
void **UnsetPSDashes** ()

262

class **PSGraph** : public Graph

process structure graph class.

Inheritance



Public Members

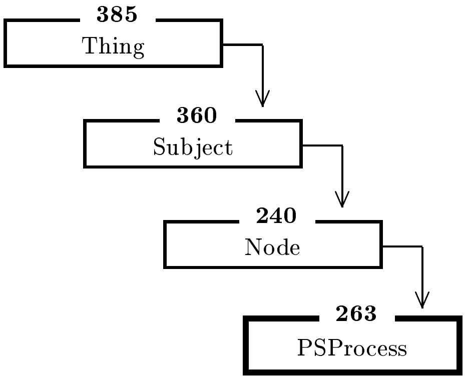
	PSGraph ()
void	**InitConnections** ()

263

class **PSProcess** : public Node

Process structure diagram process node class.

Inheritance



Public Members

	PSProcess (PSGraph *)
Subject*	**Clone** ()
int	**GetClassType** () const
char	**GetOperator** ()
bool	**SetOperator** (char x)
bool	**IsRoot** ()
void	**SetRoot** (bool set)
bool	**IsAction** ()
void	**SetAction** (bool set)
void	**SetTreeSequence** (int i)
int	**GetTreeSequence** ()
void	**SetActionSequence** (int i)
int	**GetActionSequence** ()

Subject::NameErrType

	SetName (const string *name)
void	**WriteMembers** (OutputFile *f)
bool	**ReadMembers** (InputFile *f, double format)

Private Members

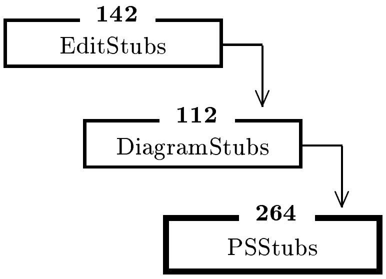
char	**cOp**
bool	**isRoot**
bool	**isAction**
int	**treeSequence**
int	**actionSequence**

264

class **PSStubs** : public DiagramStubs

Specific stubs for TPSD.

Inheritance



Public Members

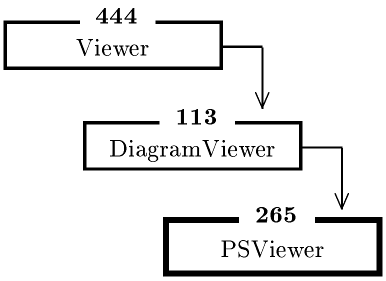
static void
ShowSequenceCB (Widget, XtPointer, XtPointer)

265

class **PSViewer** : public DiagramViewer

process structure viewer class.

Inheritance



Public Members

	PSViewer (Config *c, PSWindow *w)	
void	**SetShowSequences** (int i)	*shows Sequence of all processes in tree. 0=none, 1=actions, 2=tree.*
void	**UpdateShowSequences** ()	

Protected Members

void	**UpdateSequenceLabels** (int set)	

Private Members

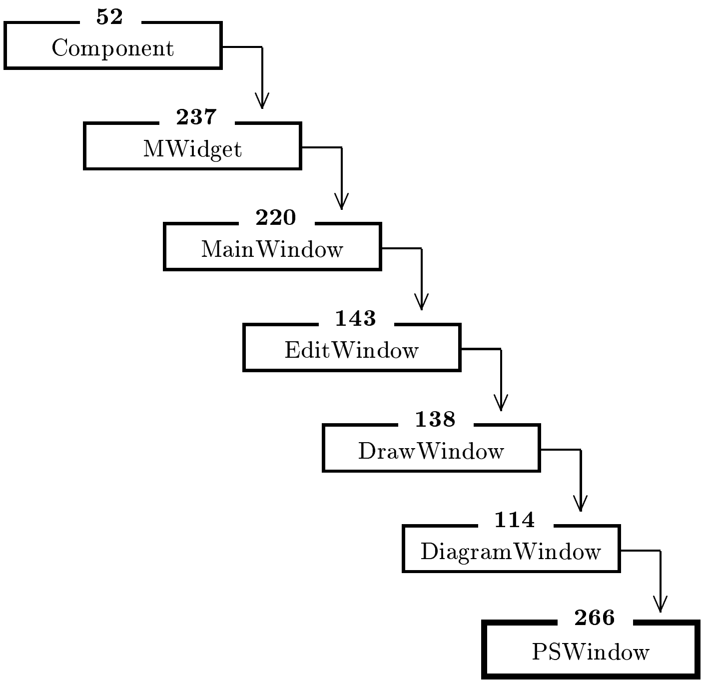
int	**showSequence**	*current type of show sequence.*

266

class **PSWindow** : public DiagramWindow

process structure edit window class.

Inheritance



Public Members

	PSWindow (const char *name)
virtual	**~PSWindow** ()

Private Members

int	**GetTool** ()
DiagramViewer*	**CreateViewer** (Config *config)
Graph*	**CreateGraph** ()

Diagram* **CreateDiagram** (Config *c, DiagramViewer *v, Graph *g)

void **DetermineIcons** ()

Widget **CreateMenuBar** (Widget parent)

MenuItem* **diagramPSShowSequenceItems**

void **InitMenuItems** ()

static const char* **PS_NODE_NAMES** []

static const Bitmap **PS_NODE_ICONS** []

static const char* **PS_EDGE_NAMES** []

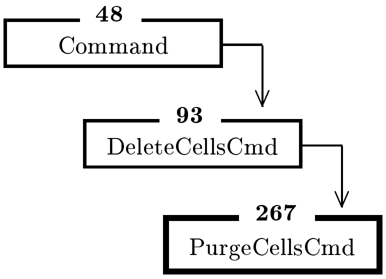
static const Bitmap **PS_EDGE_ICONS** []

267

class **PurgeCellsCmd** : public DeleteCellsCmd

purge empty rows and columns command.

Inheritance



Public Members

	PurgeCellsCmd (Table *t, TableViewer *v)
void	**Execute** ()
void	**UnExecute** ()
const char*	**GetName** () const

Private Members

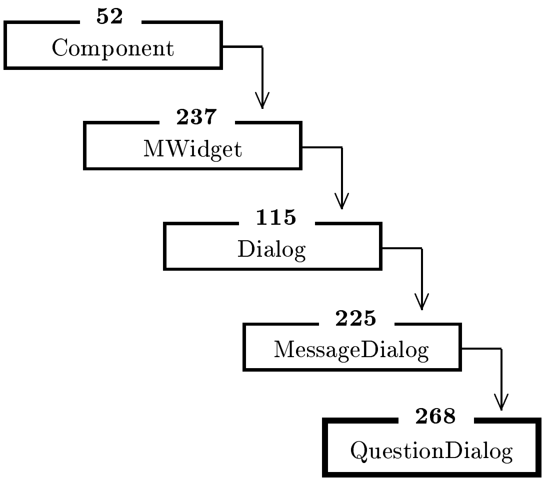
void	**DetermineRows** ()
void	**DetermineColumns** ()

268

class **QuestionDialog** : public MessageDialog

question dialog class.

Inheritance



Public Members

enum	**AnswerType**	*answers to no/yes/cancel dialog questions.*
	QuestionDialog (Widget parent, bool cancel)	
const char*	**GetClassName** ()	
void	**Popup** ()	*do not use.*
void	**Show** (const char *, const char *)	*do not use.*
void	**Show** (const char *, const string *)	*do not use.*
AnswerType	**GetAnswer** ()	
void	**SetAnswer** (AnswerType a)	

Protected Members

void **CreateWidget** ()

void **Configure** ()

static void
ResponseCB (Widget, XtPointer answer, XtPointer cbs)

Private Members

bool **withCancel**

AnswerType
answer

269

class **ReadDirection**

different read directions

Public Members

enum **Type**

static Type
String2Type (const string* s)
e.g. convert "FromShape" to FROMSHAPE.

static void
Type2String (Type t, string* s)
e.g. convert FROMSHAPE to "FromShape".

270

class **Rectangle**

rectangle ADT, defined by x, y, width and height.

Public Members

static const Rectangle
ZERO

Rectangle (int a = 0, int b = 0, int w = 0, int h=0)
Construct and initialize new rectangle.

Rectangle (int w, int h)
Construct and initialize new rectangle.

Rectangle (int d) *Construct and initialize new rectangle.*

Rectangle (Point origin, Point extent)
Construct and initialize new rectangle.

void **Set** (int a, int b, int w, int h)
Reshape rectangle.

void **SetPosition** (int a, int b)
Set rectangle position.

void **Move** (int a, int b)
Move rectangle position.

bool **Inside** (Point p) const
Return if point (a, b) is inside rectangle.

bool **Inside** (int a, int b) const
Return if point (a, b) is inside rectangle.

bool **Intersects** (Rectangle* r) const
Return if rectangle r intersects with 'this'.

friend bool

operator== (const Rectangle &x, const Rectangle &y)
Return whether rectangles are equal.

friend bool
operator!= (const Rectangle &x, const Rectangle &y)
Return whether rectangles are unequal.

friend ostream&
operator<< (ostream &s, const Rectangle r)
Write string representation to ostream.

friend istream&
operator>> (istream &s, Rectangle &r)
Read string representation from istream.

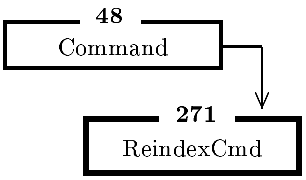
int	**x**	*the left x-coordinate of the rectangle*
int	**y**	*the top y coordinate of the rectangle*
int	**width**	*the width of the rectangle*
int	**height**	*the height of the rectangle*

271

class **ReindexCmd** : public Command

command for re-indexing nodes.

Inheritance



Public Members

	ReindexCmd (ShapeView *v, List<Node *> *nodes)
virtual	**~ReindexCmd** ()
void	**Execute** ()
void	**UnExecute** ()
const char*	**GetName** () const

Private Members

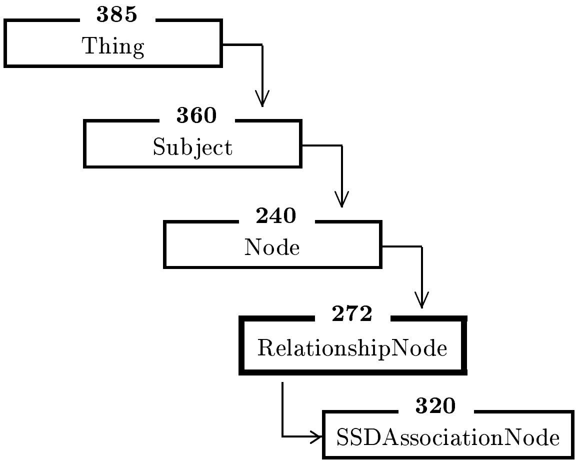
List <Node *>	**nodes**
List <string *>	**oldIndexes**
string	**parent**
void	**SetIndex** (Node *node, const string *index)

272

class **RelationshipNode** : public Node

relationship (ER) node class.

Inheritance



Public Members

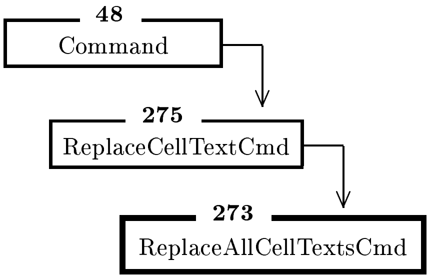
	RelationshipNode (ERGraph *g)
Subject*	**Clone** ()
int	**GetClassType** () const
Subject::NameErrType	**SetName** (const string *name)
virtual void	**GetComponents** (List<string *> *components)
friend int	**CompareComponent** (string *c1, string *c2)

273

class **ReplaceAllCellTextsCmd** : public ReplaceCellTextCmd

command class for replacing text of all cells.

Inheritance



Public Members

ReplaceAllCellTextsCmd (Table *,
TableViewer *v,
const string *s1,
const string *s2,
bool sens,
bool substr)

void **Execute** ()

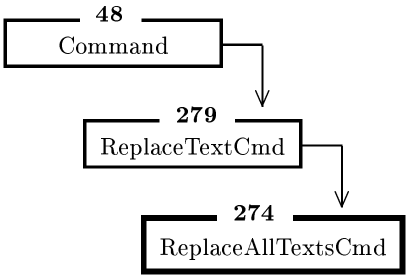
const char*
GetName () const

274

class **ReplaceAllTextsCmd** : public ReplaceTextCmd

command class for replacing text of all text shapes.

Inheritance



Public Members

ReplaceAllTextsCmd (ShapeView *v,
const string *s1,
const string *s2, bool sens,
bool substr,
bool nameOnly)

void **Execute** ()

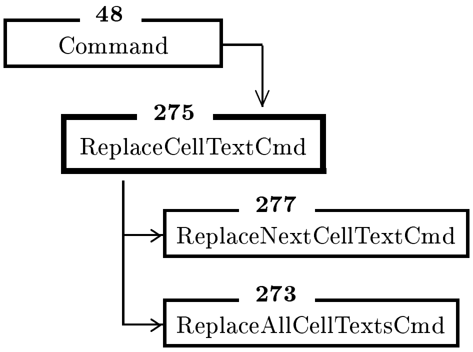
const char*
GetName () const

275

class **ReplaceCellTextCmd** : public Command

abstract command class for replacing text of cells.

Inheritance



Public Members

	ReplaceCellTextCmd (Table *, TableViewer *v, const string *s1, const string *s2, bool sens, bool substr)
virtual	**~ReplaceCellTextCmd** ()
void	**UnExecute** ()
void	**ReExecute** ()

Protected Members

TableViewer*	**GetTableViewer** ()
string*	**GetPattern** ()
string*	**GetReplace** ()
bool	**IsCaseSensitive** ()
bool	**IsSubstring** ()

List <Cell *> *
GetCells ()

List <string *> *
GetOldStrings ()

Private Members

TableViewer*
tviewer

List <Cell *> *
oldSelection

List <Cell *> *
cells

List <string *> *
oldStrings

string* **pattern**

string* **replace**

bool **caseSensitive**

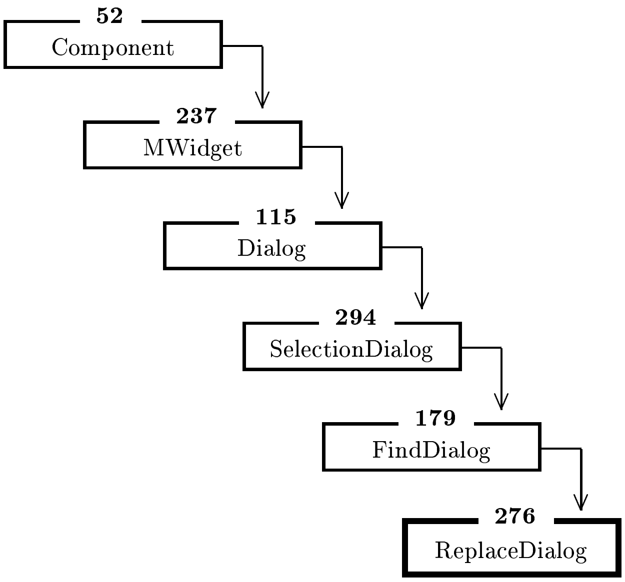
bool **substring**

276

class **ReplaceDialog** : public FindDialog

replace dialog class.

Inheritance



Public Members

	ReplaceDialog (Widget parent)
const char*	**GetClassName** ()
void	**SetReplaceText** (const char *)
void	**SetReplaceText** (const string *)
void	**GetReplaceText** (char *)
void	**GetReplaceText** (string *)
void	**SetReplaceCallback** (XtCallbackProc fun, XtPointer clientData)
static void	**ReplaceClearCB** (Widget, XtPointer, XtPointer)

Protected Members

void	**CreateWidget** ()
void	**Configure** ()
void	**AttachToggles** ()

Private Members

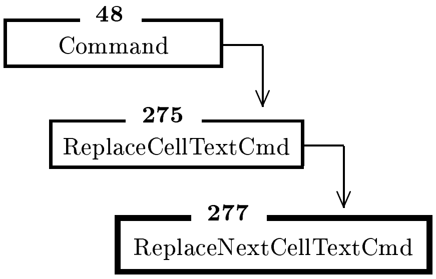
Widget	**replaceLabel**
Widget	**replaceText**
Widget	**replaceButton**

277

class **ReplaceNextCellTextCmd** : public ReplaceCell-TextCmd

command class for replacing next cells' text.

Inheritance



Public Members

ReplaceNextCellTextCmd (Table *,
TableViewer *v,
const string *s1,
const string *s2,
bool sens,
bool substr)

void **Execute** ()

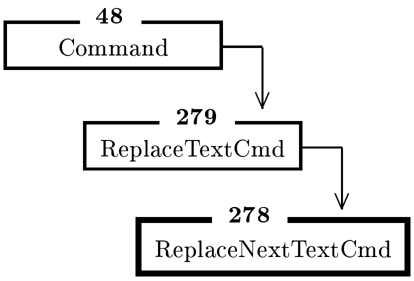
const char*
GetName () const

278

class **ReplaceNextTextCmd** : public ReplaceTextCmd

command class for replacing next shapes' text.

Inheritance



Public Members

ReplaceNextTextCmd (ShapeView *v,
const string *s1,
const string *s2,
bool sens, bool substr,
bool nameOnly)

void **Execute** ()

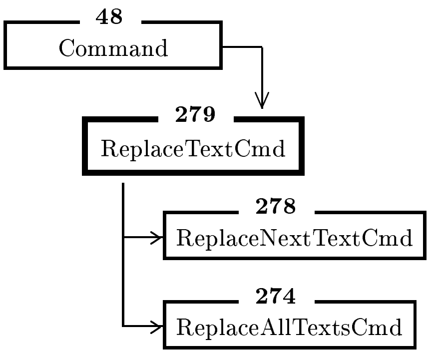
const char*
GetName () const

279

class **ReplaceTextCmd** : public Command

abstract command class for replacing text of all text shapes.

Inheritance



Public Members

ReplaceTextCmd (ShapeView *v, const string *s1, const string *s2, bool sens, bool substr, bool nameOnly)

virtual **~ReplaceTextCmd** ()

void **UnExecute** ()

void **ReExecute** ()

Protected Members

Diagram* **GetDiagram** ()

ShapeView* **view**

Diagram* **diagram**

List <GShape *> * **oldSelection**

List <TextShape *> *

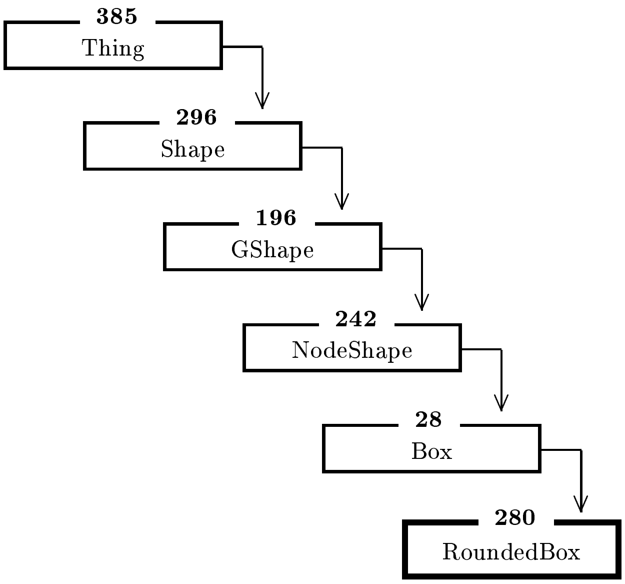
	textShapes
List <string *> *	**oldStrings**
List <Point *> *	**oldSizes**
string*	**pattern**
string*	**replace**
bool	**caseSensitive**
bool	**substring**
bool	**nameOnly**

280

class **RoundedBox** : public Box

box drawn as a rounded rectangle.

Inheritance



Public Members

	RoundedBox (ShapeView *v, Grafport *g, int x, int y, int w = BOX_WIDTH, int h = BOX_HEIGHT)
	RoundedBox (const NodeShape &c)
Shape*	**Clone** ()
int	**GetClassType** () const
void	**DrawOutLine** (const Point *center)
Point	**GiveSnp** (Point to) const
Point	**GiveOrthoSnp** (Point to) const

Point **GiveSnp** (Point p, int order, int count, int &code)

Protected Members

void **DrawShape** () *draws a rounded rectangle.*

int **GetRadius** () const *return radius of rounded corner.*

Private Members

static const int **RD_FACTOR** *radius factor of the rounded corner.*

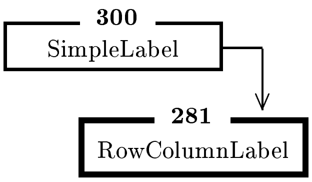
Point **GiveRectSnp** (Point p) const

281

class **RowColumnLabel** : public SimpleLabel

row/column label class.

Inheritance



Public Members

RowColumnLabel (Grafport *g,
CellVector *vector)

virtual void
Draw ()

CellVector*
GetVector ()

Private Members

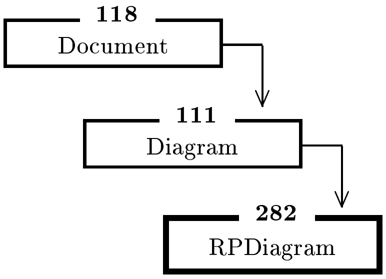
CellVector*
vector

282

class **RPDiagram** : public Diagram

recursive process graph class

Inheritance



Public Members

	RPDiagram (Config *, RPWindow *, RPViewer *, RPGraph *)
virtual	**~RPDiagram** ()
Node*	**CreateNode** ()
Edge*	**CreateEdge** (Subject *s1, Subject *s2)
NodeShape*	**CreateNodeShape** (Node *node, int x, int y)
Line*	**CreateLine** (Edge *edge, GShape *fromShape, GShape *toShape, List<Point *> *line)
void	**UpdateNodeType** (int n)
void	**UpdateEdgeType** (int n)
void	**CheckDocument** ()
static const int	**BOX_WIDTH**
static const int	**BOX_HEIGHT**
static const int	

CIRCLE_WIDTH

Protected Members

Thing* **CreateThing** (int classNr)

void **PlaceShapes** ()

bool **CheckEdgeConstraints** (Subject *s1, Subject *s2)

ProcessGraphRoot*
FindRoot (Subject *subj)

Private Members

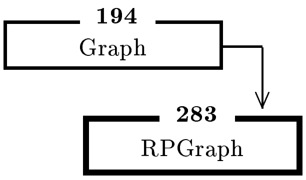
DiagramChecks*
checks

283

class **RPGraph** : public Graph

recursive process graph class.

Inheritance



Public Members

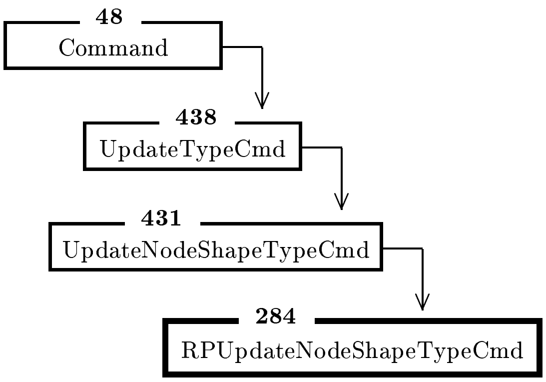
	RPGraph ()
void	**InitConnections** ()

284

class **RPUpdateNodeShapeTypeCmd** : public UpdateNodeShapeTypeCmd

command for changing process graph node shape types

Inheritance



Public Members

RPUpdateNodeShapeTypeCmd (ShapeView *v, List<int *> *fromTypes, int toType)

const char* **GetName** () const

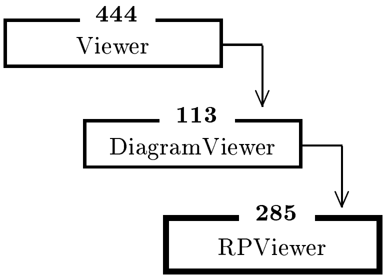
Thing* **Transform** (Thing *thing)

285

class **RPViewer** : public DiagramViewer

recursive process graph viewer class.

Inheritance



Public Members

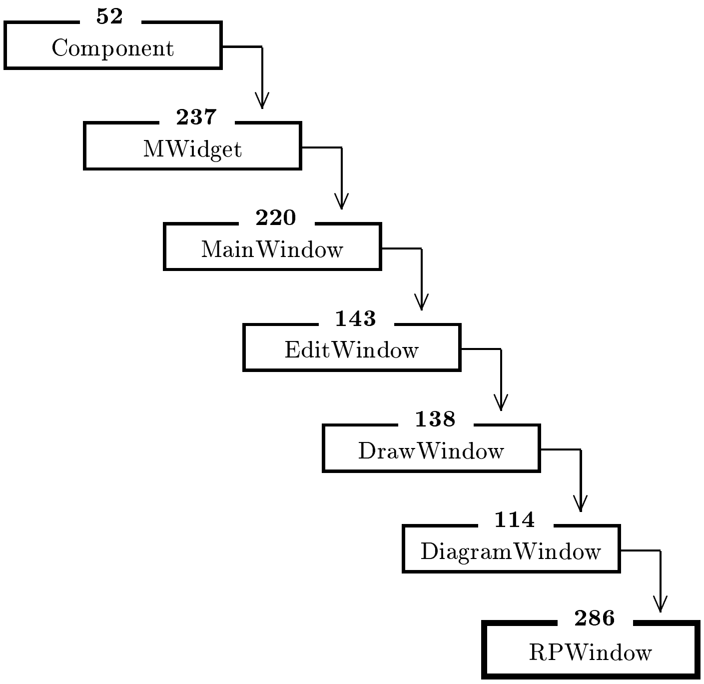
	RPViewer (Config *c, RPWindow *w)	
void	**UpdateNodeShapeType** (int)	*change node shape type of selected types.*
void	**UnlabelNodeShapes** ()	*make the labels of mini-ellipses empty.*

286

class **RPWindow** : public DiagramWindow

recursive process graph edit window class.

Inheritance



Public Members

RPWindow (const char *name)

Private Members

int **GetTool** ()

DiagramViewer*
CreateViewer (Config *config)

Graph* **CreateGraph** ()

Diagram* **CreateDiagram** (Config *c, DiagramViewer *v, Graph *g)

void **DetermineIcons** ()

static const char* **RP_NODE_NAMES** []

static const Bitmap **RP_NODE_ICONS** []

static const char* **RP_EDGE_NAMES** []

static const Bitmap **RP_EDGE_ICONS** []

void **InitMenuItems** ()

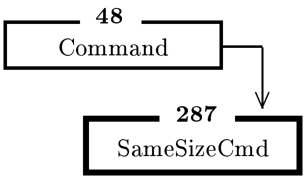
MenuItem* **convertItems**

287

class **SameSizeCmd** : public Command

command class for setting selected node shapes to the same size.

Inheritance



Public Members

	SameSizeCmd (ShapeView *v)
virtual	**~SameSizeCmd** ()
void	**Execute** ()
void	**UnExecute** ()
const char*	**GetName** () const

Private Members

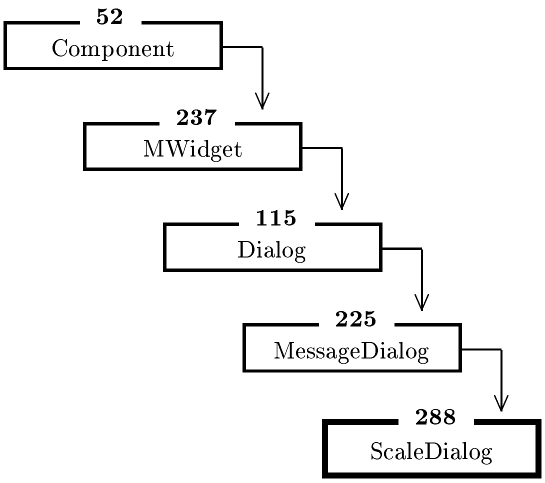
Diagram*	**diagram**
List <GShape *> *	**shapes**
List <Point *> *	**oldSizes**

288

class **ScaleDialog** : public MessageDialog

scale dialog class.

Inheritance



Public Members

	ScaleDialog (Widget parent)
const char*	**GetClassName** ()
void	**SetScaleValues** (int min, int max, int init, int decimal, int def)
void	**SetScaleLabel** (const char *text)
void	**SetValueChangedCallback** (XtCallbackProc fun, XtPointer clientData)
void	**SetScaleValue** (int val)
int	**GetScaleValue** ()
int	**GetDefaultValue** ()
int	**GetInitValue** ()

Protected Members

void	**CreateWidget** ()
void	**Configure** ()
static void	**SetDefaultCB** (Widget, XtPointer cd, XtPointer)
static void	**SetInitCB** (Widget, XtPointer cd, XtPointer)

Private Members

Widget	**scale**
int	**defaultValue**
int	**initValue**
XtCallbackProc	**valueChangeFun**
XtPointer	**clientData**

289

class **Scaler**

class for scaling coordinates

Public Members

	Scaler (Config *c, Viewer *v, DrawWindow *d)	
virtual	**~Scaler** ()	
void	**MakeLarger** ()	*Make larger. Change scale value.*
void	**MakeSmaller** ()	*Make smaller. Change scale value.*
void	**ScaleFactor** ()	*Popup scale factor dialog.*
void	**NormalScale** ()	*Return to normal scaling.*
void	**WholeDrawing** ()	*Calculate scale value such that whole document fits to one page.*
void	**ScreenFit** ()	
int	**Scale** (int x) const	*scale coordinate x.*
Point	**Scale** (const Point *p) const	
int	**ScaleCorrect** (int x) const	*Correct X-window coordinate to the right scaled coordinate.*
Point	**ScaleCorrect** (const Point *p) const	
void	**SetScaleValue** (double newValue)	*Really do the scaling. Redraws the contents of the viewer.*
double	**GetScaleValue** () const	
void	**SetScaleFactor** (double d)	
DrawWindow*	**GetMainWindow** () const	
ScaleDialog*	**GetScaleDialog** () const	
void	**Save** (OutputFile *f)	

Save scale value to file.

bool **Load** (InputFile *f, double format)
Load scale value from file.

bool **Check** (InputFile *f, double format)
Check (but not load) scale value from file.

Private Members

Viewer* **viewer**

DrawWindow* **mainwindow**

ScaleDialog* **scaleDialog**

double **scaleFactor** *current value to multiply coordinates with 1 = no scaling.*

double **scaleValue**

double **defaultScaleFactor**

static const double **MIN_SCALE_VALUE**

static const double **MAX_SCALE_VALUE**

static const double **MIN_SCALE_FACTOR**

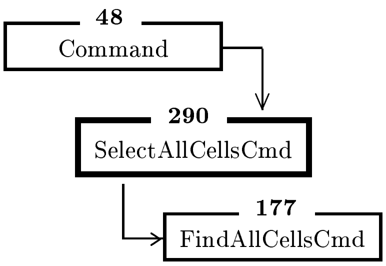
static const double **MAX_SCALE_FACTOR**

290

class **SelectAllCellsCmd** : public Command

select all cells in table command.

Inheritance



Public Members

	SelectAllCellsCmd (Table *t, TableViewer *v)
virtual	**~SelectAllCellsCmd** ()
void	**Execute** ()
void	**UnExecute** ()
const char*	**GetName** () const

Protected Members

TableViewer*	**GetTableViewer** ()

Private Members

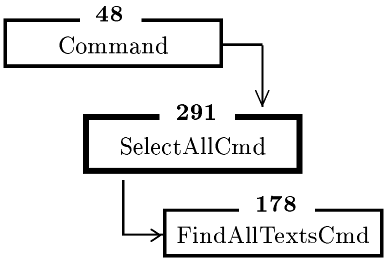
List <Cell *> *	**oldSelection**
TableViewer*	**tviewer**

291

class **SelectAllCmd** : public Command

command class for selecting all shapes in the view.

Inheritance



Public Members

	SelectAllCmd (ShapeView *v)
virtual	**~SelectAllCmd** ()
void	**Execute** ()
void	**UnExecute** ()
const char*	**GetName** () const

Protected Members

ShapeView*	**GetView** () const

Private Members

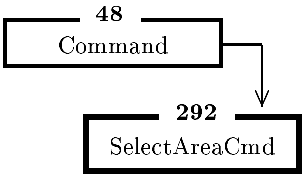
List <GShape *> *	**oldSelection**
ShapeView*	**view**

292

class **SelectAreaCmd** : public Command

command class for selecting a diagram area.

Inheritance



Public Members

	SelectAreaCmd (ShapeView *view)
virtual	**~SelectAreaCmd** ()
void	**TrackMouse** (TrackType aPhase, Point *anchorPoint, Point *previousPoint, Point *nextPoint)
void	**Execute** ()
void	**UnExecute** ()
const char*	**GetName** () const

Private Members

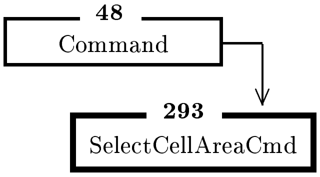
ShapeView*	**view**
Rectangle	**area**
List <GShape *> *	**oldSelection**
bool	**InArea** (Shape *s)

293

class **SelectCellAreaCmd** : public Command

select part of table command.

Inheritance



Public Members

	SelectCellAreaCmd (Table *t, TableViewer *v)
virtual	**~SelectCellAreaCmd** ()
void	**TrackMouse** (TrackType aPhase, Point *anchorPoint, Point *previousPoint, Point *nextPoint)
void	**Execute** ()
void	**UnExecute** ()
void	**ReExecute** ()
const char*	**GetName** () const

Private Members

List <Cell *> *
oldSelection

List <Cell *> *
newSelection

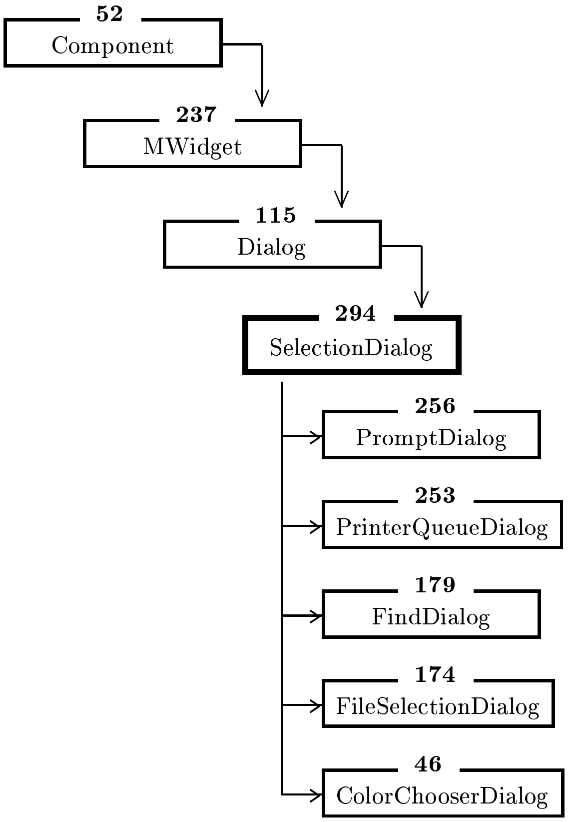
TableViewer*
tviewer

294

class **SelectionDialog** : public Dialog

selection dialog class.

Inheritance



Public Members

	SelectionDialog (Widget parent)
const char*	**GetClassName** ()
void	**SetTextString** (const char *v) *sets selection value*
void	**SetTextString** (const string *v)

void **SetEditable** (bool b)
make text field (un)editable

void **GetTextString** (char *n)
returns selection value.

void **GetTextString** (string *n)

void **ManageTextString** (bool)

void **SetSelectionLabel** (const char *l)

void **ManageSelectionLabel** (bool)

void **SetApplyCallback** (XtCallbackProc fun, XtPointer clientData)

void **SetApplyButtonLabel** (const char *txt)

void **ManageApplyButton** (bool)

static void
ClearCB (Widget, XtPointer, XtPointer)

static void
SetDefaultCB (Widget, XtPointer, XtPointer)

void **SetDefaultValue** (const char *s)

void **SetDefaultValue** (const string *s)

const string*
GetDefaultValue ()

void **SelectItem** (const string *label)

void **DeselectAllItems** ()

void **DeleteAllItems** ()

void **AddItem** (const char *text, int pos = 0)

void **AddItem** (const string *text, int pos = 0)

Protected Members

Widget **GetText** ()

void **SetText** (Widget w)

Widget **GetList** ()

Widget **GetSelectionLabel** ()

void **SetSelectionLabel** (Widget w)

void **CreateWidget** ()

void **Configure** ()

Private Members

string **defaultValue**

Widget **text**

Widget **selectionLabel**

Widget **list**

295

class **Sequence**

Class that helps to generate unique numbers (unsigned longs).

Public Members

	Sequence (unsigned long starts_with = 0)
void	**Set** (unsigned long num)
unsigned long	**Get** ()
unsigned long	**GetNext** ()
void	**Reset** ()

Private Members

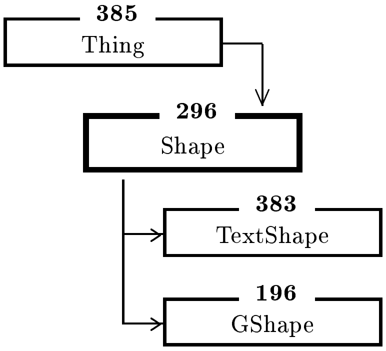
unsigned long	**seq**
unsigned long	**start**

296

class **Shape** : public Thing

(abstract) shape class, superclass of all possible shapes (graphics + text).

Inheritance



Public Members

Shape (ShapeView *s, Grafport *g)
creates a shape at position (0, 0) and size (0, 0).

bool **IsShape** () const

virtual bool
IsTextShape () const

virtual Shape*
Clone ()

virtual void
Draw () *Draws the shape: shape itself and its text and selection handles.*

virtual void
Undraw () *Erases the shape: by redrawing it in XOR mode.*

virtual void
UpdatePosition (const Point *position, bool snap=True)
update position and redraw.

virtual void

UpdateSize (int w, int h)
update size and redraw.

virtual void
UpdateMove (const Point *delta)
update movement and redraw.

virtual void
UpdateSelect (bool s)
update selection handles and redraw.

virtual void
UpdateFirstSelect (bool s)
update first selection handles and redraw.

virtual void
UpdateColor (string *color)
update color and redraw.

virtual void
SetSelect (bool s)

virtual void
SetFirstSelect (bool s)

bool **IsFirstSelected** () const

void **SetVisible** (bool s)

void **SetSubject** (Subject *s)

virtual void
SetView (ShapeView *v)

Subject* **GetSubject** () const

ShapeView*
GetView () const

virtual void
SetGrafport (Grafport *g)

void **SetColor** (const string *c)

const string*
GetColor () const

virtual void
Move (const Point *delta)
move position by delta.

virtual void

AdjustSize () *update size to make the shape fit the text shape strings.*

virtual void
DrawOutLine (const Point *center)
Draw outline of the shape (when it is dragged).

virtual Point
GiveSnp (Point) const
intersection of boundary and line from given point to shape center.

virtual Point
GiveSnp (Point, int, int, int &)
As GiveSnp but takes into account distribution of multiple lines.

virtual Point
GiveOrthoSnp (Point to) const
Calc the intersection of an orthogonal line with shape boundary.

void **SetHandles** (int n)
set n handle rectangles.

virtual int **HitHandles** (int x, int y)
Return (x, y) handle hit (-1 if not, 0 when shape is line).

virtual bool
ContainsPt (int x, int y)
returns whether (x, y) is geometrically in the shape.

const Point*
GetPosition () const

const Point*
GetTopLeft () const

bool **IsSelected** () const

bool **IsVisible** () const

bool **IsDuplicate** () const

int **GetWidth** () const

int **GetHeight** () const

static const int
MIN_HEIGHT *minimal height of a shape.*

static const int

MIN_WIDTH *minimal width of a shape.*

virtual void
DrawDuplicate ()
draw duplication indicator (asterisk).

void **UndrawDuplicate** ()

void **SetDuplicate** (bool b)

void **UpdateDuplicate** (bool b)
set whether shape is a duplicate shape.

virtual bool
IsLine () const

virtual int **GetLeftMost** () const
smallest x-coordinate in shape.

virtual int **GetTopMost** () const
smallest y-coordinate in shape.

virtual int **GetRightMost** () const
largest x-coordinate in shape.

virtual int **GetBottomMost** () const
largest y-coordinate in shape.

void **WriteMembers** (OutputFile *ofile)
write members of shape to file.

bool **ReadMembers** (InputFile *f, double format)
read and parse shape members from file ofile.

virtual bool
SetAssocSubject (AssocList *)
update subject reference from assocList.

bool **SetAssocView** (AssocList *)
update view reference from assocList.

bool **CheckReferences** ()
Check if subject exists as graph element and that view exists.

virtual void

SetTextShape () *update the strings of text shapes with the subject strings.*

virtual TextShape*
ChooseTextShape (int, int)
return a text shape near (x, y).

virtual TextShape*
HitTextShape (int, int)
return the text shape at (x, y).

virtual bool
InTextArea (int x, int y)
Returns if (x, y) is in text-area.

virtual bool
HasNameString (const string *, bool, bool) const
true if shape's name text shape contains 's'.

virtual bool
HasString (const string *, bool, bool) const
true if some of shape's text shapes contains 's'.

virtual bool
HasString (const string *, bool, bool, List<TextShape *> *)
put all text shapes containing 's' in the list l.

virtual bool
HasTextShape (TextShape *t) const
return if it has t as one of the text shapes.

virtual void
MoveRaw (const Point *delta)
move the shape over exact delta, without snap.

static const int
HANDLE_WIDTH
size of selection handle.

Protected Members

Grafport* **GetGrafport** () const

virtual void

SetDrawAttributes ()
set attributes of grafport before drawing.

void **SetOutlineAttributes** ()
set attributes of grafport for drawing outlines.

virtual void
SetSize (int w, int h)
set the size of the shape (with a minimum).

virtual void
SetPosition (const Point *position, bool snap=True)
set the position (takes into account the grid when snap=True).

virtual void
SnapTopleft () *align the shape to the grid (by its topleft).*

virtual void
SnapPosition () *align the shape to the grid (by its center).*

void **SetWidth** (int w)

void **SetHeight** (int h)

void **SetNrHandles** (int n)

DiagramViewer*
GetViewer () const

void **PositiveCoord** (const Point*)
change position so that shape has only positive coordinates.

static const int
MIN_HANDLES
minimal nr. of handles.

virtual void
DrawSelect () *Draw the handles if object is (first) selected.*

virtual void

	DrawGraySelect ()	*Draw grey handles if the object is selected.*
virtual void	**UndrawSelect** ()	
virtual void	**UndrawGraySelect** ()	
void	**SetTopLeft** (const Point *topLeft)	*set the top-left and position (take into account width and height).*
virtual void	**SetPositionHandles** ()	*set all handle rectangles.*
void	**ForcePosition** (const Point *position)	

Private Members

Subject*	**subject**	*the element in the graph that the shape represents.*
DiagramViewer*	**viewer**	*the diagram viewer (manages all views and shapes).*
ShapeView*	**view**	*the particular view in which shape resides.*
Grafport*	**grafport**	*grafport to draw the shape.*
Point	**position**	*the center position of the shape.*
Point	**topleft**	*the top-left position of the shape.*
int	**height**	*the height of the shape.*
int	**width**	*the width of the shape.*
bool	**selected**	*whether the shape is selected.*
bool	**firstSelected**	*whether the shape is selected as first.*
bool	**visible**	*the shape is visible?*
string	**color**	*the color of the shape.*
Rectangle	**handles** [8]	*the position and size of the selection handles.*

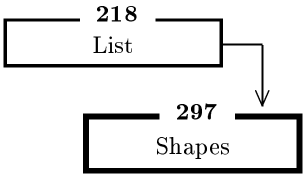
int	**nrHandles**	*the number of handles (<= 8).*
bool	**duplicate**	*Subject of shape has more than one representation?*
virtual bool	**HasMinSize** ()	*Has the shape always a minimal size?*

297

class **Shapes** : public List<GShape *>

list of shapes class; copied shapes are put in the assoc list.

Inheritance



Public Members

	Shapes (AssocList *al)
	Shapes (const Shapes &shapes)
AssocList*	**GetAssocList** () const

Private Members

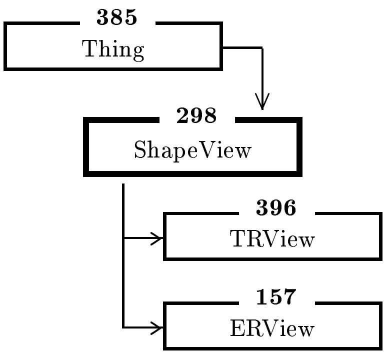
GShape*	**CloneShape** (GShape *shape)
AssocList*	**assocList**

298

class **ShapeView** : public Thing

Class containing a single view (set of shapes).

Inheritance



Public Members

	ShapeView (DiagramViewer *v)	
virtual	**~ShapeView** ()	
int	**GetClassType** () const	
bool	**IsView** () const	
void	**Draw** ()	*draw all shapes in the view.*
void	**Undraw** ()	*erase all shapes in the view.*
void	**SetGrafport** (Grafport *g)	*set grafport of all shapes.*
Node*	**GetParent** () const	*each view has a parent node.*
void	**SetParent** (Node *n)	
const string*	**GetIndex** () const	*each view has a unique index.*
void	**SetIndex** (string *s)	
unsigned	**NrOfShapes** () const	

the number of shapes.

List <GShape *> *
GetShapes () const

List <GShape *> *
GetSelection () const

bool **HasShape** (GShape *s)

void **Snap** (Point *p) *Snap point to discrete point-position.*

int **Snap** (int i) *Snap int to discrete point-position.*

DiagramViewer*
GetViewer () const

void **WriteShapes** (OutputFile *f)
write all shapes to file (in separate clauses).

bool **ReadMembers** (InputFile *f, double format)
read view section.

bool **SetReferences** (AssocList *)
update references when view is read.

bool **CheckReferences** ()
Check parent node, Check that shapes have right view reference.

void **CheckShapes** () *find all shapes with wrong references;*

unsigned **NrSelected** () const
the number of selected shapes.

GShape* **FirstSelected** () const
return the first selected shape.

void **DeselectAll** () *deselect all shapes in the view.*

virtual void
SelectShape (GShape *shape)
add shape to selection.

virtual void
FirstSelectShape (GShape *shape)
add shape to selection as primus inter pares.

void **DeselectShape** (GShape *shape)

remove shape to selection.

GShape* **HitShape** (int x, int y)
return shape from view if (x, y) is in shape.

NodeShape* **HitNodeShape** (int x, int y)
return node from view if (x, y) is in shape.

Line* **HitLine** (int x, int y, int &whichsegment)
return line from view if (x, y) is in line. segment is set to which segment.

Line* **HitLineHandle** (int x, int y, int &which)
return line from view if (x, y) hits handle.

TextShape* **HitTextShape** (int x, int y, bool emptyallowed=True)
return text-shape from view if (x, y) is in shape.

int **GetCurrentShapeNr** () const

void **SetCurrentShapeNr** (int i)

void **SetCurrentShapeNr** (GShape *s)

GShape* **GetNextShape** (const string *s, bool sens, bool substring, bool nameOnly)
return the next shape containing s as name.

void **GetShapes** (List<GShape *> *shapes, const string *s, bool sens, bool substring, bool nameOnly)
get shapes containing s as name (case sensitive, substring).

void **CalcSizeElements** (Point &topLeft, Point &bottomRight)

void **CalcSizeSelection** (Point &topLeft, Point &bottomRight)

void **CalcSizeShapes** (List<GShape *> *shapes, Point &topLeft, Point &bottomRight)

calculate covered area of shapes in grafport.

void **AddShapeRaw** (GShape *shape)
add to viewer and draw/redraw shape.

virtual void **AddShape** (GShape *shape)
add to viewer and draw/redraw shape and recalc. line positions.

void **AddShapes** (List<GShape *> *shapes)

void **RemoveShapeRaw** (GShape *shape)
remove from viewer and erase shape.

virtual void **RemoveShape** (GShape *shape)
remove from viewer, and erase drawing and recalc line positions.

void **RemoveShapes** (List<GShape *> *shapes)

GShape* **GetShape** (Subject *subject)
return first shape from shapes having given subject.

void **GetShapes** (Subject *subject, List<GShape *> *shapes)
put in shapes all shapes from shapes having given subject.

void **MultipleLinesPosUpdate** (Line *line)
update the positions of the lines connecting same shapes as line.

void **MultipleLines** (Line *line)
update multiple lines between two nodes (called when line added).

void **CompleteShapes** (List<GShape *> *s, GShape *n1, GShape *n2)
add to s all lines that connect shapes n1 and n2.

void **CompleteLines** (List<GShape *> *shapes)
add shapes which are connected by the lines in shapes.

void **CompleteShapes** (List<GShape *> *shapes)

		add lines to shapes which connect the shapes in shapes.
void	**DrawNames** ()	*draw only names in view.*
void	**ShowAll** ()	*make all shapes in view visible.*

Private Members

DiagramViewer*	**viewer**	*the viewer that manages the view.*
Node*	**parent**	*the parent node.*
string	**index**	*the view index.*
int	**currentShapeNr**	*current shape number.*
List <GShape *> *	**shapes**	*the list of displayed shapes.*
List <GShape *> *	**selection**	*the list of selected shapes.*
void	**WriteMembers** (OutputFile *)	*write view clause to file.*

299

class **Side**

different sides of an item

Public Members

enum **Type** *the 4 boundaries of a cell + all boundaries.*

static Type
String2Type (const string* s)
e.g. convert "Top" to TOP.

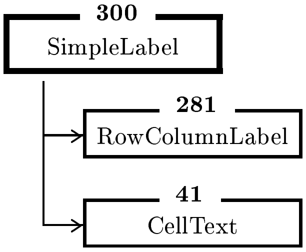
static void
Type2String (Type t, string* s)
e.g. convert TOP to "Toop".

300

class **SimpleLabel**

abstract (simple) text label class.

Inheritance



Public Members

SimpleLabel (Grafport *g, XFont *font)

SimpleLabel (const SimpleLabel &text)

virtual **~SimpleLabel** ()

friend int **operator==** (const SimpleLabel &t1, const SimpleLabel &t2)

friend int **Compare** (SimpleLabel *r1, SimpleLabel *r2)

virtual void **Draw** () — *draw label.*

virtual void **Undraw** () — *erase label.*

void **DrawOutLine** (const Point *c) — *when label is dragged only a dotted box (outline) is drawn*

virtual void **UpdatePosition** (const Point *) — *update position and redraw.*

virtual void **UpdateText** (const string *) — *update label and redraw.*

void **SetPosition** (const Point *pt)

set position.

void **SetText** (const string *s)
set label text

void **SetGrafport** (Grafport *g)
set grafport

void **CalcSize** () *calculate text size.*

bool **HitText** (int x, int y)
true iff (x, y) is in text.

const Point*
GetPosition () const
simple 'get' actions.

const Point*
GetTopLeft () const

const string*
GetText () const

int **GetWidth** () const

int **GetHeight** () const

XFont* **GetFont** () const

void **SetFont** (XFont *f)

virtual void
UpdateFont (XFont *font)

int **FontHeight** () const

static const int
MIN_TEXT_WIDTH
minimal width of cell text outline.

static const int
MIN_TEXT_HEIGHT
minimal height of cell text outline.

Protected Members

Grafport* **GetGrafport** () const

Private Members

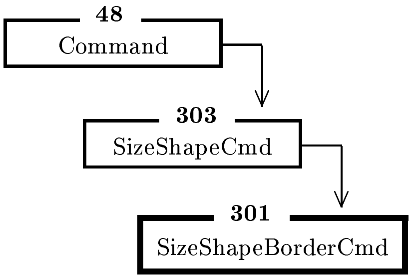
Point	**position**	*center position of text.*
Point	**topLeft**	*topleft position of text.*
int	**width**	*text width (depends on number of characters and point size + font).*
int	**height**	*text height (depends on font).*
string	**text**	*text string*
Grafport*	**grafport**	
XFont*	**font**	
void	**SetSize** (int wd, int ht)	*set size*
void	**PositiveCoord** ()	*make sure text has positive coordinates.*

301

class **SizeShapeBorderCmd** : public SizeShapeCmd

command class for resizing a shape.

Inheritance



Public Members

	SizeShapeBorderCmd (GShape *shape, int direction)
void	**Execute** ()
void	**UnExecute** ()
const char*	**GetName** () const

Private Members

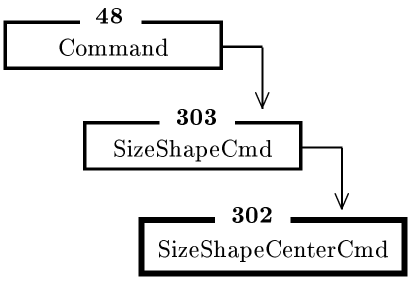
Point	**oldTopLeft**
Point	**newTopLeft**
void	**DrawOutline** (const Point *pt)

302

class **SizeShapeCenterCmd** : public SizeShapeCmd

command class for resizing a shape.

Inheritance



Public Members

	SizeShapeCenterCmd (GShape *shape, int direction)
void	**Execute** ()
void	**UnExecute** ()
void	**Executable** (int w, int h)
const char*	**GetName** () const

Private Members

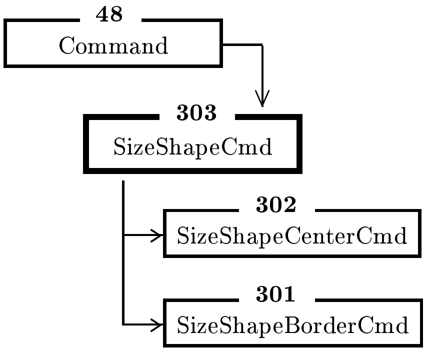
Point	**center**
void	**DrawOutline** (const Point *pt)

303

class **SizeShapeCmd** : public Command

abstract command class for resizing a shape.

Inheritance



Public Members

	SizeShapeCmd (GShape *shape, int direction)
void	**TrackMouse** (TrackType aPhase, Point *anchorPoint, Point *previousPoint, Point *nextPoint)

Protected Members

GShape*	**shape**
int	**newWidth**
int	**newHeight**
int	**oldWidth**
int	**oldHeight**
int	**direction**
bool	**constrainX**
bool	**constrainY**

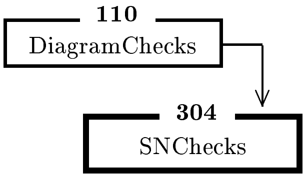
virtual void
DrawOutline (const Point *pt)

304

class **SNChecks** : public DiagramChecks

implements checks for soft constraints on SNDs.

Inheritance



Public Members

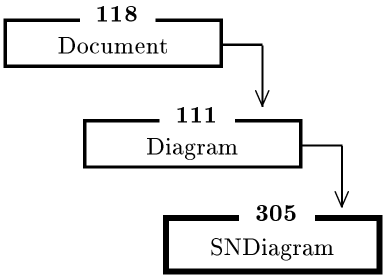
	SNChecks (Diagram *d, Graph *g)
unsigned	**CheckConnectionCoherence** (int type, string &chkbuf)

305

class **SNDiagram** : public Diagram

system network diagram class.

Inheritance



Public Members

	SNDiagram (Config *, SNWindow *, SNViewer *, SNGraph *)
virtual	**~SNDiagram** ()
Node*	**CreateNode** ()
Edge*	**CreateEdge** (Subject *n1, Subject *n2)
NodeShape*	**CreateNodeShape** (Node *node, int x, int y)
Line*	**CreateLine** (Edge *edge, GShape *fromShape, GShape *toShape, List<Point *> *line)
void	**UpdateNodeType** (int n)
void	**UpdateEdgeType** (int n)
void	**CheckDocument** ()
bool	**SetConstraint** (Subject *subject, const string *s, unsigned nr)
bool	**SetText** (TextShape *t, const string *s)

Protected Members

Thing*	**CreateThing** (int classNr)
bool	**CheckEdgeConstraints** (Subject *s1, Subject *s2)
unsigned	**CheckConnectionCoherence** (int type, string &chkbuf)

Private Members

SNChecks*
snChecks

static const int
DIAMOND_WIDTH

static const int
DIAMOND_HEIGHT

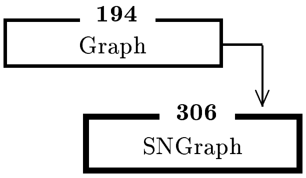
static const int
CIRCLE_WIDTH

306

class **SNGraph** : public Graph

system network graph class.

Inheritance



Public Members

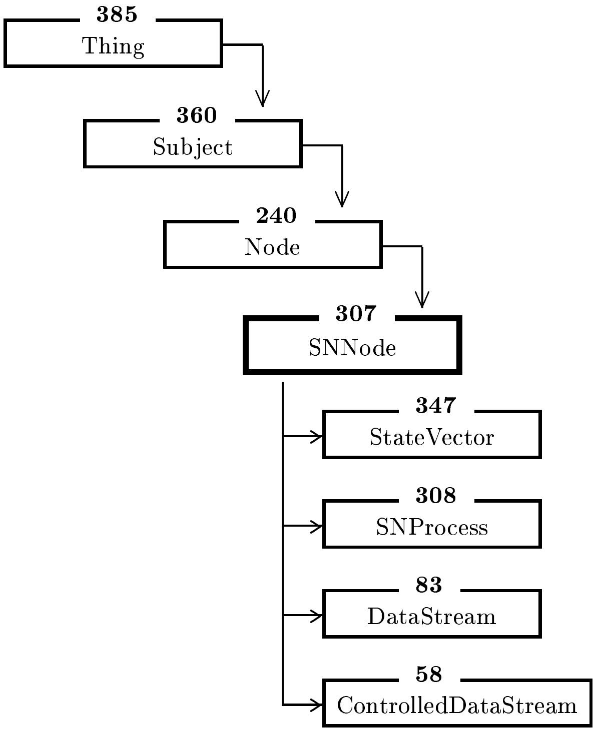
	SNGraph ()
void	**InitConnections** ()

307

class **SNNode** : public Node

abstract system network node class.

Inheritance



Public Members

SNNode (SNGraph *g)

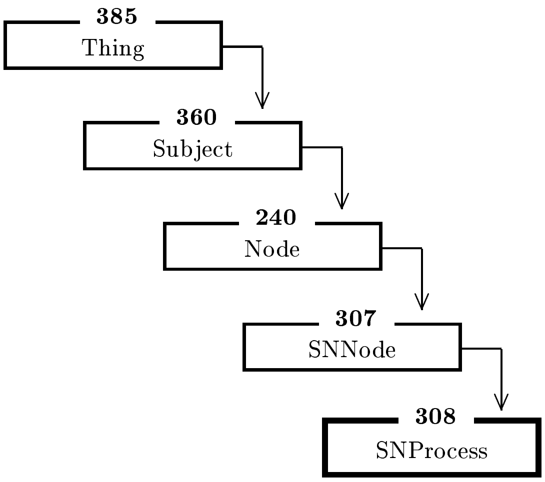
Subject::NameErrType
SetName (const string *name)

308

class **SNProcess** : public SNNode

system network process node class.

Inheritance



Public Members

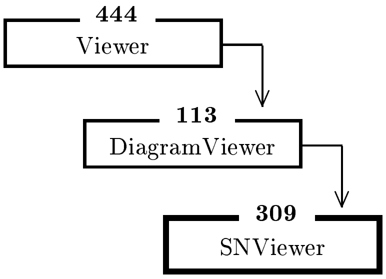
	SNProcess (SNGraph *g)
int	**GetClassType** () const
Subject*	**Clone** ()

309

class **SNViewer** : public DiagramViewer

system network viewer class

Inheritance



Public Members

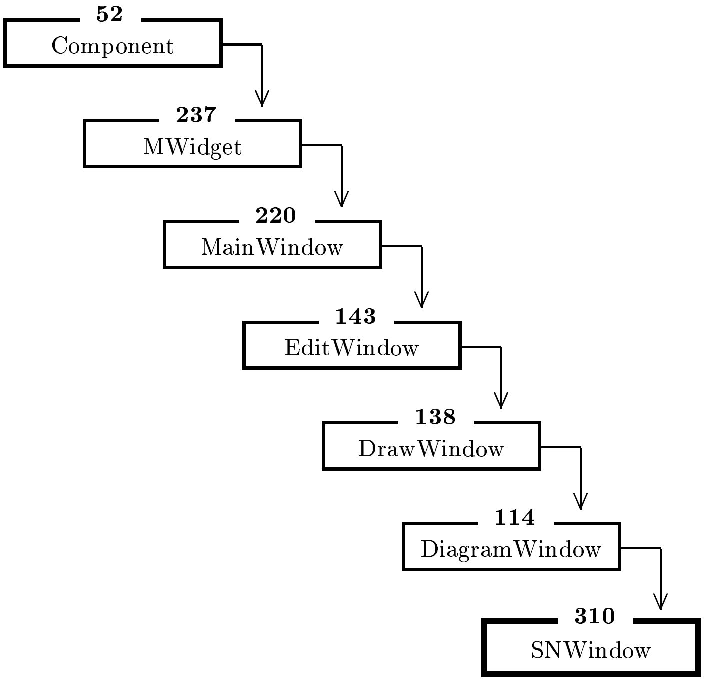
SNViewer (Config *c, SNWindow *w)

310

class **SNWindow** : public DiagramWindow

system network edit window class.

Inheritance



Public Members

SNWindow (const char *name)

Private Members

int **GetTool** ()

DiagramViewer* **CreateViewer** (Config *c)

Graph* **CreateGraph** ()

Diagram* **CreateDiagram** (Config *c, DiagramViewer *v, Graph *g)

void **DetermineIcons** ()

static const char*
SN_NODE_NAMES []

static const Bitmap
SN_NODE_ICONS []

static const char*
SN_EDGE_NAMES []

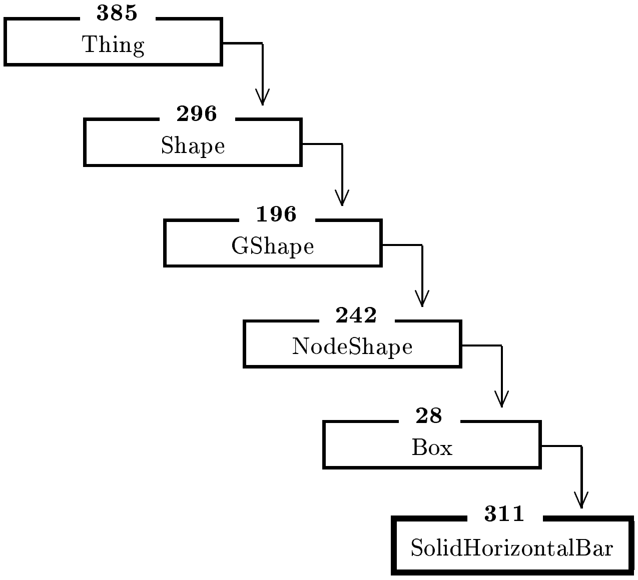
static const Bitmap
SN_EDGE_ICONS []

311

class **SolidHorizontalBar** : public Box

A solid horizontal bar with fixed height 1

Inheritance



Public Members

static const int
BAR_WIDTH

static const int
BAR_HEIGHT

SolidHorizontalBar (ShapeView *v, Grafport *g, int x, int y, int w = BAR_WIDTH, int h = BAR_HEIGHT)

SolidHorizontalBar (const NodeShape &c)

Shape* **Clone** ()

Point **GiveSnp** (Point to) const

Point **GiveSnp** (Point, int, int, int &)
As GiveSnp but takes into account distribution of multiple lines.

bool **ReadMembers** (InputFile *ifile, double format)

int **GetClassType** () const

static const int
BAR_HANDLES

Protected Members

void **SetSize** (int w, int h)

int **HitHandles** (int x, int y)

Private Members

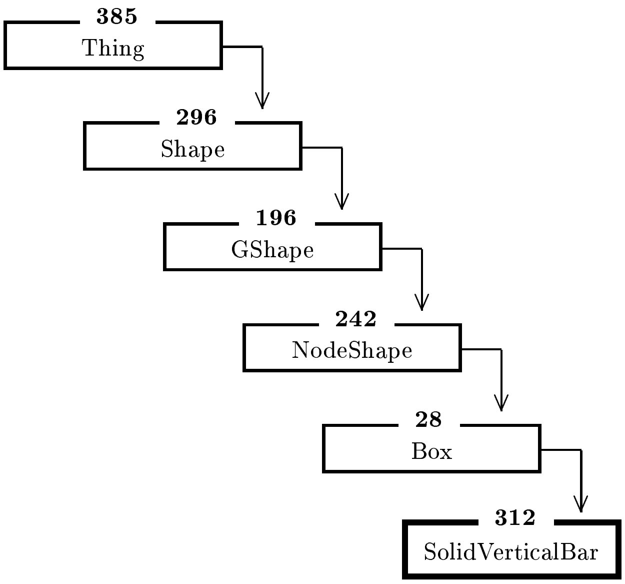
void **Init** ()

312

class **SolidVerticalBar** : public Box

A solid vertical bar with fixed width 1

Inheritance



Public Members

static const int
BAR_WIDTH

static const int
BAR_HEIGHT

SolidVerticalBar (ShapeView *v, Grafport *g,
int x, int y,
int w = BAR_WIDTH,
int h = BAR_HEIGHT)

SolidVerticalBar (const NodeShape &c)

Shape* **Clone** ()

Point **GiveSnp** (Point to) const

Point **GiveSnp** (Point, int, int, int &)
As GiveSnp but takes into account distribution of multiple lines.

bool **ReadMembers** (InputFile *ifile, double format)

int **GetClassType** () const

static const int **BAR_HANDLES**

Protected Members

void **SetSize** (int w, int h)

int **HitHandles** (int x, int y)

Private Members

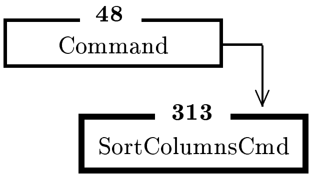
void **Init** ()

313

class **SortColumnsCmd** : public Command

sort columns in table command.

Inheritance



Public Members

	SortColumnsCmd (Table *t, TableViewer *v)
void	**Execute** ()
void	**UnExecute** ()
const char*	**GetName** () const

Private Members

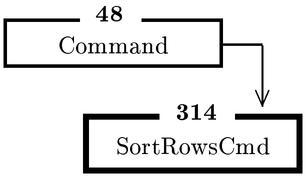
TableViewer*	**tviewer**
int	**row**
List <CellText *>	**texts**
List <CellColumn *>	**originals**

314

class **SortRowsCmd** : public Command

sort rows in table command.

Inheritance



Public Members

	SortRowsCmd (Table *t, TableViewer *v)
void	**Execute** ()
void	**UnExecute** ()
const char*	**GetName** () const

Private Members

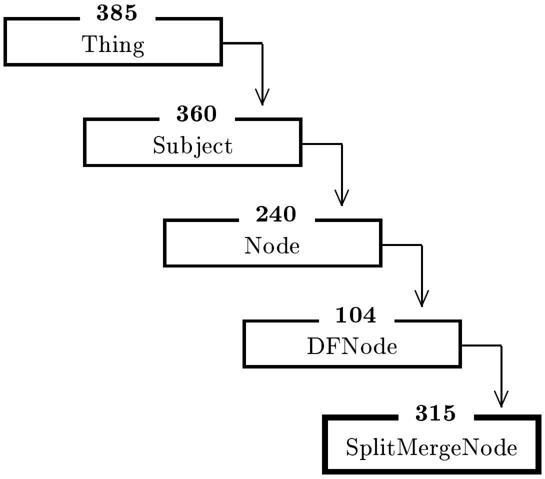
TableViewer*	**tviewer**
int	**column**
List <CellText *>	**texts**
List <CellRow *>	**originals**

315

class **SplitMergeNode** : public DFNode

split or merge (DF) node class.

Inheritance



Public Members

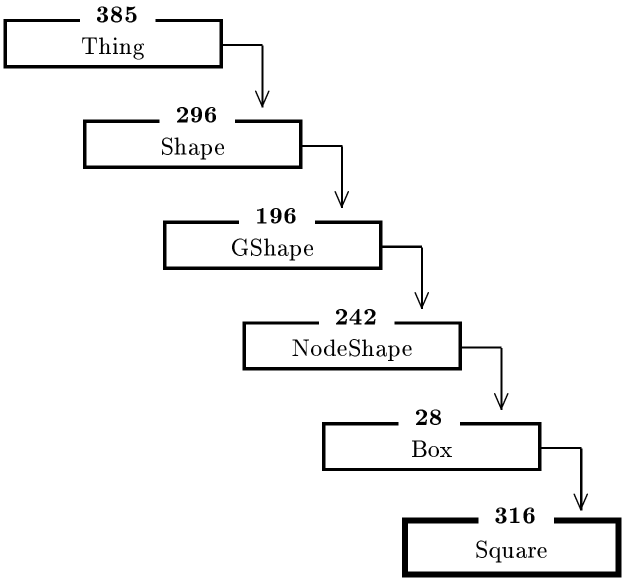
	SplitMergeNode (DFGraph *g)
int	**GetClassType** () const
Subject*	**Clone** ()
Subject::NameErrType	**SetName** (const string *s)

316

class **Square** : public Box

box class with equal height and width.

Inheritance



Public Members

static const int
SQUARE_WIDTH

Square (ShapeView *v, Grafport *g, int x, int y, int w = SQUARE_WIDTH)

Square (const NodeShape &)

Shape* **Clone** ()

int **GetClassType** () const

Protected Members

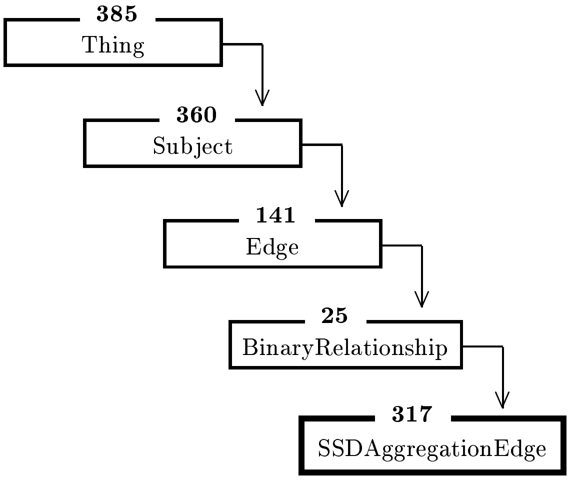
void **SetSize** (int width, int height)

317

class **SSDAggregationEdge** : public BinaryRelationship

aggregation edge class in a SSD.

Inheritance



Public Members

	SSDAggregationEdge (SSDGraph *g, Subject *n1, Subject *n2)
Subject*	**Clone** ()
int	**GetClassType** () const

Private Members

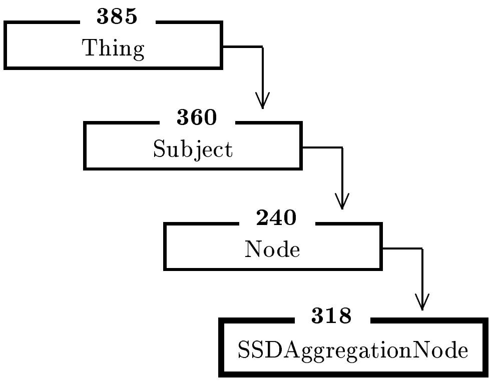
bool	**SetConstraint** (string *cons, const string *name)

318

class **SSDAggregationNode** : public Node

TSSD aggregation node class.

Inheritance



Public Members

	SSDAggregationNode (SSDGraph *g)	
int	**GetClassType** () const	
Subject*	**Clone** ()	

Private Members

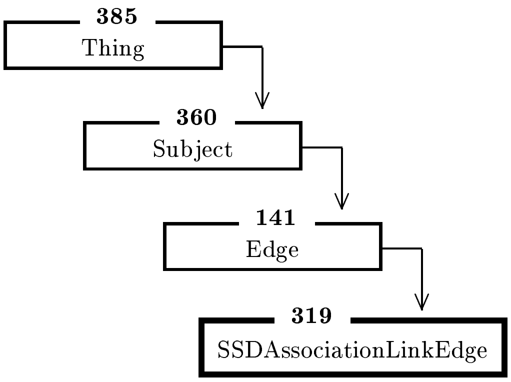
bool	**composite**	*a bool composite : True = composition, False = Aggregation*

319

class **SSDAssociationLinkEdge** : public Edge

SSD association-link edge class.

Inheritance



Public Members

SSDAssociationLinkEdge (SSDGraph *g,
Subject *n1,
Subject *n2)

int **GetClassType** () const

Subject* **Clone** ()

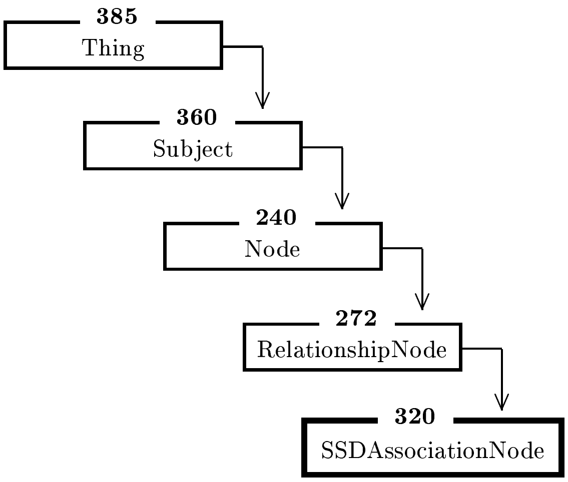
Subject::NameErrType
SetName (const string *s)

320

class **SSDAssociationNode** : public RelationshipNode

SSD association node class.

Inheritance



Public Members

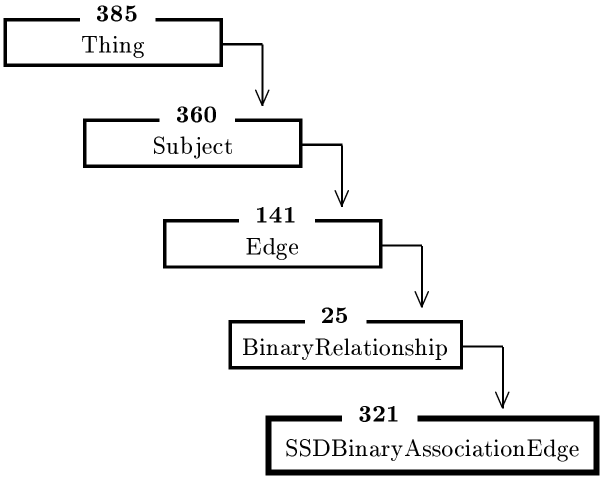
	SSDAssociationNode (SSDGraph *g)
void	**GetComponents** (List<string *> *components)
int	**GetClassType** () const
Subject*	**Clone** ()

321

class **SSDBinaryAssociationEdge** : public BinaryRelationship

binary association edge class in an SSD.

Inheritance



Public Members

	SSDBinaryAssociationEdge (SSDGraph *g, Subject *n1, Subject *n2)
Subject*	**Clone** ()
int	**GetClassType** () const

Private Members

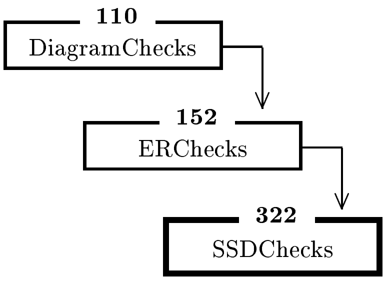
bool	**SetConstraint** (string *cons, const string *name)

322

class **SSDChecks** : public ERChecks

implements checks for soft constraints on SSD diagrams.

Inheritance



Public Members

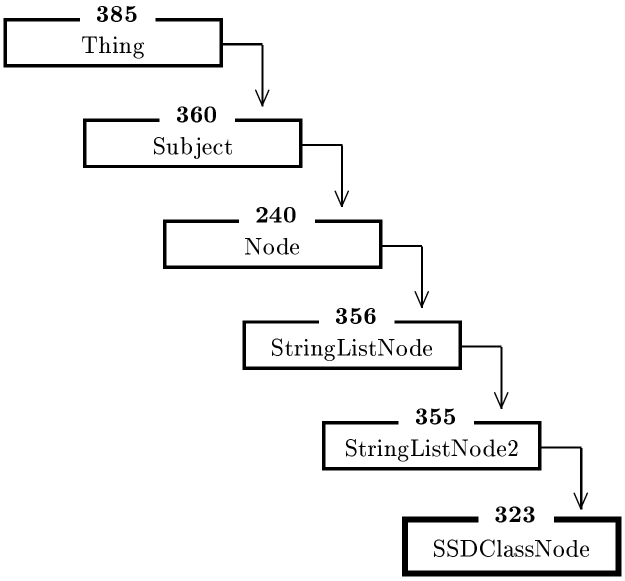
SSDChecks (Diagram *d, Graph *g)

323

class **SSDClassNode** : public StringListNode2

TSSD class node.

Inheritance



Public Members

	SSDClassNode (SSDGraph *g)
	SSDClassNode (const SSDClassNode &c)
int	**GetClassType** () const
Subject*	**Clone** ()

StringListNode::StringErrorType
SetString (const string *s, unsigned n, bool update)

StringListNode::StringErrorType
SetString2 (const string *s, unsigned n, bool update)

StringListNode::StringErrorType

SetStereotype (const string *s)

StringListNode::StringErrorType
SetProperties (const string *s)

const string*
GetStereotype ()

const string*
GetProperties () const

void **WriteMembers** (OutputFile *f)

bool **ReadMembers** (InputFile *f, double format)

Private Members

string **stereotype**

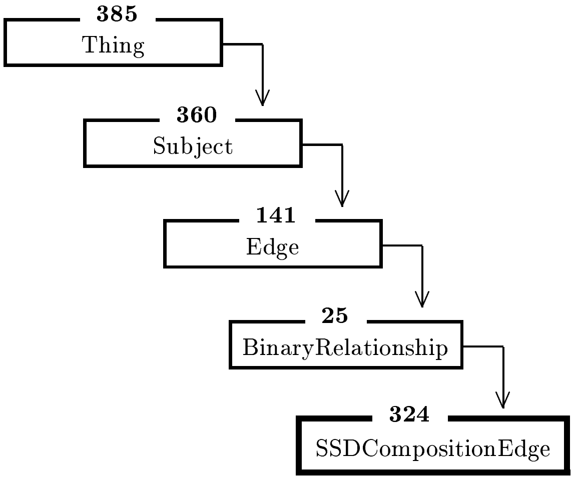
string **properties**

324

class **SSDCompositionEdge** : public BinaryRelationship

composition edge class in a SSD.

Inheritance



Public Members

	SSDCompositionEdge (SSDGraph *g, Subject *n1, Subject *n2)
Subject*	**Clone** ()
int	**GetClassType** () const

Private Members

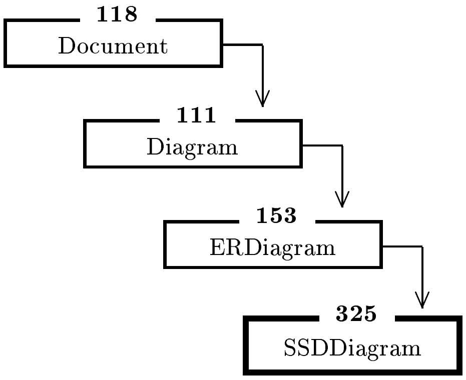
bool	**SetConstraint** (string *cons, const string *name)

325

class **SSDDiagram** : public ERDiagram

UML static structure diagram class.

Inheritance



Public Members

	SSDDiagram (Config *, SSDWindow *, SSDViewer *, SSDGraph *)
virtual	**~SSDDiagram** ()
Node*	**CreateNode** ()
Edge*	**CreateEdge** (Subject *n1, Subject *n2)
NodeShape*	**CreateNodeShape** (Node *node, int x, int y)
Line*	**CreateLine** (Edge *edge, GShape *fromShape, GShape *toShape, List<Point *> *line)
void	**UpdateNodeType** (int n)
void	**UpdateEdgeType** (int n)
bool	**SetText** (TextShape *t, const string *s)
bool	**SetAttribute** (StringListNode *n, const string *s, unsigned nr)

Set Attribute in class obj at position nr.

bool **SetOperation** (StringListNode2 *n, const string *s, unsigned nr)
Set Operation in obj at position nr.

bool **SetRoleName** (Subject *subj, const string *s, unsigned nr)
Set nr-th role name of subject. Check name syntax.

bool **SetStereotype** (SSDClassNode *node, const string *s)
Set stereotype string of a class node.

bool **SetProperties** (SSDClassNode *node, const string *s)
set properties string of a class node.

void **CheckDocument** ()
check the soft constraints on a SSD.

Protected Members

void **MakeErrorMessage** (StringListNode *node, StringListNode::StringErrorType result, const string *s, string *text)
Give error after failing to update attribute or operation.

Thing* **CreateThing** (int classNr)

SSDChecks* **GetSSDChecks** ()

static const int **MINI_ELLIPSE_WIDTH**

static const int **MINI_ELLIPSE_HEIGHT**

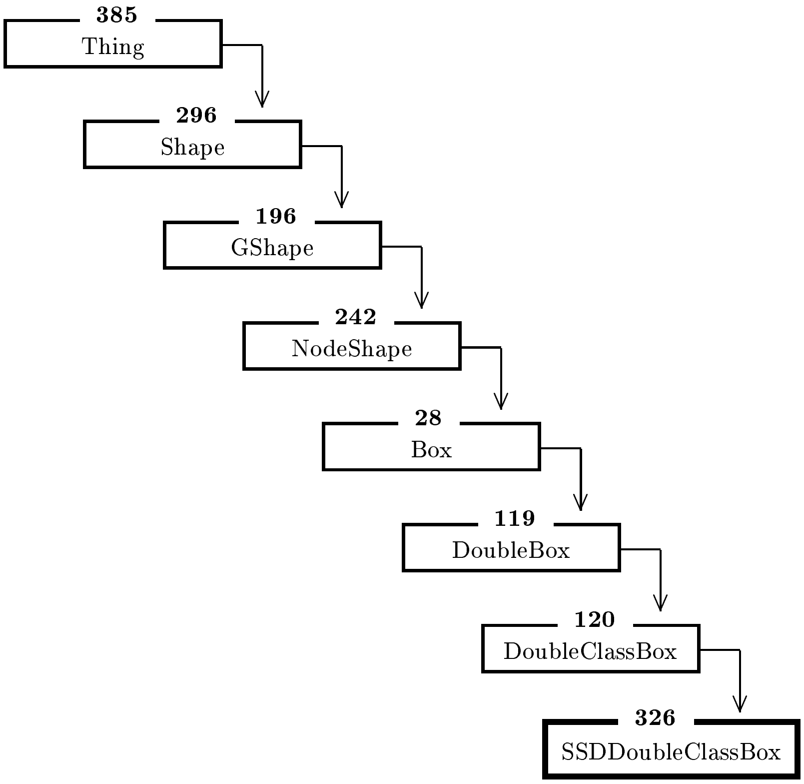
Private Members

SSDChecks*
ssdChecks

326 class **SSDDoubleClassBox** : public DoubleClassBox

DoubleBox for representing object classes in a SSD.

Inheritance



Public Members

SSDDoubleClassBox (ShapeView *v, Grafport *g, int x, int y, int w = DOUBLECLASS-BOX_WIDTH, int h = DOUBLECLASS-BOX_HEIGHT)

SSDDoubleClassBox (const SSDDoubleClassBox &s)

SSDDoubleClassBox (const SSDTripleClassBox &s)

SSDDoubleClassBox (const SSDSingleClassBox &s)

~SSDDoubleClassBox ()

Shape* **Clone** ()

int **GetClassType** () const

void **SetFont** (XFont *ft)

void **SetTextColor** (const string *color)

void **SetGrafport** (Grafport *g)

void **SetView** (ShapeView *v)

TextShape* **ChooseTextShape** (int x, int y)

bool **HasTextShape** (TextShape *t) const

void **AdjustSize** ()

void **UpdateShowStereotype** (bool b)

void **UpdateShowProperties** (bool b)

void **UpdateStereotypeLabelString** (const string *s)
update string of stereotype text shape and redraw.

const string* **GetStereotypeLabelString** ()

TextShape* **GetStereotypeLabel** () const

void **UpdatePropertiesLabelString** (const string *s)
update string of properties text shape and redraw.

const string* **GetPropertiesLabelString** ()

TextShape* **GetPropertiesLabel** () const

void **SetTextShape** ()

bool **HasString** (const string *s, bool sens, bool sub) const

		contains s as string (case sensitive, substring).
bool	**HasString** (const string*, bool, bool, List<TextShape *> *list)	
void	**SetShowStereotype** (bool b)	
void	**SetShowProperties** (bool b)	
bool	**IsShowStereotype** () const	
bool	**IsShowProperties** () const	
void	**WriteMembers** (OutputFile *f)	
bool	**ReadMembers** (InputFile *f, double format)	

Private Members

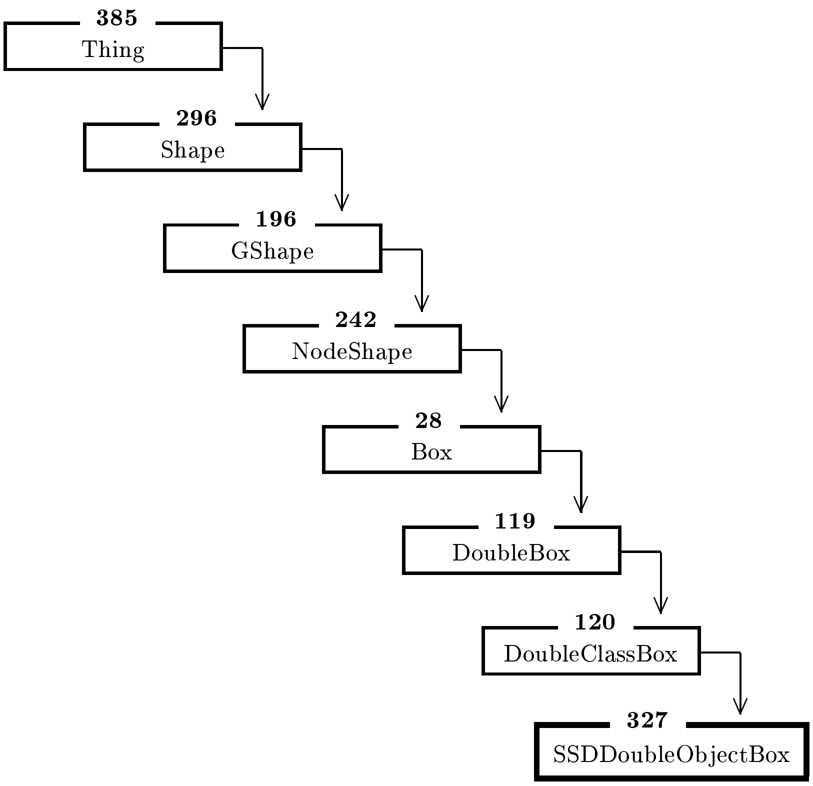
void	**DrawTextShapes** ()
int	**RequiredHeight** ()
int	**RequiredWidth** ()
void	**SetStereotypeLabelString** (const string *s)
void	**SetPropertiesLabelString** (const string *s)
void	**CalcPositionLabels** ()
void	**CalcPositionStereotypeLabel** ()
void	**CalcPositionPropertiesLabel** ()
void	**AdjustSizes** (const string *s)
TextShape*	**stereotypeLabel**
TextShape*	**propertiesLabel**
bool	**showStereotype**
bool	**showProperties**

327

class **SSDDoubleObjectBox** : public DoubleClassBox

An object double box used for an object

Inheritance



Public Members

SSDDoubleObjectBox (ShapeView *v, Grafport *g, int x, int y, int w = DOUBLECLASS-BOX_WIDTH, int h = DOUBLECLASS-BOX_HEIGHT)

SSDDoubleObjectBox (const Box &s)

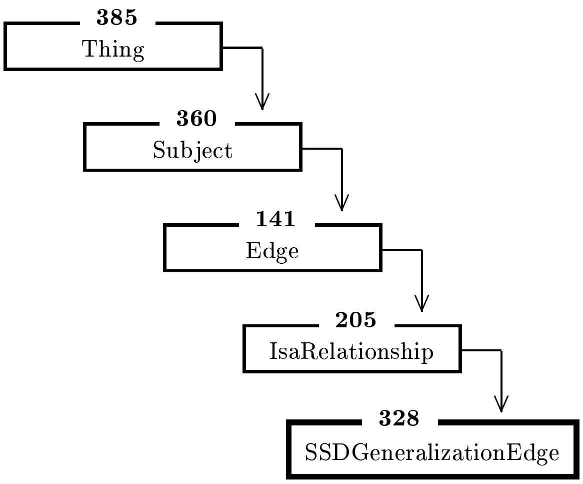
Shape*	**Clone** ()
int	**GetClassType** () const
bool	**SetAssocSubject** (AssocList *al)
void	**SetTextShape** ()
void	**DrawTextShapes** ()

328

class **SSDGeneralizationEdge** : public IsaRelationship

SSD generalization edge class.

Inheritance



Public Members

SSDGeneralizationEdge (SSDGraph *g, Subject *n1, Subject *n2)

Subject* **Clone** ()

int **GetClassType** () const

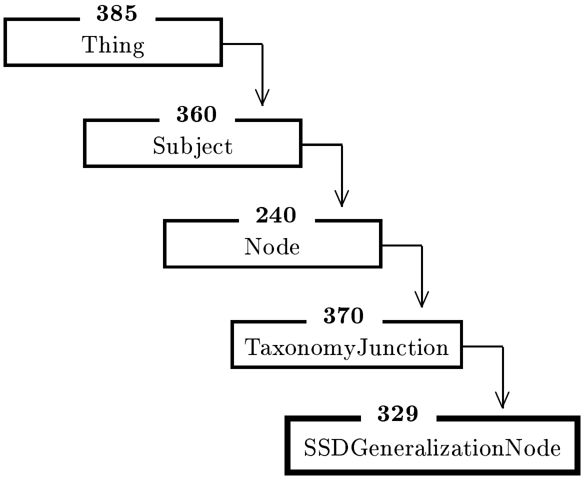
Subject::NameErrType **SetName** (const string *s)

329

class **SSDGeneralizationNode** : public TaxonomyJunction

TSSD generalization node class.

Inheritance



Public Members

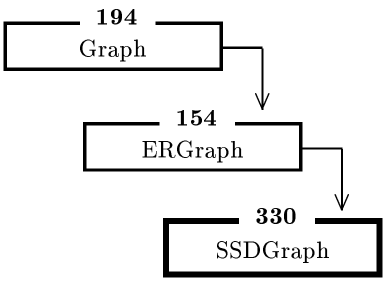
	SSDGeneralizationNode (SSDGraph *g)
Subject*	**Clone** ()
int	**GetClassType** () const
Subject::NameErrType	**SetName** (const string *name)

330

class **SSDGraph** : public ERGraph

UML static structure graph class

Inheritance



Public Members

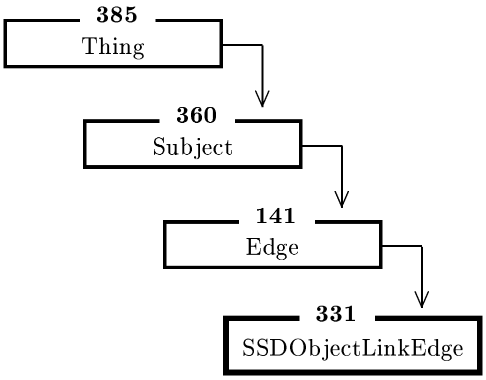
	SSDGraph ()
void	**InitConnections** ()

331

class **SSDObjectLinkEdge** : public Edge

Object Link (SSD) edge class.

Inheritance



Public Members

SSDObjectLinkEdge (SSDGraph *g, Subject *n1, Subject *n2)

Subject* **Clone** ()

int **GetClassType** () const

Subject::NameErrType
SetName (const string *name)

bool **SetRoleName1** (const string *s)

const string*
GetRoleName1 () const

bool **SetRoleName2** (const string *s)

const string*
GetRoleName2 () const

bool **SetRoleName** (string *cons, const string *name)

void **WriteMembers** (OutputFile *f)

bool **ReadMembers** (InputFile *f, double format)

Private Members

string **roleName1**

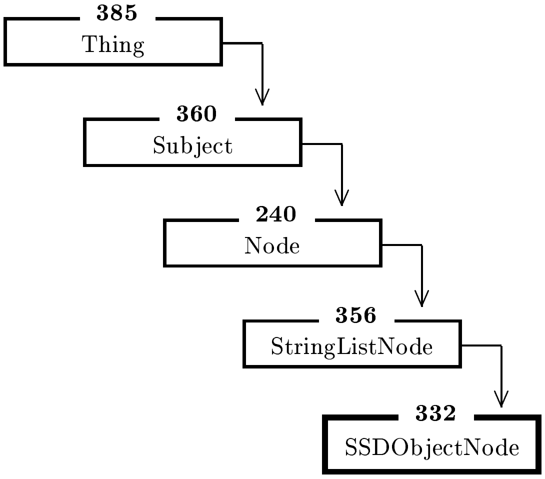
string **roleName2**

332

class **SSDObjectNode** : public StringListNode

object class (TSSD) node class.

Inheritance



Public Members

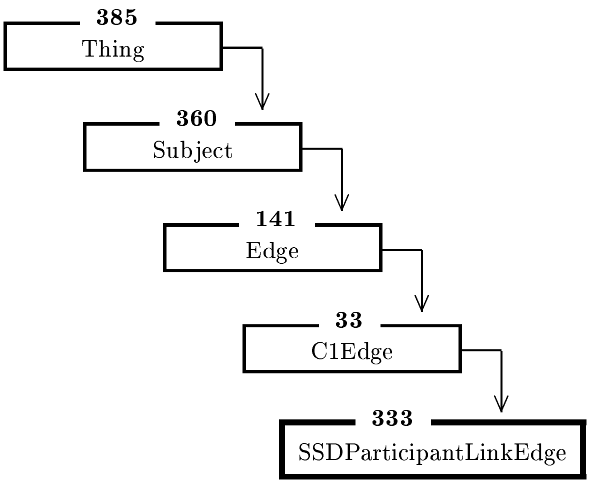
	SSDObjectNode (SSDGraph *g)
Subject*	**Clone** ()
int	**GetClassType** () const
StringListNode::StringErrorType	**SetString** (const string *s, unsigned n, bool update)

333

class **SSDParticipantLinkEdge** : public C1Edge

SSD participant edge class connecting N-ary relationship nodes.

Inheritance



Public Members

	SSDParticipantLinkEdge (SSDGraph *g, Subject *n1, Subject *n2)
	SSDParticipantLinkEdge (const SSDParticipantLinkEdge &bin)
int	**GetClassType** () const
Subject*	**Clone** ()
Subject::NameErrType	**SetName** (const string *s)
bool	**SetConstraint** (const string *s)
bool	**SetRoleName** (const string *s)
const string*	

	GetRoleName () const
void	**WriteMembers** (OutputFile *f)
bool	**ReadMembers** (InputFile *f, double format)

Private Members

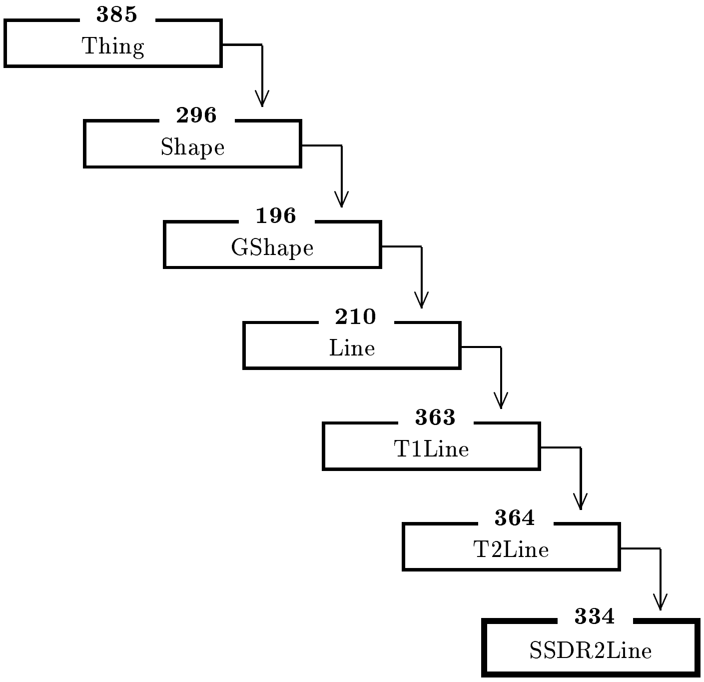
string	**roleName**

334

class **SSDR2Line** : public T2Line

line having role names at both sides (SSD links)

Inheritance



Public Members

SSDR2Line (ShapeView *v, Grafport *g,
GShape *n1, GShape *n2,
List<Point *> *line, bool Curved)

SSDR2Line (ShapeView *v, Grafport *g,
GShape *n1, GShape *n2,
bool Curved)

int **GetClassType** () const

Shape* **Clone** ()

bool	**SetAssocSubject** (AssocList *)	
void	**SetTextShape** ()	
TextShape*	**ChooseTextShape** (int x, int y)	
TextShape*	**HitTextShape** (int x, int y)	
void	**WriteMembers** (OutputFile *f)	
bool	**ReadMembers** (InputFile *f, double format)	

Protected Members

void	**DrawShape** ()

Private Members

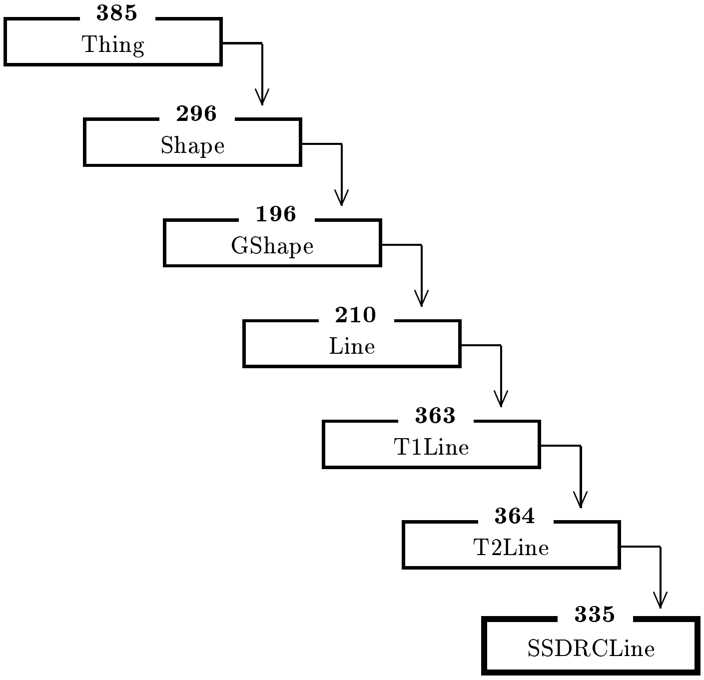
void	**InitTextShapes** ()

335

class **SSDRCLine** : public T2Line

line for representing SSD participant edges.

Inheritance



Public Members

	SSDRCLine (ShapeView *v, Grafport *g, GShape *n1, GShape *n2, List<Point *> *line, bool Curved)
	SSDRCLine (ShapeView *v, Grafport *g, GShape *n1, GShape *n2, bool Curved)
int	**GetClassType** () const
Shape*	**Clone** ()

bool	**SetAssocSubject** (AssocList *)
void	**SetTextShape** ()

Private Members

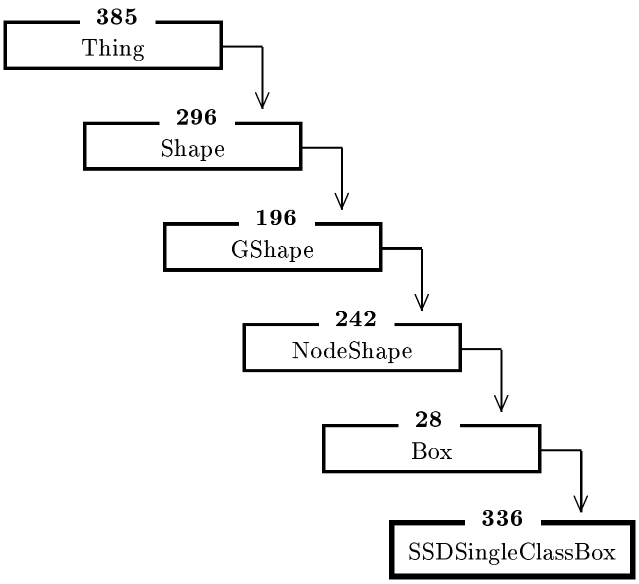
void	**InitTextShapes** ()

336

class **SSDSingleClassBox** : public Box

Box for representing object classes in a SSD.

Inheritance



Public Members

	SSDSingleClassBox (ShapeView *v, Grafport *g, int x, int y, int w = BOX_WIDTH, int h = BOX_HEIGHT)
	SSDSingleClassBox (const SSDSingleClassBox &s)
	SSDSingleClassBox (const SSDDoubleClassBox &s)
	SSDSingleClassBox (const SSDTripleClassBox &s)
	~SSDSingleClassBox ()
Shape*	**Clone** ()

int **GetClassType** () const

void **SetSize** (int w, int h)

void **SetPosition** (const Point *p, bool snap)

void **SetFont** (XFont *ft)

void **SetTextColor** (const string *color)

void **SetGrafport** (Grafport *g)

void **SetView** (ShapeView *v)

TextShape*
ChooseTextShape (int x, int y)

bool **HasTextShape** (TextShape *t) const

void **AdjustSize** ()

void **UpdateShowStereotype** (bool b)

void **UpdateShowProperties** (bool b)

void **UpdateStereotypeLabelString** (const string *s)
update string of stereotype text shape and redraw.

const string*
GetStereotypeLabelString ()

TextShape*
GetStereotypeLabel () const

virtual void
UpdatePropertiesLabelString (const string *s)
update string of properties text shape and redraw.

const string*
GetPropertiesLabelString ()

TextShape*
GetPropertiesLabel () const

bool **SetAssocSubject** (AssocList *)

void **SetTextShape** ()

bool **HasString** (const string *s, bool sens, bool sub) const
contains s as string (case sensitive, substring).

bool **HasString** (const string*, bool, bool, List<TextShape *> *list) void **SetShowStereotype** (bool b)

void **SetShowProperties** (bool b)

bool **IsShowStereotype** () const

bool **IsShowProperties** () const

void **WriteMembers** (OutputFile *f)

bool **ReadMembers** (InputFile *f, double format)

Private Members

void **DrawTextShapes** ()

int **RequiredHeight** ()

int **RequiredWidth** ()

void **SetStereotypeLabelString** (const string *s)

void **SetPropertiesLabelString** (const string *s)

void **CalcPositionLabels** ()

void **CalcPositionStereotypeLabel** ()

void **CalcPositionPropertiesLabel** ()

static const int **TEXTMARGIN**

TextShape* **stereotypeLabel**

TextShape* **propertiesLabel**

bool **showStereotype**

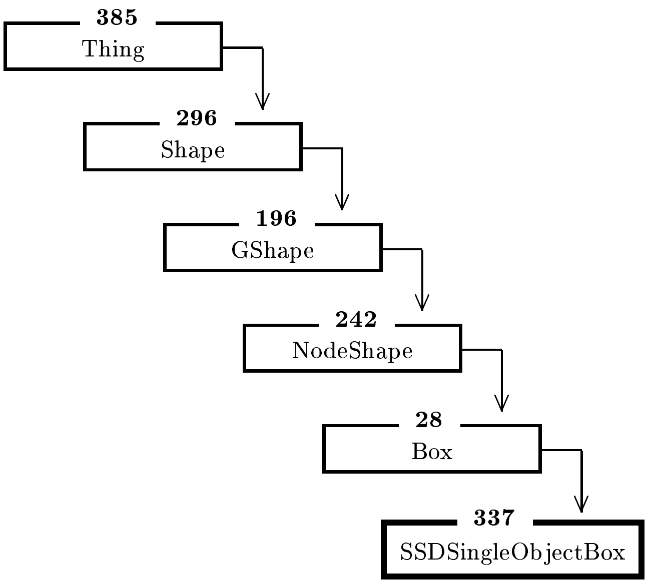
bool **showProperties**

337

class **SSDSingleObjectBox** : public Box

An object box used for an object

Inheritance



Public Members

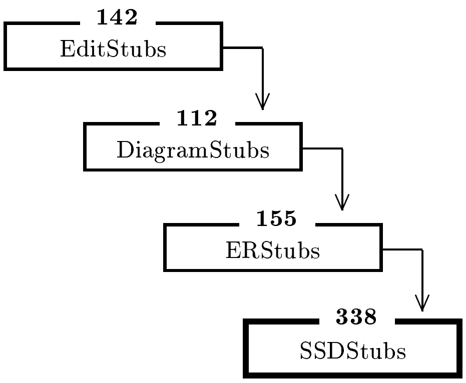
	SSDSingleObjectBox (ShapeView *v, Grafport *g, int x, int y, int w = BOX_WIDTH, int h = BOX_HEIGHT)
	SSDSingleObjectBox (const Box &s)
Shape*	**Clone** ()
int	**GetClassType** () const

338

class **SSDStubs** : public ERStubs

Specific stubs for a TSSD.

Inheritance



Public Members

static void
ShowStereotypeCB (Widget, XtPointer, XtPointer)

static void
HideStereotypeCB (Widget, XtPointer, XtPointer)

static void
ShowPropertiesCB (Widget, XtPointer, XtPointer)

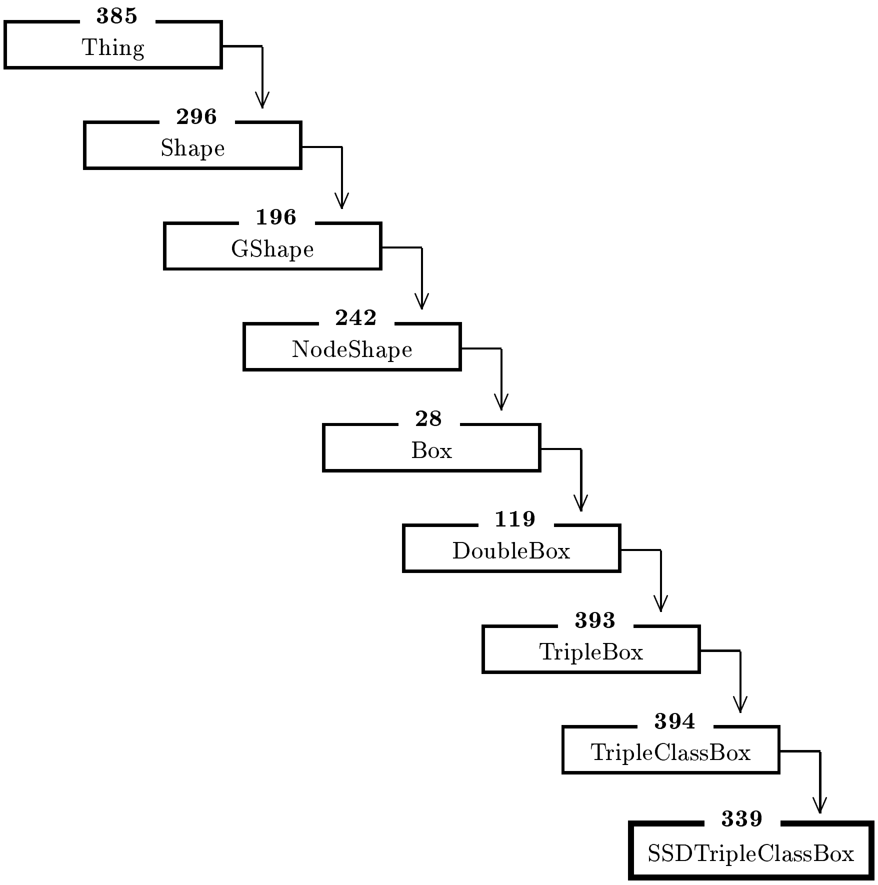
static void
HidePropertiesCB (Widget, XtPointer, XtPointer)

339

class **SSDTripleClassBox** : public TripleClassBox

A triple box used for a object class.

Inheritance



Public Members

SSDTripleClassBox (ShapeView *v, Grafport *g,
int x, int y,
int w = TRIPLECLASS-
BOX_WIDTH,
int h = TRIPLECLASS-
BOX_HEIGHT)

	SSDTripleClassBox (const SSDTripleClassBox &s)
	SSDTripleClassBox (const SSDDoubleClassBox &s)
	SSDTripleClassBox (const SSDSingleClassBox &s)
	~SSDTripleClassBox ()
Shape*	**Clone** ()
int	**GetClassType** () const
void	**SetFont** (XFont *ft)
void	**SetTextColor** (const string *color)
void	**SetGrafport** (Grafport *g)
void	**SetView** (ShapeView *v)
TextShape*	**ChooseTextShape** (int x, int y)
bool	**HasTextShape** (TextShape *t) const
void	**AdjustSize** ()
void	**UpdateShowStereotype** (bool b)
void	**UpdateShowProperties** (bool b)
void	**UpdateStereotypeLabelString** (const string *s) *update string of stereotype text shape and redraw.*
const string*	**GetStereotypeLabelString** ()
TextShape*	**GetStereotypeLabel** () const
void	**UpdatePropertiesLabelString** (const string *s) *update string of properties text shape and redraw.*
const string*	**GetPropertiesLabelString** ()
TextShape*	**GetPropertiesLabel** () const
void	**SetTextShape** ()
bool	**HasString** (const string *s, bool sens, bool sub) const

contains s as string (case sensitive, substring).

bool **HasString** (const string*, bool, bool, List<TextShape *> *list)

void **SetShowStereotype** (bool b)

void **SetShowProperties** (bool b)

bool **IsShowStereotype** () const

bool **IsShowProperties** () const

void **WriteMembers** (OutputFile *f)

bool **ReadMembers** (InputFile *f, double format)

Private Members

void **DrawTextShapes** ()

int **RequiredHeight** ()

int **RequiredWidth** ()

int **DoubleBoxHeight** ()

int **TripleBoxHeight** ()

void **SetStereotypeLabelString** (const string *s)

void **SetPropertiesLabelString** (const string *s)

void **CalcPositionLabels** ()

void **CalcPositionLabels2** ()

void **CalcPositionStereotypeLabel** ()

void **CalcPositionPropertiesLabel** ()

void **AdjustSizes** (const string *s)

TextShape* **stereotypeLabel**

TextShape* **propertiesLabel**

bool **showStereotype**

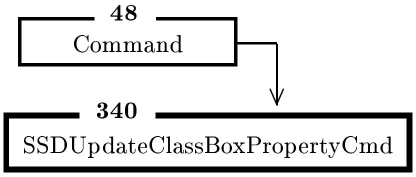
bool **showProperties**

340

class **SSDUpdateClassBoxPropertyCmd** : public Command

command class for updating SSD specific properies of shapes.

Inheritance



Public Members

enum **PropertyType** *boolean properties to change.*

SSDUpdateClassBoxPropertyCmd (ShapeView *view, PropertyType p, bool set)

virtual **~SSDUpdateClassBoxPropertyCmd** ()

void **Execute** ()

void **UnExecute** ()

const char* **GetName** () const

Private Members

void **ChangeProperties** ()

bool **set**

List <GShape *> * **shapes**

PropertyType **property**

ShapeView*

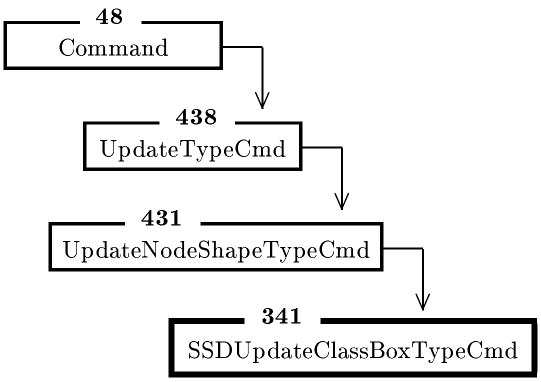
view

341

class **SSDUpdateClassBoxTypeCmd** : public UpdateNodeShapeTypeCmd

command for changing class box types

Inheritance



Public Members

SSDUpdateClassBoxTypeCmd (ShapeView *v, List<int *> *fromTypes, int toType)

const char* **GetName** () const

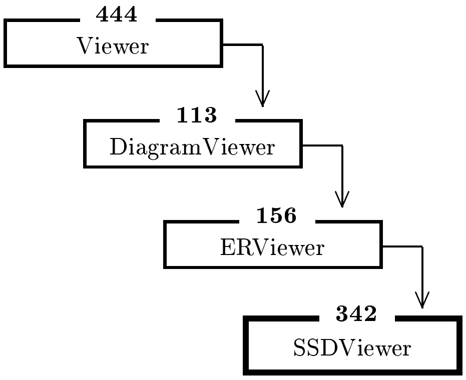
Thing* **Transform** (Thing *thing)

342

class **SSDViewer** : public ERViewer

class-relationship viewer class.

Inheritance



Public Members

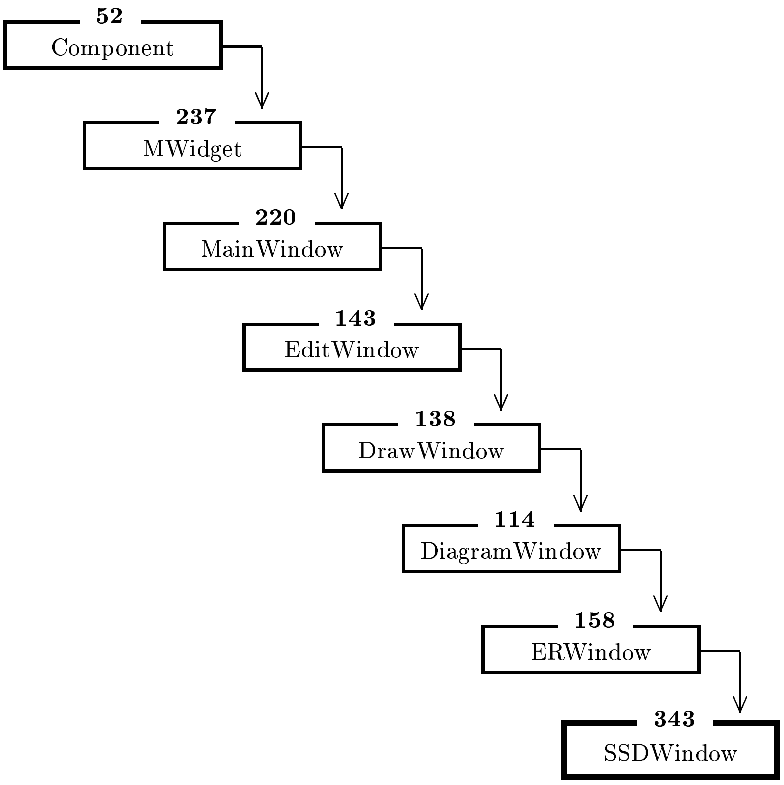
	SSDViewer (Config *c, SSDWindow * w)	
	SSDViewer (Config *c, ESDWindow * w)	
void	**UpdateNodeShapeType** (int)	*change box type of selected object types.*
void	**UpdateShowStereotypes** (bool)	*update show stereotypes type of selected object types.*
void	**UpdateShowProperties** (bool)	*update show properties type of selected object types.*

343

class **SSDWindow** : public ERWindow

class-relationship edit window class.

Inheritance



Public Members

	SSDWindow (const char *name)
virtual	**~SSDWindow** ()

Private Members

int	**GetTool** ()

DiagramViewer*
CreateViewer (Config *config)

Graph* **CreateGraph** ()

Diagram* **CreateDiagram** (Config *c, DiagramViewer *v, Graph *g)

void **DetermineIcons** ()

MenuItem*
diagramSSDBoxItems

MenuItem*
diagramSSDClassStereotypeItems

MenuItem*
diagramSSDClassPropertiesItems

void **InitMenuItems** ()

static const char*
SSD_NODE_NAMES []

static const Bitmap
SSD_NODE_ICONS []

static const char*
SSD_EDGE_NAMES []

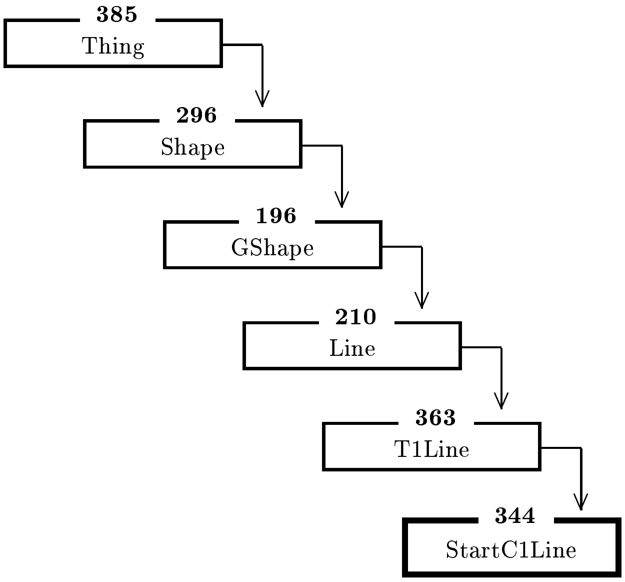
static const Bitmap
SSD_EDGE_ICONS []

344

class **StartC1Line** : public T1Line

line having card constraint at 'from' side, name is not editable (SND).

Inheritance



Public Members

StartC1Line (ShapeView *v, Grafport *g, GShape *n1, GShape *n2, List<Point *> *line, bool Curved)

StartC1Line (ShapeView *v, Grafport *g, GShape *n1, GShape *n2, bool Curved)

Shape* **Clone** ()

bool **SetAssocSubject** (AssocList *)

void **SetTextShape** ()

Private Members

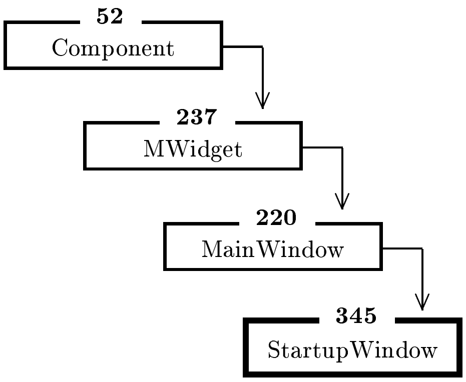
void **Initialize** ()

345

class **StartupWindow** : public MainWindow

Main window of the startup program.

Inheritance



Public Members

	StartupWindow (const char *name)
virtual	**~StartupWindow** ()
const char*	**GetClassName** ()
void	**Quit** ()
void	**Project** ()
void	**OpenDocument** ()
void	**DrawingArea** ()
void	**StartEditor** (int editor, const string *filename = 0)
void	**StartTableEditor** (unsigned rows, unsigned columns, unsigned cellwidth, unsigned cellheight)
void	**SetDrawingArea** (unsigned idw, unsigned idh, unsigned mdw, unsigned mdh)

void **SetProject** (const string *s)

void **OpenDocument** (const string *s)

void **SetStatus** (const char *s)

void **SetStatus** (const string *s)

FileSelectionDialog*
GetFileSelectionDialog ()

FileSelectionDialog*
GetDirSelectionDialog ()

TextFieldListDialog*
GetTableOptionsDialog ()

TextFieldListDialog*
GetDrawingOptionsDialog ()

static void
QuitCB (Widget, XtPointer, XtPointer)

static void
HelpCB (Widget, XtPointer, XtPointer)

static void
ProjectCB (Widget, XtPointer, XtPointer)

static void
DocumentCB (Widget, XtPointer, XtPointer)

static void
OpenDocumentCB (Widget, XtPointer, XtPointer)

static void
DrawingAreaCB (Widget, XtPointer, XtPointer)

static void
StartEditorCB (Widget, XtPointer, XtPointer)

static void
StartTableEditorCB (Widget, XtPointer, XtPointer)

static void
SetProjectCB (Widget, XtPointer, XtPointer)

static void
SetDrawingAreaCB (Widget, XtPointer, XtPointer)

Private Members

Config* **config**

Helper* **helper**

Widget **statusBar**

FileSelectionDialog* **fileSelectionDialog**

FileSelectionDialog* **dirSelectionDialog**

TextFieldListDialog* **tableOptionsDialog**

TextFieldListDialog* **drawingOptionsDialog**

string **projDir**

unsigned **initDrawingWidth**

unsigned **initDrawingHeight**

unsigned **maxDrawingWidth**

unsigned **maxDrawingHeight**

string **binDir** *the directory where the binaries are.*

void **InitMenuItems** ()

Widget **CreateWorkArea** (Widget w, int argc, char **argv)

void **ParseUserOptions** (int argc, char **argv)

void **CreateComponents** (Widget parent)

Widget **CreateMenuBar** (Widget parent)

Menu* **fileMenu**

Menu* **viewMenu**

Menu* **helpMenu**

MenuItem* **fileItems**

MenuItem* **viewItems**

MenuItem*

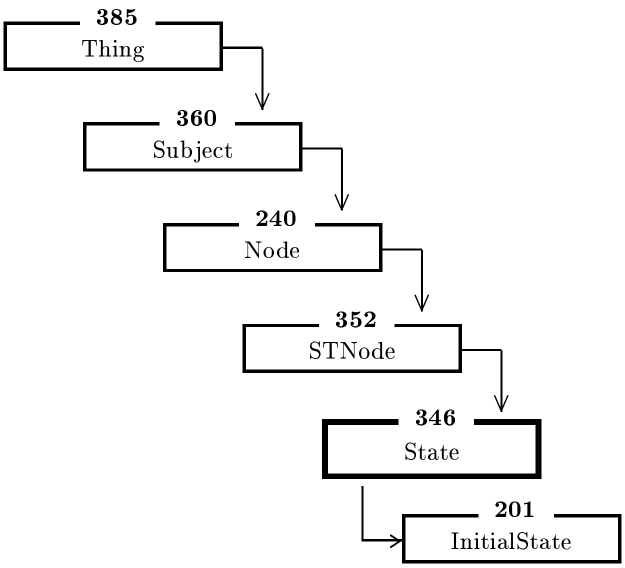
	helpItems
void	**AddToolEntry** (Widget parent, int i)
void	**StartDocumentEditor** (int editor, const string *filename = 0)
void	**ShowTableDialog** (int editor)
void	**AppendExec** (string *s, int editor)
void	**AppendProjDir** (string *s)
void	**AppendDrawingArea** (string *s)
void	**ExecCommand** (string *s)
void	**InitTextFields** ()
void	**Finalize** ()
List <string *>	**tableFields**
List <string *>	**tableDefaults**
List <string *>	**fetDefaults**
List <string *>	**tutDefaults**
List <string *>	**drawingFields**
List <string *>	**drawingDefaults**
int	**tmpEditor**
static const int	**MIN_DRAWING_SIZE**
string	**welcomeLabel**
Widget	**mainForm**
Widget	**scrolledWindow**

346

class **State** : public STNode

state (STD) node class.

Inheritance



Public Members

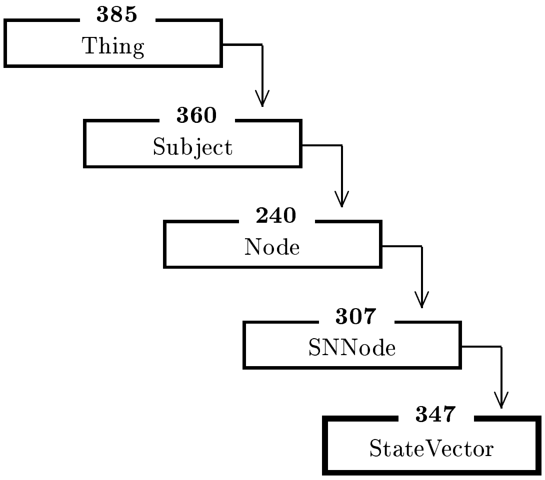
	State (STGraph *g)
int	**GetClassType** () const
Subject*	**Clone** ()

347

class **StateVector** : public SNNode

system network state vector node class.

Inheritance



Public Members

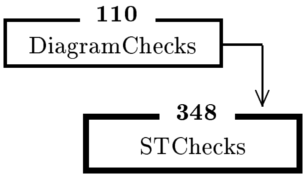
	StateVector (SNGraph *g)
int	**GetClassType** () const
Subject*	**Clone** ()

348

class **STChecks** : public DiagramChecks

implements generic checks for soft constraints on ST diagrams.

Inheritance



Public Members

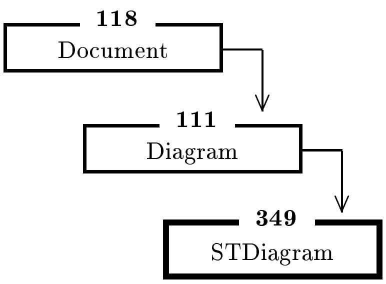
	STChecks (Diagram *d, Graph *g)
unsigned	**CheckUnreachableStates** (string &chkbuf)
unsigned	**CheckEmptyEvents** (string &chkbuf)
unsigned	**CheckEmptyActions** (string &chkbuf)
unsigned	**CheckNoActions** (string &chkbuf)
unsigned	**CheckDoubleEvents** (string &chkbuf)

349

class **STDiagram** : public Diagram

state transition diagram class

Inheritance



Public Members

	STDiagram (Config *, STWindow *, STViewer *, STGraph *)
virtual	**~STDiagram** ()
Node*	**CreateNode** ()
Edge*	**CreateEdge** (Subject *n1, Subject *n2)
NodeShape*	**CreateNodeShape** (Node *node, int x, int y)
Line*	**CreateLine** (Edge *edge, GShape *fromShape, GShape *toShape, List<Point *> *line)
void	**UpdateNodeType** (int n)
void	**UpdateEdgeType** (int n)
void	**CheckDocument** ()
bool	**SetEvent** (Transition *t, const string *s)
bool	**SetAction** (Subject *t, const string *s, unsigned nr)
bool	**SetText** (TextShape *t, const string *s)

Protected Members

Thing* **CreateThing** (int classNr)

bool **CheckEdgeConstraints** (Subject *s1, Subject *s2)

InitialState*
FindInitialState (Subject *s)

Private Members

STChecks* **stChecks**

static const int
BOX_WIDTH *width and hight of boxes.*

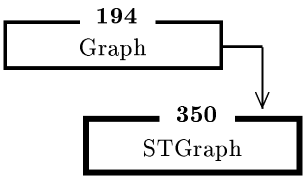
static const int
BOX_HEIGHT

350

class **STGraph** : public Graph

state transition graph class.

Inheritance



Public Members

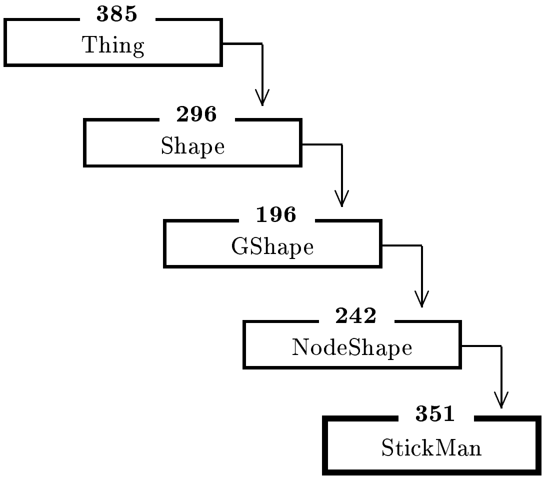
	STGraph ()
void	**InitConnections** ()

351

class **StickMan** : public NodeShape

StickMan shape class.

Inheritance



Public Members

static const int
STICKMAN_WIDTH
default width.

static const int
STICKMAN_HEIGHT
default height.

StickMan (ShapeView *v, Grafport *g, int x, int y, int w = STICKMAN_WIDTH, int h = STICKMAN_HEIGHT)

StickMan (const NodeShape &c)

Shape* **Clone** ()

void **SetPosition** (const Point *pos, bool snap=True)

void **SetSize** (int w, int h)

void **SetTextShape** ()

void	**AdjustSize** ()
void	**DrawOutLine** (const Point *center)
void	**DrawDuplicate** () *the asterisk has a slightly different position.*
Point	**GiveSnp** (Point to) const
Point	**GiveSnp** (Point p, int count, int order, int &code)
int	**GetClassType** () const
static const int	**STICKMAN_HANDLES** *default nr. of handles.*

Protected Members

void	**CalcPositionLabels** ()
void	**DrawShape** ()
void	**DrawTextShapes** ()
void	**DrawStickMan** (int x, int y, int wd, int ht)
void	**FillStickMan** (int x, int y, int wd, int ht)

Private Members

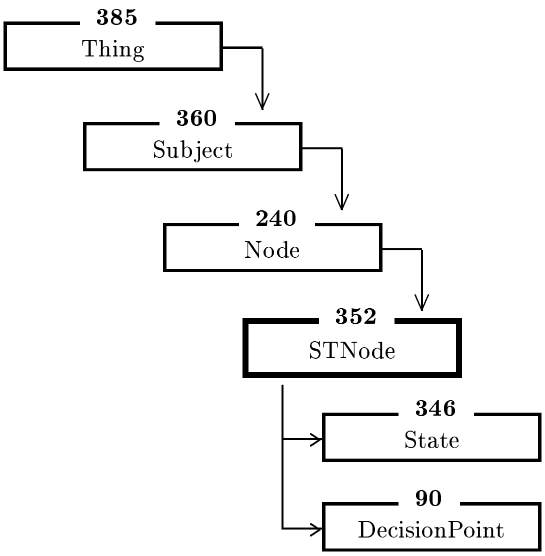
int	**RequiredHeight** ()
int	**TextHeight** ()

352

class **STNode** : public Node

A node in a STD.

Inheritance



Public Members

STNode (STGraph *g)

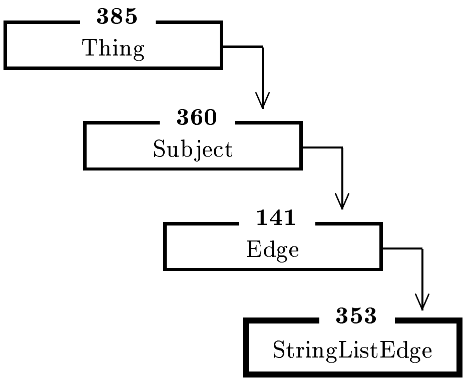
Subject::NameErrType
SetName (const string *name)

353

class **StringListEdge** : public Edge

An edge that has an extra list of strings.

Inheritance



Public Members

	StringListEdge (Graph *g, Subject *n1, Subject *n2)
	StringListEdge (const StringListEdge &s)
virtual	**~StringListEdge** ()
enum	**StringErrorType**
virtual StringListEdge::StringErrorType	**SetString** (const string *s, unsigned n, bool update)
const string*	**GetString** (unsigned n)
unsigned	**NrStrings** ()
bool	**HasString** (const string *s)
const string*	**GetStringListLabel** ()
const string*	

GetStringLabel ()

const List <string *> *
GetStringList ()

void **SetStringListLabel** (const string *s)

void **SetStringLabel** (const string *s)

void **WriteMembers** (OutputFile *f)

bool **ReadMembers** (InputFile *f, double format)

Private Members

List <string *> *
stringList

string **stringListLabel**

string **stringLabel**

354

class **StringList**

utility for updating a list of strings.

Public Members

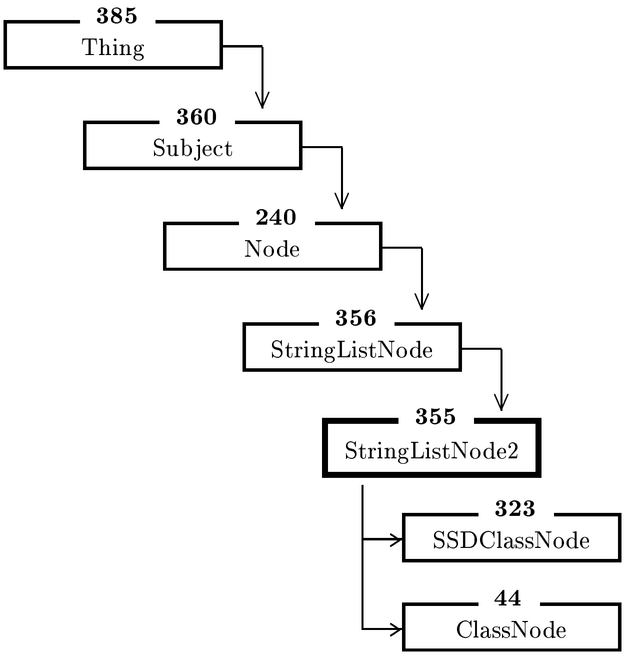
static void
Insert (List<string *> *list, const string *s, unsigned n, bool update)
Set string s in list at pos. When update = true s[n] is overwritten.

355

class **StringListNode2** : public StringListNode

A node that has two extra lists of strings.

Inheritance



Public Members

StringListNode2 (Graph *g)

StringListNode2 (const StringListNode2 &s)

virtual **~StringListNode2** ()

virtual StringListNode::StringErrorType
SetString2 (const string *s, unsigned n, bool update)

const string*
GetString2 (unsigned n)

unsigned **NrStrings2** ()

const string*
GetStringListLabel2 ()

const string*
GetStringLabel2 ()

void **SetStringListLabel2** (const string *s)

void **SetStringListLabel2** (const char *s)

void **SetStringLabel2** (const string *s)

void **SetStringLabel2** (const char *s)

bool **HasString2** (const string *s)

void **WriteMembers** (OutputFile *f)

bool **ReadMembers** (InputFile *f, double format)

Protected Members

List <string *> *
GetStringList2 ()

Private Members

List <string *> *
stringList2

string **stringListLabel2**

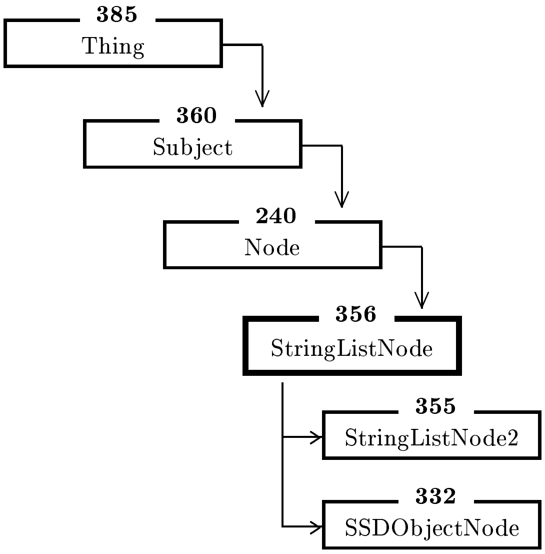
string **stringLabel2**

356

class **StringListNode** : public Node

A node that has an extra list of strings.

Inheritance



Public Members

StringListNode (Graph *g)

StringListNode (const StringListNode &s)

virtual **~StringListNode** ()

enum **StringErrorType**

virtual StringListNode::StringErrorType
SetString (const string *s, unsigned n, bool update)

const string*
GetString (unsigned n)

unsigned **NrStrings** ()

const string*

	GetStringListLabel ()
const string*	**GetStringLabel** ()
void	**SetStringListLabel** (const string *s)
void	**SetStringListLabel** (const char *s)
void	**SetStringLabel** (const string *s)
void	**SetStringLabel** (const char *s)
bool	**HasString** (const string *s)
void	**WriteMembers** (OutputFile *f)
bool	**ReadMembers** (InputFile *f, double format)

Protected Members

List <string *> *	**GetStringList** ()

Private Members

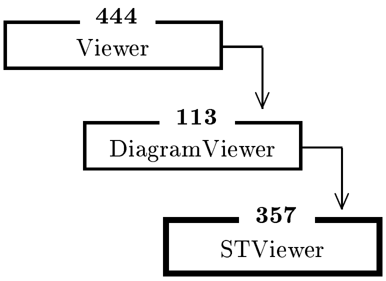
List <string *> *	**stringList**
string	**stringListLabel**
string	**stringLabel**

357

class **STViewer** : public DiagramViewer

state transition viewer class

Inheritance



Public Members

STViewer (Config *c, STWindow *w)

virtual Command*
Drag (int x, int y)

Protected Members

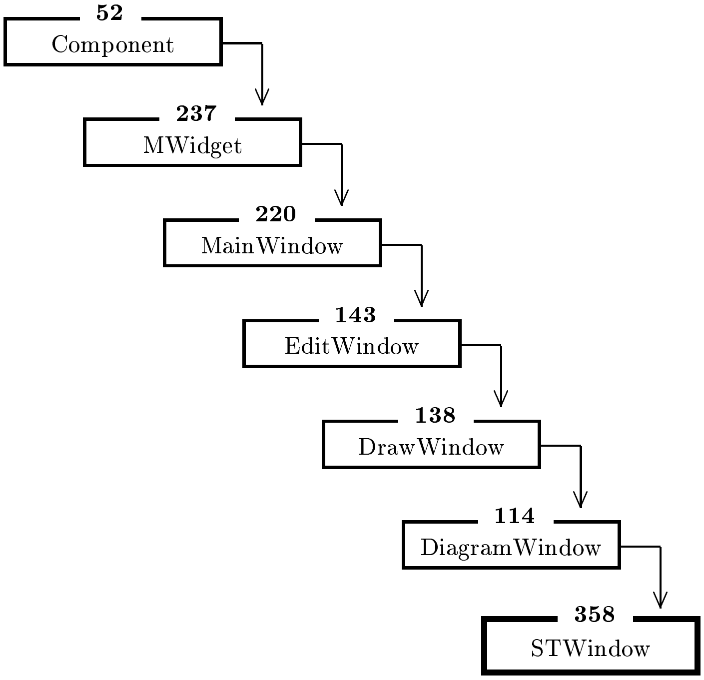
TransitionArrow*
HitHLine (int x, int y)

358

class **STWindow** : public DiagramWindow

state transition window class.

Inheritance



Public Members

STWindow (const char *name)

Private Members

int **GetTool** ()

DiagramViewer* **CreateViewer** (Config *c)

Graph* **CreateGraph** ()

Diagram* **CreateDiagram** (Config *c, DiagramViewer *v, Graph *g)

void **DetermineIcons** ()

static const char* **ST_NODE_NAMES** []

static const Bitmap **ST_NODE_ICONS** []

static const char* **ST_EDGE_NAMES** []

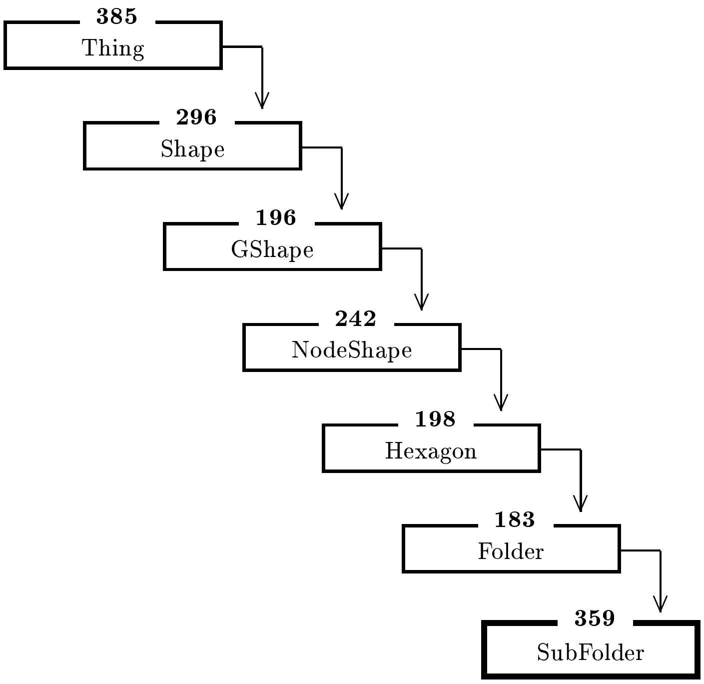
static const Bitmap **ST_EDGE_ICONS** []

359

class **SubFolder** : public Folder

subfolder drawn as folder with additional fork in tab

Inheritance



Public Members

	SubFolder (ShapeView *v, Grafport *g, int x, int y)
	SubFolder (const NodeShape &c)
Shape*	**Clone** ()
int	**GetClassType** () const

Protected Members

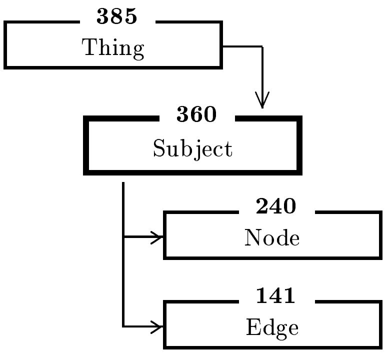
void **DrawShape** () *draws a subfolder*

360

class **Subject** : public Thing

(abstract) graph element class (is either a node or an edge).

Inheritance



Public Members

	Subject (Graph *g)	
virtual Subject*	**Clone** ()	
bool	**IsSubject** () const	*it is a subject*
virtual bool	**IsEdge** () const	*is it a node or an edge?*
virtual bool	**InGraph** () const	*return if subject is part of the graph.*
enum	**NameErrType**	*The reasons why setting the subject name might fail (error codes)*
const string*	**GetName** () const	
virtual NameErrType	**SetName** (const string *s)	*set and check the name*
void	**WriteMembers** (OutputFile *f)	

		write members of subject to file.
bool	**ReadMembers** (InputFile *f, double format)	*read and parse members of subject from file.*
bool	**SetReferences** (AssocList *)	*set subject references using the assoclist.*
bool	**CheckReferences** ()	*Check if parent reference exists as graph node element.*
void	**SetParent** (Node *p)	*Set the parent node (parent may be 0).*
Node*	**GetParent** () const	
void	**SetAnnotation** (string *s)	
const string*	**GetAnnotation** () const	
Graph*	**GetGraph** () const	

Private Members

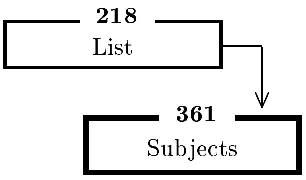
string	**name**	*The name of the subject.*
string	**annotation**	*free text as annotation.*
Graph*	**graph**	*The graph in which the subject is a node or edge.*
Node*	**parent**	*Parent node (parent = 0 when subject has no parent).*

361

class **Subjects** : public List<Subject *>

list of subjects class; copied subjects are put in the assoc list.

Inheritance



Public Members

	Subjects (AssocList *al)
	Subjects (const Subjects &subjects)
AssocList*	**GetAssocList** () const

Private Members

Subject*	**CloneSubject** (Subject *subject)
AssocList*	**assocList**

362

class **System**

Encapsulates some O.S. specific functions.

Public Members

static void
SetSignalHandler ()
set unix signal handler.

static bool
GetHostName (char* host)
copies into host the hostname of the machine.

static bool
GetHostName (string* host)

static bool
GetLoginName (char* login)
copies into login the loginname of the user.

static bool
GetLoginName (string* login)

static bool
GetTime (char* time)
copies in time the current machine time as a string.

static bool
GetTime (string* time)

static unsigned long
GenerateId (unsigned long)
returns a random number based on the time and the given number.

static bool
GetPrinter (char* printer)
copies into printer the PRINTER environment variable.

static bool
GetPrinter (string* printer)

static bool

GetCwd (char* dir)
retrieve current working directory.

static bool
GetCwd (string* dir)

static bool
GetHome (char* home)
retrieve home directory.

static bool
GetHome (string* home)

static void
GetToolkitHome (char* home)
copies in home the tool home directory variable.

static void
GetToolkitHome (string* home)

static void
GiveFile (const char* path, char* dir, char* file)
copies in dir and file the directory and file in the name 'path'.

static void
GiveFile (const string* path, string* dir, string* file)

static void
GiveFile (const char* path, char* file)
copies in file the file in the name 'path'.

static void
GiveFile (const string* path, string* file)

static bool
FileRegular (const char* file)
returns if file is regular (not directory, named pipe etc).

static bool
FileExists (const char* file)
returns whether file exists

static bool

FileExecutable (const char* file)
returns whether executable file exists

static bool
DirExists (const char* dir)
returns whether directory exists

static bool
FindProgram (char* program, const char* command)
returns if there is command in a std dir, program becomes the path.

static bool
FindProgram (string* program, const char* command)

static bool
NxnParse (const char* desc, int &num1, int &num2)
return true if desc is a 'num1'x'num2' string.

static bool
FindArg (char** argv, int argc, const char* arg, int &index)
return if argv contains arg. Index will be the position in argv.

static void
SendEmail (const char* recipient, const char* subject, const char* program, const char* version, const char* text)
send an email message (for instance for stats or bug reports).

static const char
TOOLKIT_HOME_DEFAULT []
default toolkit home directory.

static const char
TOOLKIT_STAT_HEADER []
Statistics message subject for logging.

static const char

TOOLKIT_BUG_HEADER []
Bug report message subject.

static const char
TOOLKIT_HOME_VAR_NAME []
Variable name that contains the tool home directory.

static const char*
STD_SEARCH_DIRS []
most common Unix directories for system wide available binaries.

static bool
sendBugEmail *send an email after crash*

static char
emailAddress [] *email address to send to.*

static char
currentProgram []
the command name of the current program (argv[0]).

Protected Members

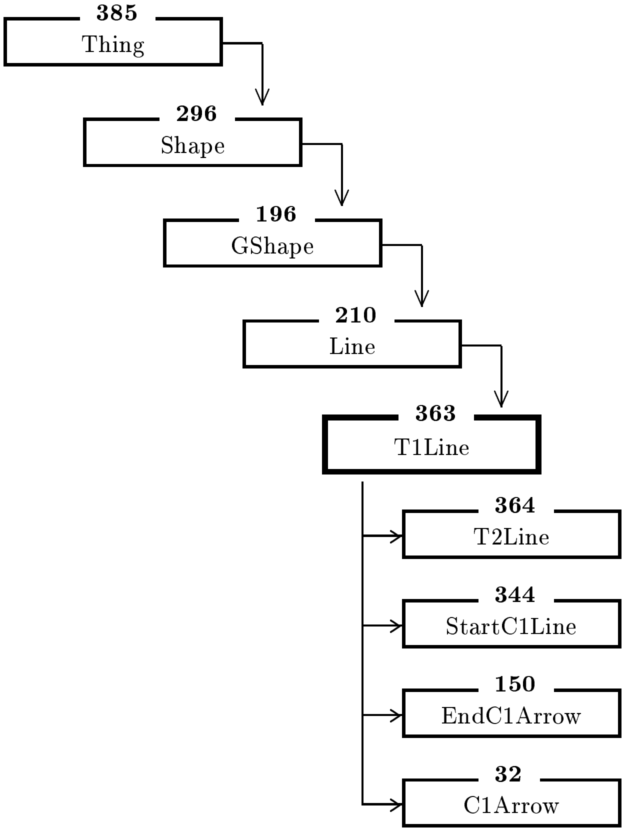
static void
SignalHandler (int s)
called when signal occurs.

363

class **T1Line** : public Line

line class (abstract) having an extra text shape.

Inheritance



Public Members

T1Line (ShapeView *v, Grafport *g, GShape *n1, GShape *n2, List<Point *> *line, bool Curved)

T1Line (ShapeView *v, Grafport *g, GShape *n1, GShape *n2, bool Curved)

	T1Line (const T1Line &aline)
virtual	**~T1Line** ()
int	**GetClassType** () const
void	**CalcPosition** ()
void	**UpdateTextShape1** (const string *s) *calls corresponding members of textShape1.*
void	**SetSelect** (bool s)
void	**SetFont** (XFont *ft)
void	**SetTextColor** (const string *color)
int	**GetLeftMost** () const *as line, but the extra textshape is also considered.*
int	**GetTopMost** () const
int	**GetRightMost** () const
int	**GetBottomMost** () const
void	**WriteMembers** (OutputFile *f)
bool	**ReadMembers** (InputFile *f, double format)
enum	**PositionType**
void	**SetGrafport** (Grafport *g)
void	**SetView** (ShapeView *v)
bool	**InTextArea** (int x, int y) *return if (x, y) in text-area of in textshape.*
TextShape*	**HitTextShape** (int x, int y) *returns name or extra textshape if hit.*
bool	**HasTextShape** (TextShape *t) const
bool	**HasString** (const string *s, bool sens, bool sub) const
bool	**HasNameString** (const string *, bool, bool) const

int **ReplaceNameString** (const string *, const string *, bool, bool)

void **SetTextShape** ()

void **DrawShape** () *draw line+textshapes.*

TextShape* **ChooseTextShape** (int x, int y) *returns textshape if hit or if empty, otherwise return name.*

void **MoveRaw** (const Point *delta)

Protected Members

static const int **CL_FRACTION** *fraction of line for the extra textshapes.*

virtual void **CalcPositionTextShapes** () *calculate position of textshape on first line segment.*

void **CalcPositionTextShape** (TextShape *t, PositionType tpos)

bool **HitTextShapeArea** (TextShape *t, PositionType tpos, int x, int y)

bool **HitExtraTextShapeArea** (int x, int y, const Point *from, const Point *to) *returns of (x, y) hits area for extra textshapes.*

TextShape* **GetTextShape1** () const

void **SetTextShape1** (TextShape *t)

PositionType **GetT1Position** ()

void **SetT1Position** (PositionType t)

Private Members

TextShape*
textShape1

PositionType
t1Position

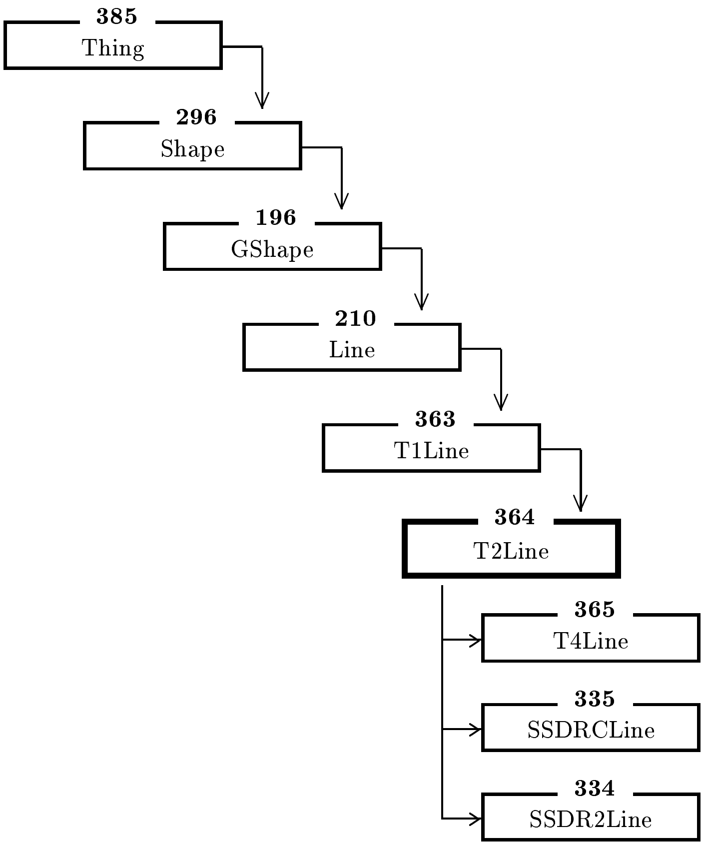
virtual void
InitTextShapes ()

364

class **T2Line** : public T1Line

line class (abstract) having two extra text shape.

Inheritance



Public Members

T2Line (ShapeView *v, Grafport *g, GShape *n1, GShape *n2, List<Point *> *line, bool Curved)

T2Line (ShapeView *v, Grafport *g, GShape *n1, GShape *n2, bool Curved)

	T2Line (const T2Line &)
virtual	**~T2Line** ()
int	**GetClassType** () const
void	**UpdateTextShape2** (const string *s)
void	**SetSelect** (bool s)
void	**SetFont** (XFont *ft)
void	**SetTextColor** (const string *color)
int	**GetLeftMost** () const *as line, but the extra text shape is also considered.*
int	**GetTopMost** () const
int	**GetRightMost** () const
int	**GetBottomMost** () const
void	**WriteMembers** (OutputFile *f)
bool	**ReadMembers** (InputFile *f, double format)
void	**SetGrafport** (Grafport *g)
void	**SetView** (ShapeView *v)
bool	**InTextArea** (int x, int y) *return if (x, y) in text-area or in text shapes.*
TextShape*	**HitTextShape** (int x, int y) *returns name or extra text shape if hit.*
TextShape*	**ChooseTextShape** (int x, int y) *returns text shape if hit or if empty, otherwise return name.*
bool	**HasTextShape** (TextShape *t) const
bool	**HasString** (const string *s, bool sens, bool sub) const
bool	**HasNameString** (const string *, bool, bool) const

int **ReplaceNameString** (const string *, const string *, bool, bool)

void **SetTextShape** ()

void **DrawShape** () *draw line+textshapes.*

void **MoveRaw** (const Point *delta)

Protected Members

virtual void **CalcPositionTextShapes** () *calculate position of textshape on first line segment.*

TextShape* **GetTextShape2** () const

void **SetTextShape2** (TextShape *t)

PositionType **GetT2Position** ()

void **SetT2Position** (PositionType t)

Private Members

TextShape* **textShape2**

PositionType **t2Position**

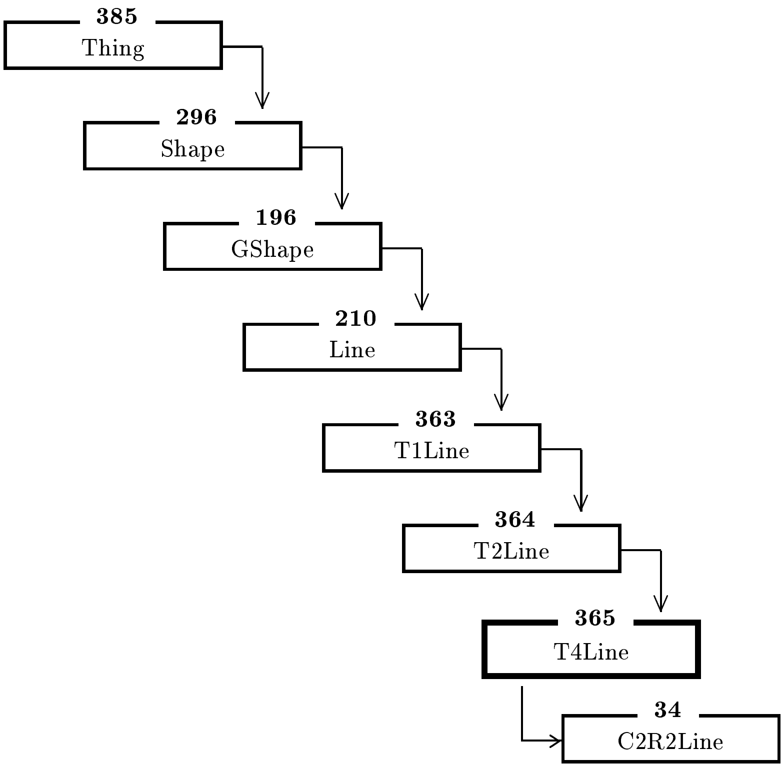
void **InitTextShapes** ()

365

class **T4Line** : public T2Line

line having 4 text shapes at 4 positions.

Inheritance



Public Members

T4Line (ShapeView *v, Grafport *g, GShape *n1, GShape *n2, List<Point *> *line, bool Curved)

T4Line (ShapeView *v, Grafport *g, GShape *n1, GShape *n2, bool Curved)

T4Line (const T4Line &)

virtual	**~T4Line** ()
int	**GetClassType** () const
bool	**InTextArea** (int x, int y) *see T2Line. Holds all text shapes into account.*
TextShape*	**HitTextShape** (int x, int y)
TextShape*	**ChooseTextShape** (int x, int y)
bool	**HasTextShape** (TextShape *t) const
void	**SetSelect** (bool s)
void	**UpdateTextShape3** (const string *s)
void	**UpdateTextShape4** (const string *s)
int	**GetLeftMost** () const *as line, but all text shapes are considered*
int	**GetTopMost** () const
int	**GetRightMost** () const
int	**GetBottomMost** () const
void	**WriteMembers** (OutputFile *f)
bool	**ReadMembers** (InputFile *f, double format)
void	**SetTextShape** ()
void	**SetFont** (XFont *ft)
void	**SetTextColor** (const string *color)
void	**SetAlignment** (TextAlign::Type alignment)
void	**SetGrafport** (Grafport *g)
void	**SetView** (ShapeView *v)
bool	**HasString** (const string *s, bool sens, bool sub) const
bool	**HasString** (const string*, bool, bool, List<TextShape *> *)
void	**MoveRaw** (const Point *delta)

Protected Members

void **CalcPositionTextShapes** () *calculate position of all text shapes.*

TextShape* **GetTextShape3** () const

void **SetTextShape3** (TextShape *t)

TextShape* **GetTextShape4** () const

void **SetTextShape4** (TextShape *t)

PositionType **GetT3Position** ()

void **SetT3Position** (PositionType t)

PositionType **GetT4Position** ()

void **SetT4Position** (PositionType t)

void **DrawShape** () *draw line+ 4 other text shapes.*

Private Members

TextShape* **textShape3**

TextShape* **textShape4**

PositionType **t3Position**

PositionType **t4Position**

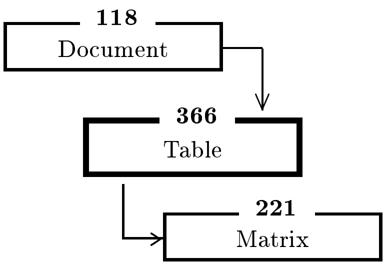
void **InitTextShapes** ()

366

class **Table** : public Document

Table class.

Inheritance



Public Members

	Table (Config *c, TableWindow *w, TableViewer *v)
virtual	**~Table** ()
void	**Initialize** ()
void	**LoadEntries** ()
void	**SaveEntries** ()
void	**RemoveAll** ()
void	**AddRows** ()
void	**AddColumns** ()
void	**DeleteRows** ()
void	**DeleteColumns** ()
void	**DeleteAll** ()
void	**Purge** ()
void	**CutTexts** ()
void	**CopyTexts** ()
void	**PasteTexts** ()
void	**SortRows** ()

void **SortColumns** ()

void **SameWidth** ()

void **SameHeight** ()

void **Span** ()

void **UnSpan** ()

void **SelectAll** ()

void **Move** (MoveType t)
Move table on mainwindow according to move type.

void **Append** ()

void **SetAppendToggle** (bool set)
set append to bottom(Trye) or to the right(False).

bool **IsAppendBottom** ()

void **SetAddRowsToggle** (bool set)
called via add rows dialog.

void **SetAddColumnsToggle** (bool set)

void **AddRows** (unsigned count)

void **AddColumns** (unsigned count)

void **FindAll** (const string *s, bool sens, bool sub, bool)

void **FindNext** (const string *s, bool sens, bool sub, bool)

void **ReplaceNext** (const string *s1, const string *s2, bool sens, bool sub, bool)

void **ReplaceAll** (const string *s1, const string *s2, bool sens, bool sub, bool)

virtual bool
UpdateText (Cell *c, const string *text)

void **SetDefaultNumberOfRows** ()

void **SetDefaultNumberOfColumns** ()

void **SetDefaultRowHeight** ()

void **SetDefaultColumnWidth** ()

void	**SetDefaultNumberOfRows** (unsigned n)
void	**SetDefaultNumberOfColumns** (unsigned n)
void	**SetDefaultRowHeight** (int n)
void	**SetDefaultColumnWidth** (int n)
void	**SetMarginHeight** (int n)
void	**SetMarginWidth** (int n)
void	**DefaultRowAlignment** ()
void	**DefaultColumnAlignment** ()
void	**UpdateRowAlignment** ()
void	**UpdateColumnAlignment** ()
void	**UpdateColumnAlignment** (TextAlign::Type type)
void	**UpdateRowAlignment** (TextAlign::Type type)
void	**UpdateDefaultColumnAlignment** (TextAlign::Type newType)
void	**UpdateDefaultRowAlignment** (TextAlign::Type newType)
void	**ShowConfigRowAlignment** ()
void	**ShowDefaultRowAlignment** ()
void	**ShowConfigColumnAlignment** ()
void	**ShowDefaultColumnAlignment** ()
void	**UpdateTableLineStyle** ()
void	**UpdateLineStyle** (Side::Type side, LineStyle::Type style)
void	**ShowConfigLineStyle** ()
void	**ShowDefaultLineStyle** ()
void	**UpdateTableLineWidth** ()
void	**UpdateLineWidth** (Side::Type side, unsigned width)
void	**ShowConfigLineWidth** ()
void	**ShowDefaultLineWidth** ()

void **UpdateCellTextUnderline** ()

void **ShowRowColumnLabels** (bool b)

void **PrintRowColumnLabels** (bool b)

void **UpdateMarginWidth** ()

void **UpdateMarginHeight** ()

void **AnnotateCell** ()

void **SetCellAnnotation** (const string *s)

ScaleDialog*
GetScaleDialog ()

ToggleListDialog*
GetPromptToggleDialog ()

LineStyleSideDialog*
GetTableLineStyleDialog ()

LineWidthSideDialog*
GetTableLineWidthDialog ()

TextAlignmentDialog*
GetRowAlignmentDialog ()

NToggleListDialog*
GetColumnAlignmentDialog ()

static const int
MIN_ROW_HEIGHT

static const int
MIN_COLUMN_WIDTH

static const int
MAX_ROW_HEIGHT

static const int
MAX_COLUMN_WIDTH

static const int
MIN_MARGIN_WIDTH

static const int
MIN_MARGIN_HEIGHT

static const int
MAX_MARGIN_WIDTH

static const int
MAX_MARGIN_HEIGHT

static const int

MIN_NUMBER_OF_COLUMNS

static const int
MIN_NUMBER_OF_ROWS

static const int
MAX_NUMBER_OF_COLUMNS

static const int
MAX_NUMBER_OF_ROWS

Protected Members

TableViewer*
GetTableViewer ()

Private Members

TableViewer*
tviewer

bool **topAdd**

bool **leftAdd**

bool **appendBottom**

ScaleDialog*
scaleDialog

ToggleListDialog*
promptToggleDialog

LineStyleSideDialog*
tableLineStyleDialog

LineWidthSideDialog*
tableLineWidthDialog

TextAlignmentDialog*
rowAlignmentDialog

NToggleListDialog*
columnAlignmentDialog

unsigned **defaultNumberOfRows**

unsigned **defaultNumberOfColumns**

int **defaultRowHeight**

int **defaultColumnWidth**

int	**defaultMarginWidth**
int	**defaultMarginHeight**
Cell*	**annotationCell**
static const int	**DEFAULT_ADD_ROWS**
static const int	**DEFAULT_ADD_COLUMNS**
void	**FitTable** ()
void	**DeleteAllCells** ()
void	**AnnotateCell** (Cell *cell)
void	**EmptySelectionMessage** ()
void	**EmptyTableMessage** ()

367

class **TableStubs** : public EditStubs

Calback stubs for table editor menu items and popup dialogs.

Inheritance

142
EditStubs

367
TableStubs

Public Members

static void
AppendCB (Widget, XtPointer, XtPointer)

static void
AppendToggleCB (Widget, XtPointer, XtPointer)

static void
AddRowsOKCB (Widget, XtPointer, XtPointer)

static void
AddColumnsOKCB (Widget, XtPointer, XtPointer)

static void
AddRowsToggleCB (Widget, XtPointer, XtPointer)

static void
AddColumnsToggleCB (Widget, XtPointer, XtPointer)

static void
NumberOfRowsCB (Widget, XtPointer, XtPointer)

static void

NumberOfColumnsCB (Widget, XtPointer, XtPointer)

static void
RowHeightCB (Widget, XtPointer, XtPointer)

static void
ColumnWidthCB (Widget, XtPointer, XtPointer)

static void
MarginHeightCB (Widget, XtPointer, XtPointer)

static void
MarginWidthCB (Widget, XtPointer, XtPointer)

static void
AddRowsCB (Widget, XtPointer, XtPointer)
Callbacks for the edit functions.

static void
AddColumnsCB (Widget, XtPointer, XtPointer)

static void
DeleteRowsCB (Widget, XtPointer, XtPointer)

static void
DeleteColumnsCB (Widget, XtPointer, XtPointer)

static void
PurgeCB (Widget, XtPointer, XtPointer)

static void
DeleteAllCB (Widget, XtPointer, XtPointer)

static void
CutTextsCB (Widget, XtPointer, XtPointer)

static void
CopyTextsCB (Widget, XtPointer, XtPointer)

static void
PasteTextsCB (Widget, XtPointer, XtPointer)

static void
SelectAllCB (Widget, XtPointer, XtPointer)

static void
SortRowsCB (Widget, XtPointer, XtPointer)

static void

SortColumnsCB (Widget, XtPointer, XtPointer)

static void
ShowRowColumnLabelsCB (Widget, XtPointer, XtPointer)
Callbacks for the view functions

static void
PrintRowColumnLabelsCB (Widget, XtPointer, XtPointer)

static void
SetNumberOfRowsCB (Widget, XtPointer, XtPointer)
Callbacks for the properties functions.

static void
SetNumberOfColumnsCB (Widget, XtPointer, XtPointer)

static void
SetDefaultRowHeightCB (Widget, XtPointer, XtPointer)

static void
SetDefaultColumnWidthCB (Widget, XtPointer, XtPointer)

static void
SetMarginWidthCB (Widget, XtPointer, XtPointer)

static void
SetMarginHeightCB (Widget, XtPointer, XtPointer)

static void
AnnotateCellCB (Widget, XtPointer, XtPointer)

static void
AnnotateCellOKCB (Widget, XtPointer, XtPointer)

static void
DefaultRowAlignmentCB (Widget, XtPointer, XtPointer)

static void

DefaultColumnAlignmentCB (Widget,
XtPointer,
XtPointer)

static void
UpdateRowAlignmentCB (Widget, XtPointer,
XtPointer)

static void
UpdateColumnAlignmentCB (Widget,
XtPointer,
XtPointer)

static void
DefaultColumnAlignmentOKCB (Widget,
XtPointer,
XtPointer)

static void
DefaultColumnAlignmentDefaultCB (Widget,
Xt-
Pointer,
XtPointer)

static void
UpdateColumnAlignmentOKCB (Widget,
XtPointer,
XtPointer)

static void
UpdateColumnAlignmentDefaultCB (Widget,
Xt-
Pointer,
XtPointer)

static void
DefaultRowAlignmentOKCB (Widget,
XtPointer,
XtPointer)

static void
DefaultRowAlignmentDefaultCB (Widget,
XtPointer,
XtPointer)

static void

UpdateRowAlignmentOKCB (Widget,
XtPointer,
XtPointer)

static void
UpdateRowAlignmentDefaultCB (Widget,
XtPointer,
XtPointer)

static void
UpdateTableLineStyleCB (Widget, XtPointer,
XtPointer)

static void
UpdateTableLineStyleOKCB (Widget,
XtPointer,
XtPointer)

static void
UpdateTableLineStyleDefaultCB (Widget,
XtPointer,
XtPointer)

static void
UpdateTableLineWidthCB (Widget, XtPointer,
XtPointer)

static void
UpdateTableLineWidthOKCB (Widget,
XtPointer,
XtPointer)

static void
UpdateTableLineWidthDefaultCB (Widget,
XtPointer,
XtPointer)

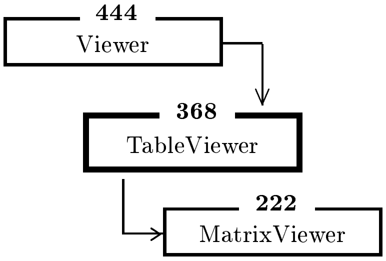
static void
UpdateCellTextUnderlineCB (Widget,
XtPointer,
XtPointer)

368

class **TableViewer** : public Viewer

table viewer class.

Inheritance



Public Members

	TableViewer (Config *c, TableWindow *t)	*creation*
virtual	**~TableViewer** ()	*deletion*
void	**SetTable** (Table *t)	
virtual void	**Initialize** ()	*create table of (defaultNumberOf)Rows x (defaultNumberOf)Columns.*
void	**InitTable** (unsigned rows, unsigned columns)	
void	**SetVisible** (bool b)	*Make all cells and lines (in)visible*
void	**Draw** ()	*draw all cells and labels*
void	**Undraw** ()	*erase all cells and labels*
bool	**TextMode** (CellText *c)	*start edit object initialized with c*
void	**TextModeOff** ()	*stop edit object and update celltext.*
void	**KeyTyped** (int x, int y, int key)	

key is pressed at (x, y)

void **Select** (int x, int y)
button1 is pressed at (x, y)

void **Adjust** (int x, int y)
button2 is pressed at (x, y)

void **MovingPointer** (int x, int y)
mouse pointer moved to (x, y)

Command*
Drag (int x, int y)
button1 is dragged to (x, y)

Command*
Connect (int x, int y)
button2 is dragged to (x, y)

Cell* **HitCell** (int x, int y)
return cell iff (x, y) is in cell,

CellText* **HitCellText** (int x, int y)
return cell-text iff (x, y) is in text,

LinePiece* **HitVLinePiece** (int x, int y)
return line iff (x, y) is on vertical line,

LinePiece* **HitHLinePiece** (int x, int y)
return line iff (x, y) is on horizontal line,

RowColumnLabel*
HitRowLabel (int x, int y)
return row label iff (x, y) is in row-label,

RowColumnLabel*
HitColumnLabel (int x, int y)
return column label iff (x, y) is in column-label,

int **HitRow** (int x, int y)
return rownr if (x, y) is in row,

int **HitColumn** (int x, int y)
return colnr if (x, y) is in column,

void **ShowRowColumnLabels** (bool b)

void **PrintRowColumnLabels** (bool b)

bool **IsShowRowColumnLabels** () const

bool **IsPrintRowColumnLabels** () const

void **RecomputeSizeCell** (Cell *c)
adapt cell size (row + column) to fit its cell text.

void **UpdateText** (Cell *c, const string *newtext)
update the text in cell c.

virtual void
MoveRow (CellRow *r, unsigned newpos)
move row from oldpos to newpos.

virtual void
MoveColumn (CellColumn *c, unsigned newpos)
move column from oldpos to newpos.

virtual void
AddRow (CellRow *c)
add row to end of table.

virtual void
AddColumn (CellColumn *c)
add column to right of table.

virtual void
InsertRow (CellRow *c)
insert row into table (automatically shifting).

virtual void
InsertColumn (CellColumn *c)
insert column into table (automatically shifting)

virtual void
DeleteRow (CellRow *c)
delete row from table.

virtual void
DeleteColumn (CellColumn *c)
delete column from table.

void **ResizeRow** (CellRow *r, int newSize)

void **ResizeColumn** (CellColumn *c, int newSize)

void **Move** (const Point *delta)

move entire table.

CellRow*	**GiveRow** (unsigned r) *return r-th row.*
CellColumn*	**GiveColumn** (unsigned c) *return c-th column.*
void	**GetSelectedRows** (List<CellRow *> *r)
void	**GetSelectedColumns** (List<CellColumn *> *c)
void	**GetEmptyRows** (List<CellRow *> *r)
void	**GetEmptyColumns** (List<CellColumn *> *c)
void	**DeselectAll** ()
void	**SelectAll** ()
void	**SelectCell** (Cell *c)
void	**DeselectCell** (Cell *c)
void	**SelectVector** (CellVector *v)
void	**DeselectVector** (CellVector *v)
bool	**IsAllSelected** (CellVector *v)
void	**MoveSelection** (int key)
void	**CalcSizeElements** (Point &topLeft, Point &bottomRight)
void	**CalcSizeSelection** (Point &topLeft, Point &bottomRight)
void	**UpdateMarginWidth** (int newWidth)
void	**UpdateMarginHeight** (int newHeight)
void	**UpdateDefaultRowHeight** (int n)
void	**UpdateDefaultColumnWidth** (int n)
int	**GetDefaultRowHeight** () const
int	**GetDefaultColumnWidth** () const
unsigned	**GetNumberOfColumns** () const
unsigned	**GetNumberOfRows** () const
unsigned	**GetDefaultNumberOfColumns** () const
unsigned	**GetDefaultNumberOfRows** () const

TextAlign::Type **GetDefaultRowAlignment** () const

TextAlign::Type **GetDefaultColumnAlignment** () const

int **GetRowHeight** (unsigned row)

int **GetColumnWidth** (unsigned column)

const Point* **GetTopLeft** () const

int **GetWidth** () const

int **GetHeight** () const

int **GetMarginWidth** () const

int **GetMarginHeight** () const

Point **GetRowTopLeft** (unsigned row)

Point **GetColumnTopLeft** (unsigned column)

void **SetGrafport** (Grafport *g)

CellTextBuffer* **GetBuffer** () const

int **GetTopSelectionRow** ()
return index of topmost selected row, or -1 if selection is empty.

int **GetLeftSelectionColumn** ()
return index of leftmost selected column, -1 if selection is empty.

List <Cell *> * **GetSelection** () const

unsigned **NrSelected** () const

unsigned **GetCurrentRow** () const

unsigned **GetCurrentColumn** () const

void **SetCurrentRow** (unsigned n)

void **SetCurrentColumn** (unsigned n)

void **GetCells** (List<Cell *> *cell, const string *s, bool sens, bool substring) *get cell containing s as text (case sensitive, substring).*

Cell*	**GetNextCell** (const string *s, bool sens, bool substring) *return the next cell containing s as text.*
void	**SetDefaultColumnAlignment** (TextAlign::Type a)
void	**SetDefaultRowAlignment** (TextAlign::Type a)
void	**SetLineStyle** (Side::Type side, LineStyle::Type style)
void	**SetDefaultNumberOfRows** (unsigned)
void	**SetDefaultNumberOfColumns** (unsigned)
void	**UpdateLineWidthSelection** (unsigned wd)
void	**UpdateLineStyleSelection** (LineStyle::Type l)
void	**UpdateTextAlignmentSelection** (TextAl-ign::Type a)
void	**UpdateLineColorSelection** (const string *color)
void	**UpdateTextColorSelection** (const string *color)
void	**UpdateFillColorSelection** (const string *color, FillStyle::Type f)
void	**SetDefaultFont** (XFont *ft)
void	**UpdateFontSelection** (int f, int s, int p) *set font family, style, size (no update when value < 0).*
bool	**ReadAttributes** (InputFile *f, bool appending, unsigned &rowsToAdd, unsigned &columnsToAdd, double format)
bool	**ReadColumns** (InputFile *f, unsigned nr, unsigned fromRow, unsigned fromColumn, double format)

bool	**ReadRows** (InputFile *f, unsigned nr, unsigned fromRow, unsigned fromColumn, double format)	
void	**WriteAttributes** (OutputFile *f)	
void	**WriteColumns** (OutputFile *f)	
void	**WriteRows** (OutputFile *f)	
void	**WriteSelection** (OutputFile *f)	

Protected Members

List <CellRow *> *	**GetRows** () const	
List <CellColumn *> *	**GetColumns** () const	
void	**SetDefaultColumnWidth** (int w)	
void	**SetDefaultRowHeight** (int h)	
void	**SetMarginWidth** (int w)	
void	**SetMarginHeight** (int h)	

Private Members

Table*	**table**	
Point	**topLeft**	*the topleft position of the table.*
int	**width**	*the width of all cells + row or column labels.*
int	**height**	*the height of all cells + row or column labels.*
List <CellRow *> *	**rows**	*a list of rows (List is user-defined ADT).*
List <CellColumn *> *	**columns**	*a list of columns*
CellSelection*		

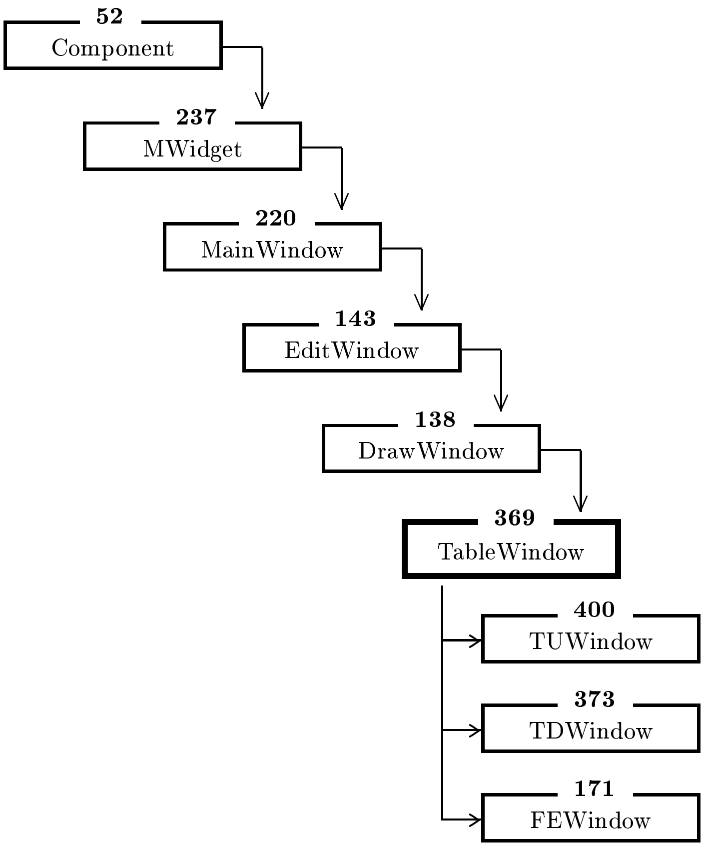
	selection	*the current selection.*
CellTextBuffer*	**buffer**	*a cut buffer for storing cell texts;*
int	**marginWidth**	*minimal distance between text and column boundaries.*
int	**marginHeight**	*minimal distance between text and row boundaries.*
TextAlign::Type	**defaultColumnAlignment**	*each newly created clm is aligned like this*
TextAlign::Type	**defaultRowAlignment**	*each newly created row is aligned like this.*
int	**defaultRowHeight**	*each newly created row has this height.*
int	**defaultColumnWidth**	*each newly created column has this width.*
unsigned	**defaultNumberOfRows**	*when there are no other rows/columns yet.*
unsigned	**defaultNumberOfColumns**	*each newly created row/column get this number of cells.*
unsigned	**currentRow**	
unsigned	**currentColumn**	
bool	**showRowColumnLabels**	
bool	**printRowColumnLabels**	
Point	**defaultTopLeft**	
void	**DeleteCells** ()	
bool	**GoIntoTextMode** (Cell *cell)	

369

class **TableWindow** : public DrawWindow

table edit window class.

Inheritance



Public Members

TableWindow (const char *name)

virtual **~TableWindow** ()

virtual TableViewer*
CreateViewer ()

virtual Table*

CreateTable ()

Protected Members

TableViewer* **GetTableViewer** ()

Private Members

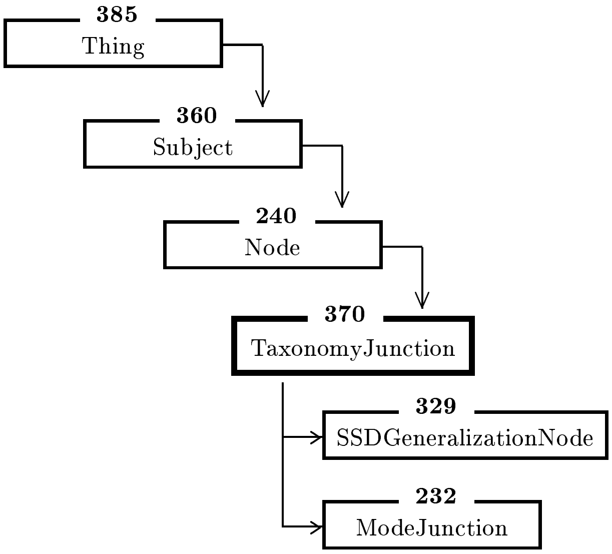
Table*	**table**
TableViewer*	**viewer**
int	**GetTool** ()
void	**ParseUserOptions** (int argc, char** argv)
void	**WriteUsage** (const char *prog)
void	**CreateComponents** (Widget window) *Functions to create the constituting parts of the table window.*
Widget	**CreateMenuBar** (Widget parent)
void	**CreateDocument** ()
void	**Finalize** ()
int	**initRows**
int	**initColumns**
int	**initCellHeight**
int	**initCellWidth**
MenuItem*	**defaultPropertiesItems**
void	**InitMenuItems** ()

370

class **TaxonomyJunction** : public Node

taxonomy junction (ER) node class.

Inheritance



Public Members

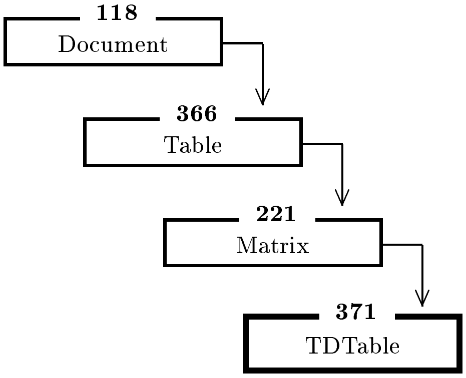
	TaxonomyJunction (ERGraph *g)
int	**GetClassType** () const
Subject*	**Clone** ()
virtual Subject::NameErrType	**SetName** (const string *name)

371

class **TDTable** : public Matrix

transaction decomposition table class.

Inheritance



Public Members

	TDTable (Config *c, TDWindow *t, TDViewer *v)
void	**CheckDocument** ()

Private Members

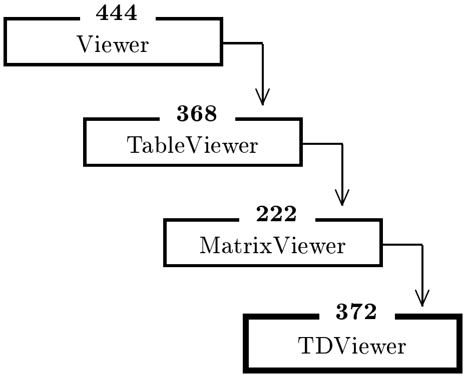
bool	**CheckTextString** (Cell *c, const string *text)

372

class **TDViewer** : public MatrixViewer

transaction decomposition table class.

Inheritance



Public Members

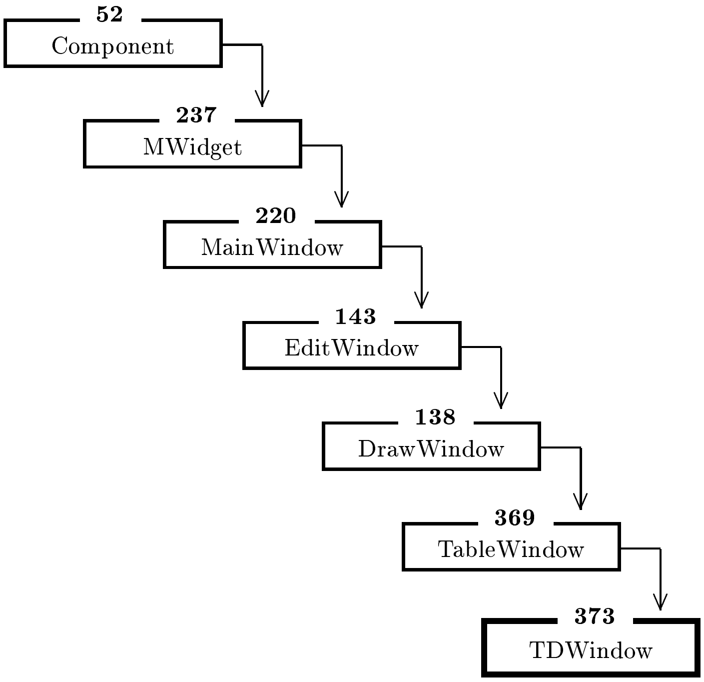
	TDViewer (Config *c, TableWindow *t)
void	**Initialize** ()

373

class **TDWindow** : public TableWindow

transaction decomposition table edit window class.

Inheritance



Public Members

	TDWindow (const char *name)
TableViewer*	**CreateViewer** ()
Table*	**CreateTable** ()

Private Members

int	**GetTool** ()

374

class **TextAlign**

these are alignments of a multiline string.

Public Members

enum **Type**

static Type
String2Type (const string* s)
e.g. convert "left" to Left.

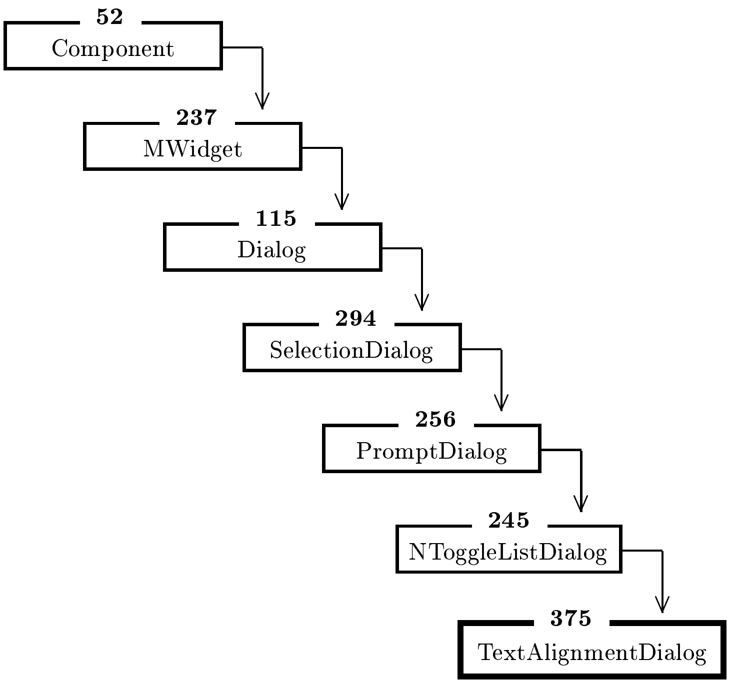
static void
Type2String (Type t, string* s)
e.g. convert Left to "left".

375

class **TextAlignmentDialog** : public NToggleListDialog

text alignment chooser dialog class.

Inheritance



Public Members

TextAlignmentDialog (Widget parent, bool horizontal=True)

const char* **GetClassName** ()

Private Members

void **Configure** ()

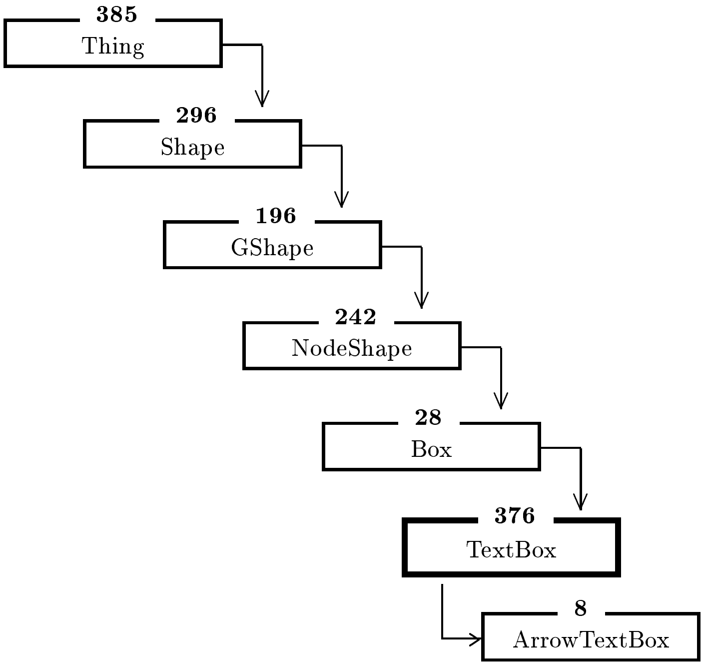
bool **horizontal**

376

class **TextBox** : public Box

invisible box class (only text is visible).

Inheritance



Public Members

	TextBox (ShapeView *v, Grafport *g, int x, int y, int w = MIN_WIDTH, int h = MIN_HEIGHT)
	TextBox (const NodeShape &c)
Shape*	**Clone** ()
int	**GetClassType** () const
int	**HitHandles** (int, int)

		returns -1 because you can't resize them yourself.
void	**SetTextShape** ()	*adapt size to name size.*
void	**AdjustSize** ()	

Protected Members

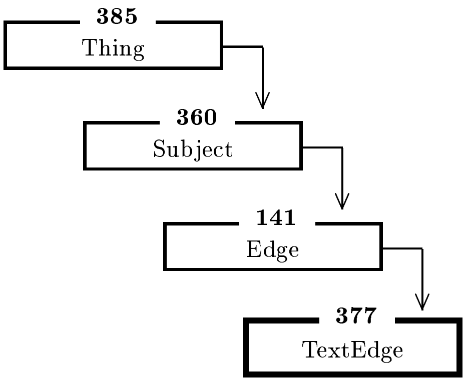
static const int
TEXT_MARGIN
margin between the name and the box itself.

377

class **TextEdge** : public Edge

generic textual tree edge class.

Inheritance



Public Members

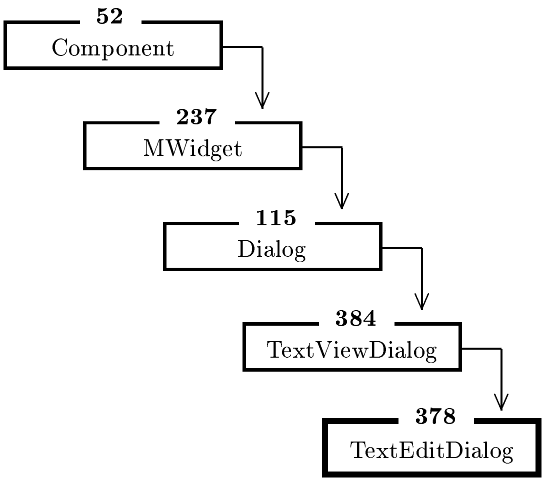
	TextEdge (TRGraph *g, Subject *n1, Subject *n2)
int	**GetClassType** () const
Subject*	**Clone** ()

378

class **TextEditDialog** : public TextViewDialog

manages text editor dialogs.

Inheritance



Public Members

	TextEditDialog (Widget parent)
const char*	**GetClassName** ()
void	**Load** ()
void	**Replace** ()
void	**ClearReplace** ()

Protected Members

void	**Configure** ()
void	**InitMenuItems** ()
static void	**LoadCB** (Widget, XtPointer, XtPointer)
static void	

LoadOKCB (Widget, XtPointer, XtPointer)

static void
DeleteAllCB (Widget, XtPointer, XtPointer)

static void
CutCB (Widget, XtPointer, XtPointer)

static void
PasteCB (Widget, XtPointer, XtPointer)

static void
ReplaceCB (Widget, XtPointer, XtPointer)

static void
ReplaceNextCB (Widget, XtPointer, XtPointer)

static void
ReplaceFindCB (Widget, XtPointer, XtPointer)

static void
ReplaceAllCB (Widget, XtPointer, XtPointer)

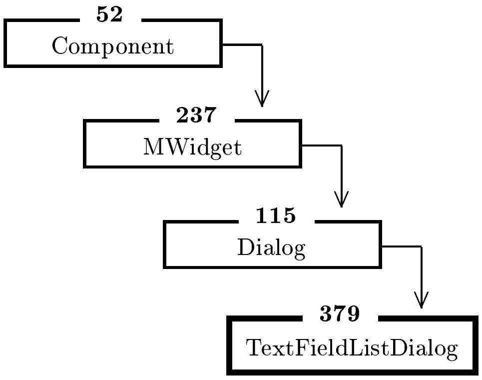
static void
ClearReplaceCB (Widget, XtPointer, XtPointer)

379

class **TextFieldListDialog** : public Dialog

dialog class for showing a list of text fields.

Inheritance



Public Members

	TextFieldListDialog (Widget parent)
virtual	**~TextFieldListDialog** ()
const char*	**GetClassName** ()
void	**CreateFields** (List<string *> *l, List<string *> *f)
void	**GetField** (unsigned i, string *f)
void	**GetLabel** (unsigned i, string *l)
void	**SetField** (unsigned i, const string *f)
void	**SetFieldWidth** (unsigned n)
void	**SetLabel** (unsigned i, const string *l)
unsigned	**NumFields** ()

Protected Members

void	**CreateWidget** ()

void **Configure** ()

Private Members

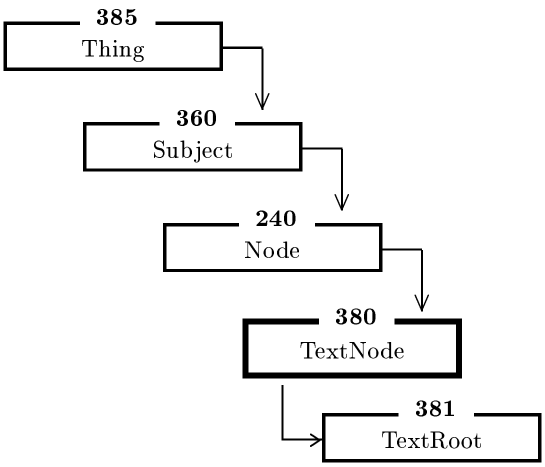
Widget	**rc**
int	**fieldWidth**
Widget*	**textFields**
unsigned	**numTextFields**
Widget*	**labels**
unsigned	**numLabels**

380

class **TextNode** : public Node

generic textual tree node class.

Inheritance



Public Members

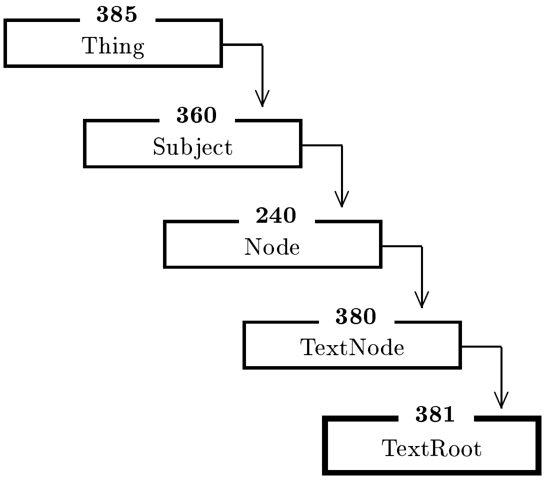
	TextNode (TRGraph *g)
int	**GetClassType** () const
Subject*	**Clone** ()
virtual bool	**IsRoot** ()
bool	**ReadMembers** (InputFile *f, double format)

381

class **TextRoot** : public TextNode

generic textual tree root node class.

Inheritance



Public Members

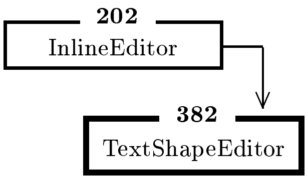
	TextRoot (TRGraph *g)
int	**GetClassType** () const
Subject*	**Clone** ()
bool	**IsRoot** ()

382

class **TextShapeEditor** : public InlineEditor

text shape editor class.

Inheritance



Public Members

	TextShapeEditor (DiagramViewer *v)	
void	**Start** (TextShape *t)	*initialize and start the inline editor with textshape.*
void	**Stop** ()	*stop the object and update textshape.*
void	**Abort** ()	*abort editing and restore everything*
void	**StartWindow** (TextShape *t)	*start editing in separate window.*
TextShape*	**GetTextShape** () const	

Private Members

TextShape*	**textshape**	*the text shape that will be edited.*
void	**UpdateText** ()	
void	**Draw** ()	
void	**SetBackground** ()	
void	**RestoreBackground** ()	

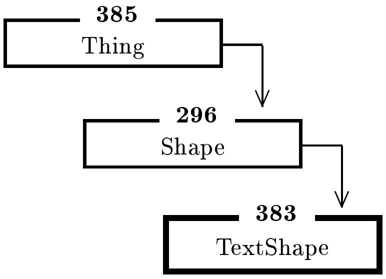
string **saveBackground**

383

class **TextShape** : public Shape

text shape class that displays a character string.

Inheritance



Public Members

	TextShape (ShapeView *v, Grafport *g, Shape *parent)
	TextShape (const TextShape &t)
Shape*	**Clone** ()
void	**Draw** ()
void	**Undraw** ()
void	**DrawDuplicate** () *no duplicate asteriks are possible for text shapes.*
void	**SetParent** (Shape *shape)
Shape*	**GetParent** () const
void	**SetString** (const string *s) *set text and recalculate the sizes.*
const string*	**GetString** () const
TextShape*	**ChooseTextShape** (int, int)
TextShape*	

	HitTextShape (int x, int y)	
bool	**InTextArea** (int x, int y)	
bool	**IsTextShape** () const	
bool	**IsEmpty** () const	
void	**SetEmpty** ()	
void	**DrawOutLine** (const Point *center)	*draws a dotted rectangle around the textshape.*
void	**SetPosition** (const Point *p, bool snap=False)	*set textshape at position (not using grid !).*
bool	**ContainsPt** (int x, int y)	*returns if (x, y) is in textshape (not empty string).*
void	**TextBox** (int &wd, int &ht)	*return text width and height (depends on font).*
void	**UpdateAlignment** (TextAlign::Type newtype)	
void	**SetAlignment** (TextAlign::Type newtype)	
TextAlign::Type	**GetAlignment** () const	
XFont*	**GetFont** () const	
void	**SetFont** (XFont *ft)	
void	**UpdateFont** (XFont *ft)	
int	**GetFontHeight** () const	
int	**GetStringHeight** () const	
int	**GetStringWidth** () const	
bool	**IsWellPositioned** () const	*if wellPositioned = false then current position isn't preferred.*
void	**SetWellPositioned** (bool b)	*!wellPositioned indicates parent should recalculate child position.*
const string*		

	GetDescription () const	*returns a string which describes the kind of text shape.*
void	**SetDescription** (const string *s)	
void	**SetDescription** (const char *s)	
bool	**IsOneLine** () const	
void	**SetOneLine** (bool o)	
bool	**IsUnderlined** () const	
void	**SetUnderlined** (bool b)	
void	**UpdateUnderlined** (bool b)	
bool	**HasTextShape** (TextShape *t) const	*return if it has t as one of the text shapes.*
bool	**HasString** (const string *s, bool sens, bool sub) const	*textshape contains s as string.*
int	**ReplaceString** (const string *s1, const string *s2, bool sens, bool sub)	*replace occurrences of s1 by s2 (if allowed by Update).*
bool	**HasString** (const string *s, bool caseSens, bool substring, List<TextShape *> *list)	*puts itself into list when text contains s*
int	**GetSequence** () const	
void	**SetSequence** (int n)	

Private Members

string	**text**	*the text of the textshape*
Shape*	**parent**	*the shape the textshape belongs to.*
XFont*	**font**	*the X font for drawing.*
TextAlign::Type		

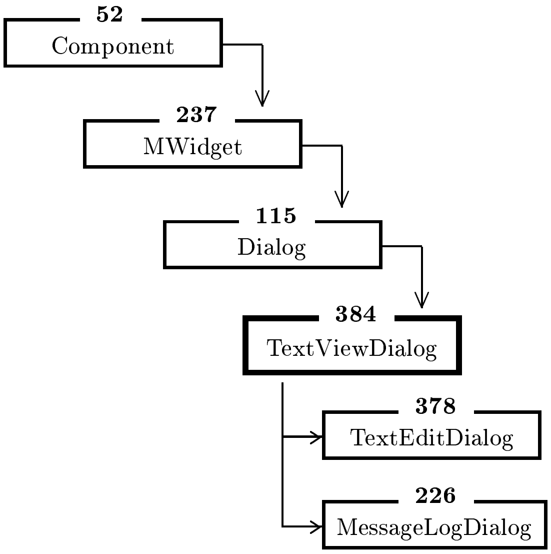
	alignment	*the text alignment of multi-line strings.*
int	**stringWidth**	*the width of the text string (depends on font).*
int	**stringHeight**	*the height of the text string (depends on font).*
string	**description**	*a short text describing what kind of text shape this is.*
bool	**wellPositioned**	*indicates that parent should recalculate text shape.*
void	**DrawSelect** ()	*draw dotted rect.*
int	**sequence**	*optional sequence number when text shape is part of a row.*
bool	**oneLine**	*is the text shape supposed to be on a single line?*
bool	**underlined**	*is the text shape underlined*

384

class **TextViewDialog** : public Dialog

manages text editor dialogs.

Inheritance



Public Members

	TextViewDialog (Widget parent)
virtual	**~TextViewDialog** ()
const char*	**GetClassName** ()
void	**Popup** ()
void	**SetTextSize** (int nrOfRows, int nrOfColumns)
void	**SetTextRows** (int nrOfRows)
void	**SetTextColumns** (int nrOfColumns)
void	**SetTextMaxSize** (int nrOfRows, int nrOfColumns)
void	**SetTextString** (const string *text)

void **GetTextString** (string *t)

void **SetStatus** (const char *s)

void **SetStatus** (const string *s)

void **SetDirectory** (const char *d)

void **SetDirectory** (const string *d)

void **Save** ()

void **Print** ()

bool **SaveFile** (const char *file)

bool **SaveFile** (const string *file)

bool **LoadFile** (const char *file)

bool **LoadFile** (const string *file)

void **DeleteAll** ()

void **Cut** ()

void **Copy** ()

void **Paste** ()

void **ClearSelection** ()

void **Find** ()

void **ClearFind** ()

void **DoFind** (int n)

FileSelectionDialog*
GetFileSelectionDialog ()

Protected Members

void **Configure** ()

void **CreateWidget** ()

Widget **GetTextWidget** ()

ReplaceDialog*
GetReplaceDialog ()

void **SetReplaceDialog** (ReplaceDialog *r)

virtual void
InitMenuItems ()

static void

PrintCB (Widget, XtPointer, XtPointer)

static void
SaveAsCB (Widget, XtPointer, XtPointer)

static void
SaveOKCB (Widget, XtPointer, XtPointer)

static void
CopyCB (Widget, XtPointer, XtPointer)

static void
FindCB (Widget, XtPointer, XtPointer)

static void
FindNextCB (Widget, XtPointer, XtPointer)

static void
FindAllCB (Widget, XtPointer, XtPointer)

static void
ClearSelectionCB (Widget, XtPointer, XtPointer)

static void
ClearFindCB (Widget, XtPointer, XtPointer)

MenuItem*
fileItems

MenuItem*
editItems

MenuItem*
searchItems

Private Members

Widget **textWidget**

Widget **menuBar**

Widget **textStatus**

FileSelectionDialog*
fileSelector

PrinterOptions*
printerOptions

FindDialog*
findDialog

ReplaceDialog*

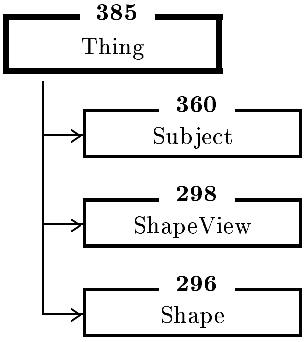
	replaceDialog
Menu*	**fileMenu**
Menu*	**editMenu**
Menu*	**searchMenu**
string	**directory**
string	**saveFile**
string	**loadFile**
int	**maxRows**
int	**maxColumns**

385

class **Thing**

superclass of all identifiable things that can be written to file.

Inheritance



Public Members

Thing ()

Thing (const Thing &s)

virtual **~Thing** ()

unsigned long **GetId** ()

void **SetId** (unsigned long val)

virtual bool **IsSubject** () const

virtual bool **IsShape** () const

virtual bool **IsView** () const

virtual int **GetClassType** () const
return code of the lowest class.

void **Write** (OutputFile *ofile)
write thing to file ofile.

virtual bool

	ReadMembers (InputFile *, double)	*read from input file. The second param is the read file format.*
virtual bool	**SetReferences** (AssocList *)	*update references when thing is read.*
enum	**AssocType**	*possible types in the assoclist.*
virtual bool	**CheckReferences** ()	*Check member references.*

Private Members

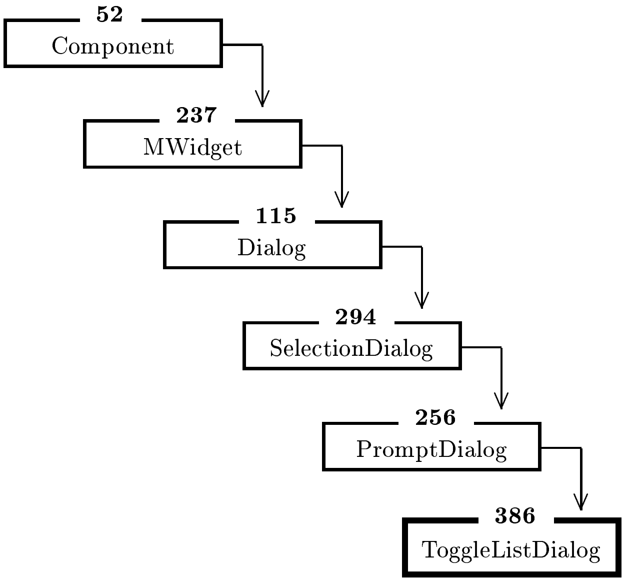
unsigned long	**id**	*internal identifier.*
virtual void	**WriteMembers** (OutputFile *)	*write parts of thing to file.*

386

class **ToggleListDialog** : public PromptDialog

dialog class that shows a list of radio buttons.

Inheritance



Public Members

	ToggleListDialog (Widget parent, bool prompt)
virtual	**~ToggleListDialog** ()
const char*	**GetClassName** ()
void	**SetToggleChangedCallback** (XtCallbackProc fun, XtPointer cd)
void	**CreateToggles** (List<string *> *items)
void	**CreateToggles** (const string *label, List<string *> *items)
void	**SetToggleText** (unsigned i, const string *text)

void	**SetToggleBitmap** (unsigned i, Bitmap *bitmap)
void	**GetToggleText** (unsigned i, string *text)
void	**SetValue** (unsigned initVal)
int	**GetValue** ()
unsigned	**NumToggles** ()
bool	**IsPromptVisible** ()

Protected Members

void	**CreateWidget** ()
void	**Configure** ()

Private Members

Widget*	**toggleList**
unsigned	**numToggles**
XtCallbackProc	**toggleFun**
XtPointer	**clientData**
bool	**promptVisible**

387

class **Toolkit**

information about the different tools.

Public Members

static const char
TOOLKIT_NAME []
The application (toolkit) name.

enum **ToolType** *Definitions of the different editors (tools) that are implemented.*

static bool
TableEditor (int tool)
is tool table editor.

static bool
DiagramEditor (int tool)
is tool diagram editor.

static bool
TreeEditor (int tool)
is tool tree editor.

static bool
TextEditor (int tool)
is tool text editor.

static bool
EditorWithoutCurves (int tool)
tool can not have curved edges.

static bool
EditorWithoutDuplicates (int tool)
tool can not have duplicate shapes.

static bool
HierarchicEditor (int tool)
tool can edit hierarchic documents.

static bool
GenericEditor (int tool)
tool has no soft constraints

static bool

EditorWithInterEdgeConnections (int tool)
tool is diagram editor in which edges can connect to edges

static bool
EditorWithModelChecker (int tool)
tool is diagram editor with model checking facilities

static const char
STARTUP_TOOL_NAME []
The name (in upper case) of the startup tool.

static const char*
TOOL_TITLES []
Window titles for the different tools.

static const char*
TOOL_NAMES []
Short names (in upper case) of the editing tools.

static const char*
EXECUTABLE_NAMES []
Names of the tool executables.

static const char*
DOCUMENT_TYPES []
The full names of the document types.

static const char*
DOCUMENT_SUFFIXES []
Document name suffixes.

static const char
DOCUMENT_DEFAULT []
Default document name.

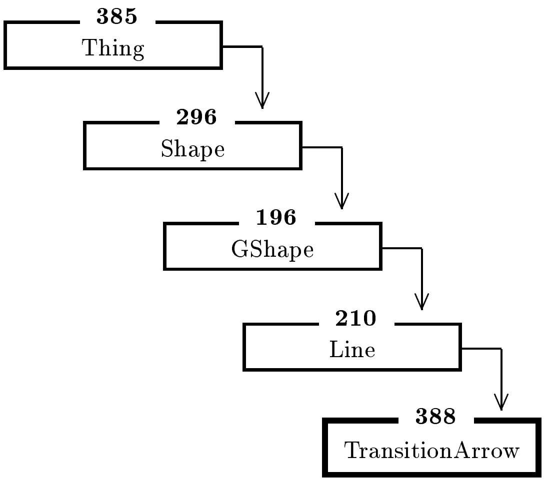
static const char*
DOCUMENT_TYPES_INFO []
The help info of the document types.

388

class **TransitionArrow** : public Line

arrow having extra horiz. line separating event and list of actions (STD).

Inheritance



Public Members

	TransitionArrow (ShapeView *v, Grafport *g, GShape *n1, GShape *n2, List<Point *> *line, bool Curved)
	TransitionArrow (ShapeView *v, Grafport *g, GShape *n1, GShape *n2, bool Curved)
	TransitionArrow (const TransitionArrow &line)
virtual	**~TransitionArrow** ()
Shape*	**Clone** ()
int	**GetClassType** () const
bool	**InTextArea** (int x, int y)
TextShape*	

	HitTextShape (int x, int y)
TextShape*	**ChooseTextShape** (int x, int y)
bool	**HasTextShape** (TextShape *t) const
bool	**ContainsPt** (int x, int y)
void	**SetSelect** (bool s)
bool	**HitHLine** (int x, int y)
const Point*	**GetHLineFrom** () const
const Point*	**GetHLineTo** () const
void	**UpdateAnchor** (const Point *delta, const Point *position)
void	**UpdateEvent** (const string *s)
void	**UpdateAction** (const string *s, unsigned n, bool update)
void	**CalcPosition** ()
int	**GetLeftMost** () const *as line, but event and actions are also considered.*
int	**GetTopMost** () const
int	**GetRightMost** () const
int	**GetBottomMost** () const
void	**WriteMembers** (OutputFile *f)
bool	**ReadMembers** (InputFile *f, double format)
bool	**SetAssocSubject** (AssocList *)
void	**SetTextShape** ()
void	**SetTextColor** (const string *color)
void	**SetFont** (XFont *ft)
void	**SetGrafport** (Grafport *g)
void	**SetView** (ShapeView *v)
bool	**HasString** (const string *s, bool sens, bool sub) const

bool **HasNameString** (const string *, bool, bool) const

bool **HasString** (const string*, bool, bool, List<TextShape *> *)

enum **SepType**

void **MoveRaw** (const Point *delta)

Private Members

TextShape* **event**

List <TextShape *> * **actions**

Point **hlineFrom**

Point **hlineTo**

Point **anchorPoint**

SepType **separator**

const char* **Sep2String** (SepType sep)

TransitionArrow::SepType **String2Sep** (const string *s)

int **HLineLength** ()

void **CalcSeparator** ()

void **CalcHLinePart** ()

void **CalcPositionEvent** ()

void **CalcPositionActions** ()

void **CalcPositionHLine** () *calculate line position and action and event.*

void **SetAction** (const string *s, unsigned n, bool update)

void **DrawShape** () *draw arrow + all labels + 1 horiz line.*

static const int

HLINELEN *minimal length hline.*

static const int
HLINEXOFFSET

static const int
HLINEYOFFSET

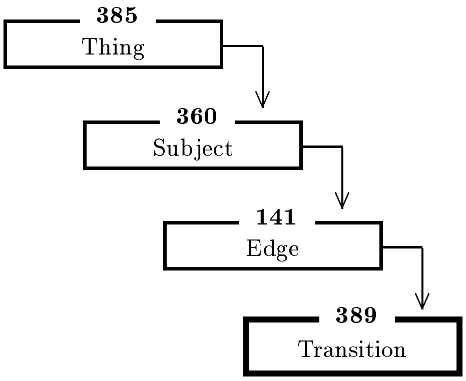
static const int
HLINEDIST *distance to horizontal line.*

389

class **Transition** : public Edge

transition (STD) edge class.

Inheritance



Public Members

	Transition (STGraph *g, Subject *s1, Subject *s2)
	Transition (const Transition &t)
virtual	**~Transition** ()
int	**GetClassType** () const
Subject*	**Clone** ()
Subject::NameErrType	**SetName** (const string *name) *return WRONG_NAME*
Subject::NameErrType	**SetAction** (const string *a, unsigned n, bool update)
const string*	**GetAction** (unsigned n)
bool	**HasAction** (const string *s)
unsigned	**NrActions** () const

Subject::NameErrType
SetEvent (const string *c)

const string*
GetEvent () const

void **WriteMembers** (OutputFile *f)

bool **ReadMembers** (InputFile *f, double format)

Private Members

string **event**

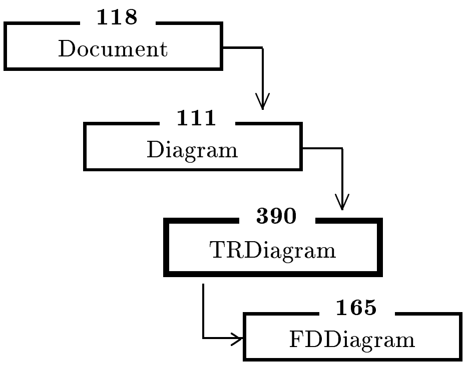
List <string *> *
actions

390

class **TRDiagram** : public Diagram

generic tree diagram class.

Inheritance



Public Members

	TRDiagram (Config *, TRWindow *, TRViewer *, TRGraph *)
virtual	**~TRDiagram** ()
Node*	**CreateNode** ()
Edge*	**CreateEdge** (Subject *n1, Subject *n2)
NodeShape*	**CreateNodeShape** (Node *node, int x, int y)
Line*	**CreateLine** (Edge *edge, GShape *fromShape, GShape *toShape, List<Point *> *line)
void	**UpdateNodeType** (int n)
void	**UpdateEdgeType** (int n)
void	**DrawTreeLayout** ()
void	**UndrawTreeLayout** ()
void	**RemoveAll** ()

void **Append** ()

Protected Members

bool **FindRoot** (Subject *subj)

bool **CheckEdgeConstraints** (Subject *n1, Subject *n2)

void **RecalculateTree** ()

Thing* **CreateThing** (int classNr)

DiagramChecks* **GetTreeChecks** ()

void **CheckDocument** ()

void **DoLoad** (const string *file)

void **PlaceShapes** () *force the viewer to draw in right mode.*

void **CalcParents** (TextNode *parent, TextNode *node)

void **GetChildren** (TextNode *node, List<Subject *> *children)

void **DumpChildren** (TextNode *node, unsigned level)

void **DrawTreeLayout** (TextNode *parent)

Private Members

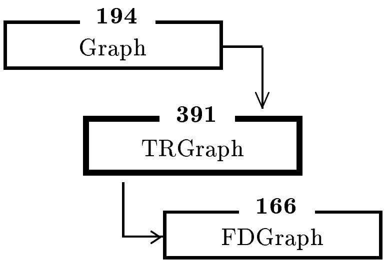
DiagramChecks* **trChecks**

391

class **TRGraph** : public Graph

generic textual tree graph class.

Inheritance



Public Members

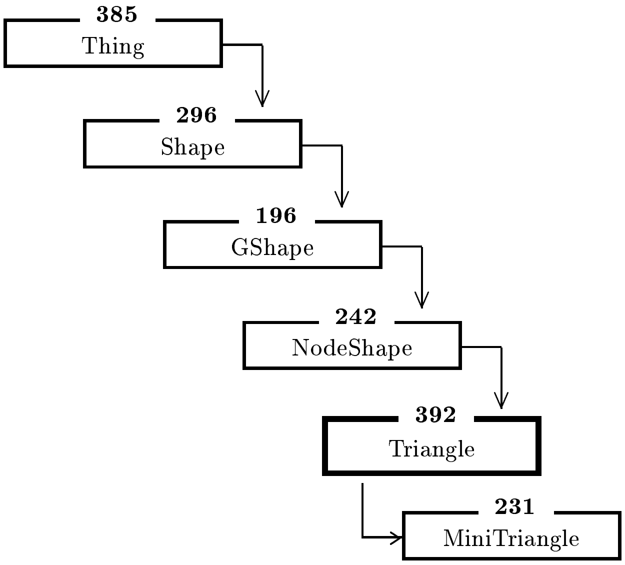
	TRGraph ()
void	**InitConnections** ()

392

class **Triangle** : public NodeShape

triangle shape class.

Inheritance



Public Members

static const int
TRIANGLE_WIDTH

static const int
TRIANGLE_HEIGHT

Triangle (ShapeView *v, Grafport *g, int x, int y,
int w = TRIANGLE_WIDTH,
int h = TRIANGLE_HEIGHT)

Triangle (const Triangle &t)

Triangle (const NodeShape &)

Shape* **Clone** ()

void **DrawOutLine** (const Point *center)

draws triangle dotted outline. snaps to the grid.

void **DrawDuplicate** ()

Point **GiveSnp** (Point to) const

Point **GiveSnp** (Point p, int order, int count, int &code)
same as GiveSnp(Point), i.e. not yet implemented.

void **SetSize** (int w, int h)

void **SetPosition** (const Point *position, bool snap=True)

bool **ContainsPt** (int x, int y)

int **GetClassType** () const

void **SetTextShape** ()

static const int **TRIANGLE_HANDLES**

Protected Members

void **CalcPoints** (int x, int y)

const Point* **GetPoints** ()

Private Members

Point **points** [3] *the tree corner points.*

void **CalcPositionName** ()

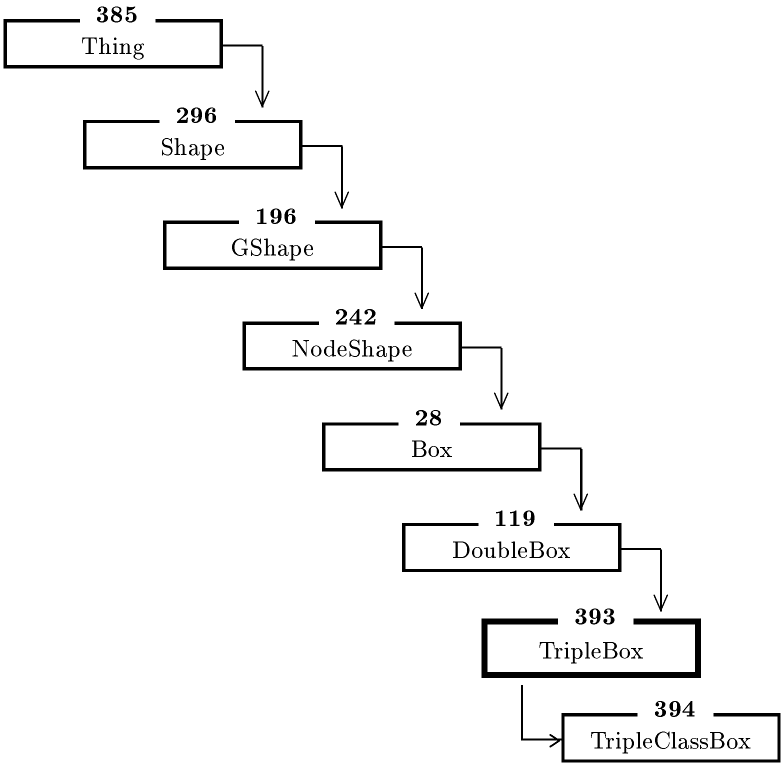
void **DrawShape** () *draws a triangle*

393

class **TripleBox** : public DoubleBox

three compartment box: 1st for name, 2nd for labels, 3rd for labels2.

Inheritance



Public Members

TripleBox (ShapeView *v, Grafport *g, int x, int y, int w, int h)

TripleBox (const TripleBox &s)

TripleBox (const DoubleBox &s)

TripleBox (const Box &s)

TripleBox (const NodeShape &s)

virtual **~TripleBox** ()

int **GetClassType** () const

Shape* **Clone** ()

TextShape*
ChooseTextShape (int x, int y)

bool **HasTextShape** (TextShape *t) const

void **UpdateLabel2** (const string *s, unsigned n, bool update)
update or insert string of n-th label2.

TextShape*
GetLabel2 (unsigned n)

unsigned **NrLabels2** ()

void **SetTextShape** ()

void **SetFont** (XFont *ft)

void **SetTextColor** (const string *color)

void **AdjustSize** ()

void **SetGrafport** (Grafport *g)

void **SetView** (ShapeView *v)

bool **HasString** (const string *s, bool sens, bool sub) const
text shape contains s as string.

bool **HasString** (const string*, bool, bool, List<TextShape *> *)

LineStyle::Type
GetSeparator2LineStyle () const

void **SetSeparator2LineStyle** (LineStyle::Type x)

Protected Members

void **DrawShape** ()

void **DrawTextShapes** ()

int **RequiredHeight** ()

int **DoubleBoxHeight** ()

void **CalcPositionLabels** ()

void **CalcPositionLabels2** ()

void **AdjustSizes** (const string *s)

List <TextShape *> *
GetLabels2 ()

string **label2Description**

Private Members

List <TextShape *> *
labels2 *the list of labels2 in the third compartment.*

LineStyle::Type
separator2LineStyle

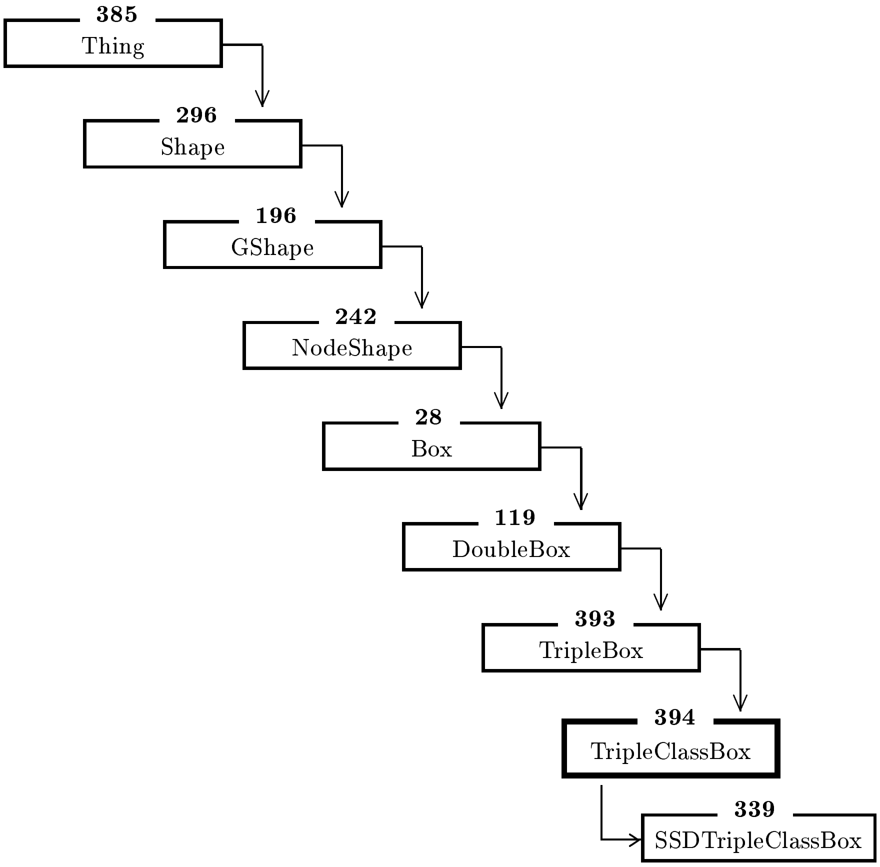
void **Init** ()

394

class **TripleClassBox** : public TripleBox

A triple box used as a class box with attributes and operations.

Inheritance



Public Members

static const int
TRIPLECLASSBOX_WIDTH

static const int

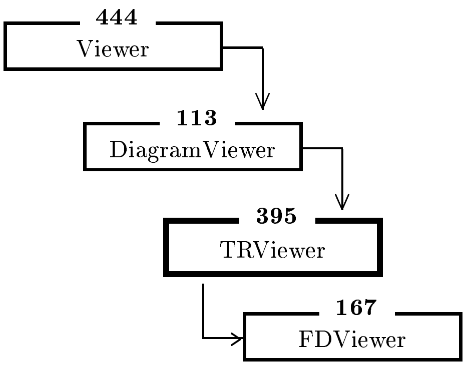
	TRIPLECLASSBOX_HEIGHT
	TripleClassBox (ShapeView *v, Grafport *g, int x, int y, int w = TRIPLECLASSBOX_WIDTH, int h = TRIPLECLASSBOX_HEIGHT)
	TripleClassBox (const Box &s)
	TripleClassBox (const TripleBox &s)
Shape*	**Clone** ()
bool	**SetAssocSubject** (AssocList *)
void	**SetTextShape** ()

395

class **TRViewer** : public DiagramViewer

generic textual tree viewer class.

Inheritance



Public Members

	TRViewer (Config *c, TRWindow *w)
void	**Initialize** ()
void	**Draw** ()
void	**Undraw** ()
void	**UpdateTreeMode** (bool newMode)
bool	**IsTreeMode** ()
void	**DrawTreeFork** (Shape *parent, List<Shape *> *children)
void	**Select** (int x, int y)
void	**Adjust** (int x, int y)
void	**MovingPointer** (int x, int y)
void	**KeyTyped** (int x, int y, int c)
Command*	**Drag** (int x, int y)
Command*	

	Connect (int x, int y)
void	**Cut** ()
void	**Copy** ()
void	**Paste** ()
void	**DeleteAll** ()
void	**SelectAll** ()
void	**DeleteSubjects** ()
void	**SameSize** ()
void	**AlignNodes** (NodeAlign::Type)
void	**Annotate** ()
void	**UpdateCurve** ()
void	**MoveAllShapes** (Document::MoveType t)
void	**FindAll** (const string *s, bool sens, bool substr, bool nameOnly)
void	**FindNext** (const string *s, bool sens, bool substr, bool nameOnly)
void	**ReplaceNext** (const string *s1, const string *s2, bool sens, bool substr, bool nameOnly)
void	**ReplaceAll** (const string *s1, const string *s2, bool sens, bool substr, bool nameOnly)

Protected Members

enum	**DirectionType**
void	**DrawTreeMode** ()
void	**DrawNormalMode** ()
void	**DrawVerticalTreeFork** (Shape *parent, List<Shape *> *children, TRViewer::DirectionType d)

void **DrawHorizontalTreeFork** (Shape *parent, List<Shape *> *children, TRViewer::DirectionType d)

Private Members

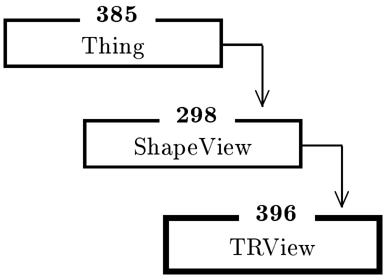
bool **treeMode**

396

class **TRView** : public ShapeView

generic textual tree shape view class.

Inheritance



Public Members

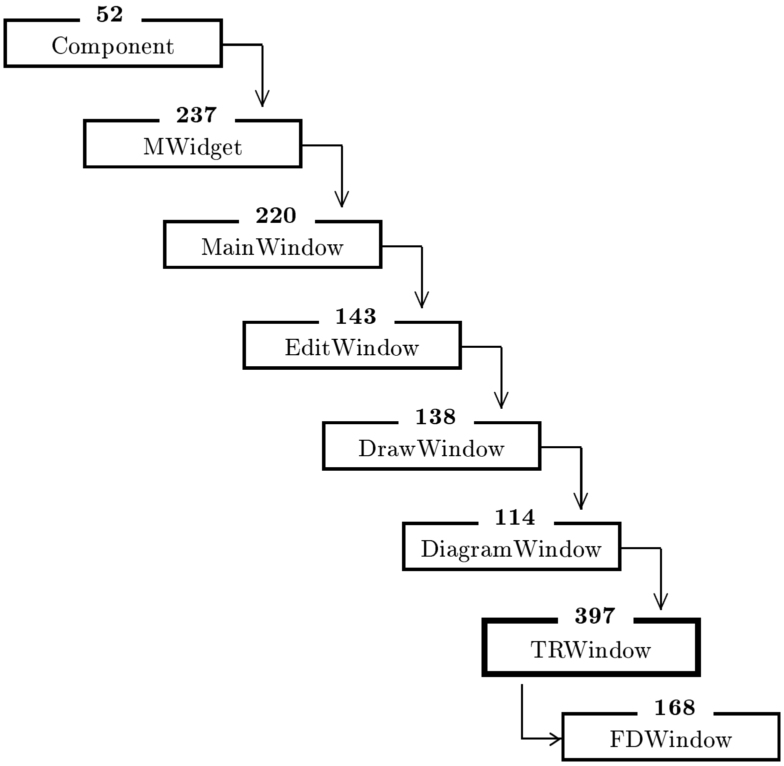
	TRView (TRViewer *t)
void	**AddShape** (GShape *s)
void	**RemoveShape** (GShape *s)

397

class **TRWindow** : public DiagramWindow

generic textual tree edit window class.

Inheritance



Public Members

	TRWindow (const char *name)
void	**SetLayout** (bool b)

Private Members

int	**GetTool** ()

DiagramViewer*
CreateViewer (Config *c)

Graph* **CreateGraph** ()

Diagram* **CreateDiagram** (Config *c, DiagramViewer *v, Graph *g)

void **DetermineIcons** ()

Widget **editButton**

Widget **treeButton**

Widget **CreateTiles** (Widget parent)

Widget **CreateLayoutTiles** (Widget parent)

Widget **CreateMenuBar** (Widget parent)

static const char*
TR_NODE_NAMES []

static const Bitmap
TR_NODE_ICONS []

static const char*
TR_EDGE_NAMES []

static const Bitmap
TR_EDGE_ICONS []

void **SetSensitive** (Menu *menu, const char *label, bool b)

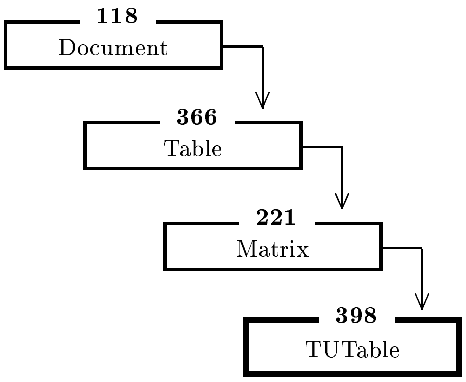
static void
LayoutCB (Widget w, XtPointer num, XtPointer)

398

class **TUTable** : public Matrix

transaction use table class.

Inheritance



Public Members

	TUTable (Config *c, TUWindow *t, TUViewer *v)
void	**CheckDocument** ()

Private Members

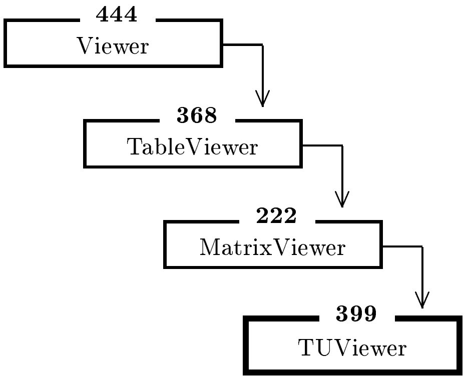
bool	**CheckTextString** (Cell *c, const string *text)

399

class **TUViewer** : public MatrixViewer

transaction use table viewer class.

Inheritance



Public Members

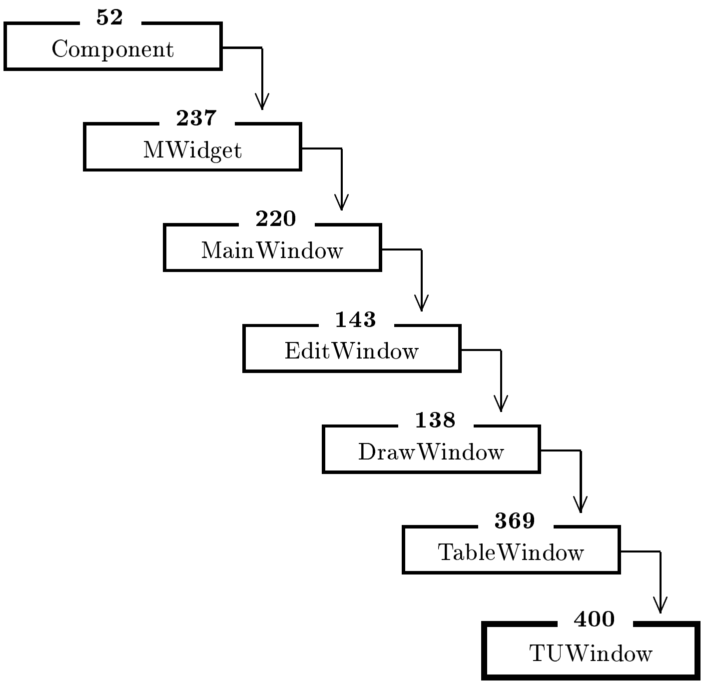
	TUViewer (Config *c, TableWindow *t)
void	**Initialize** ()

400

class **TUWindow** : public TableWindow

transaction use table edit window class.

Inheritance



Public Members

	TUWindow (const char *name)
TableViewer*	**CreateViewer** ()
Table*	**CreateTable** ()

Private Members

int	**GetTool** ()

401

class **TypeName**

Association between a code and a name.

Public Members

int **code**

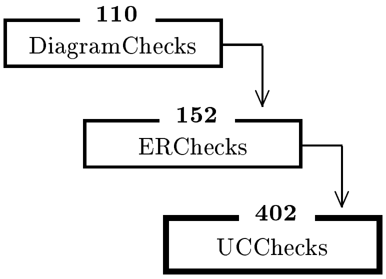
const char* **name**

402

class **UCChecks** : public ERChecks

implements checks for soft constraints on UC diagrams.

Inheritance



Public Members

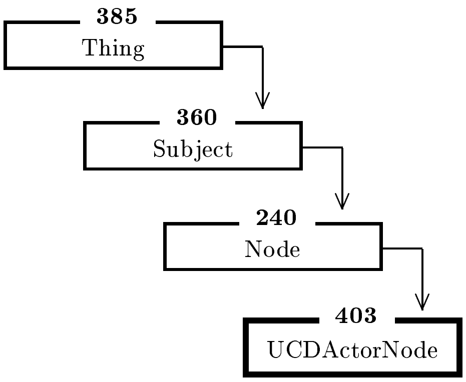
UCChecks (Diagram *d, Graph *g)

403

class **UCDActorNode** : public Node

UCD actor node class.

Inheritance



Public Members

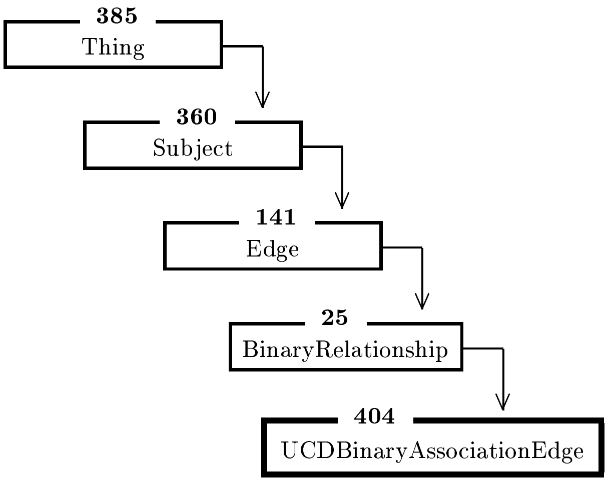
	UCDActorNode (UCGraph *)
Subject*	**Clone** ()
Subject::NameErrType	**SetName** (const string *name)
int	**GetClassType** () const

404

class **UCDBinaryAssociationEdge** : public BinaryRelationship

binary association edge class in an use-case diagram

Inheritance



Public Members

	UCDBinaryAssociationEdge (UCGraph *g, Subject *n1, Subject *n2)
Subject*	**Clone** ()
int	**GetClassType** () const

Private Members

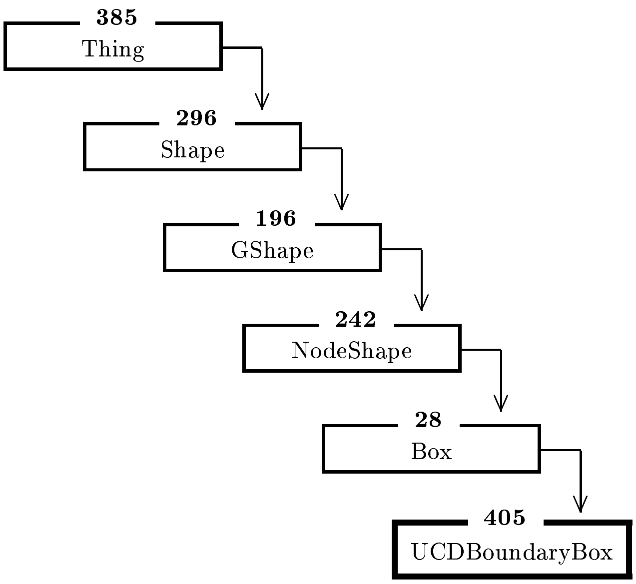
bool	**SetConstraint** (string *cons, const string *name)

405

class **UCDBoundaryBox** : public Box

Box for representing system boundary in a use-case diagram.

Inheritance



Public Members

	UCDBoundaryBox (ShapeView *v, Grafport *g, int x, int y, int w = 3*BOX_WIDTH, int h = 12*BOX_HEIGHT)
Shape*	**Clone** ()
int	**GetClassType** () const
void	**SetPosition** (const Point *p, bool snap=True)
void	**AdjustSize** ()
void	**SetSize** (int w, int h)
void	**SetTextShape** ()

Private Members

static const int **TEXTMARGIN**

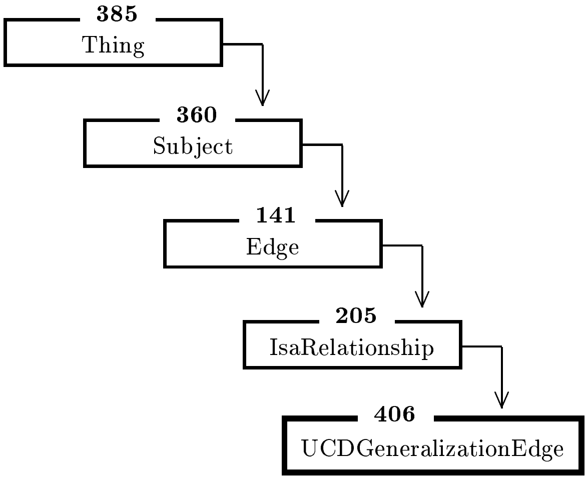
void **CalcPositionLabels** ()

406

class **UCDGeneralizationEdge** : public IsaRelationship

Generalization edge in an use-case diagram.

Inheritance



Public Members

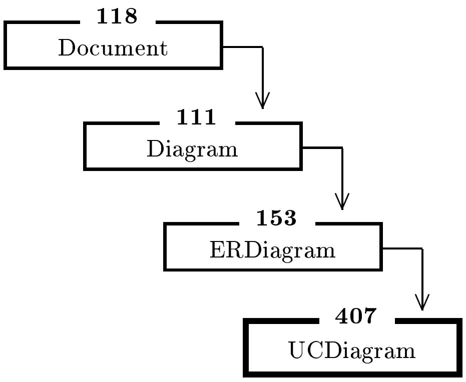
	UCDGeneralizationEdge (UCGraph *g, Subject *n1, Subject *n2)
Subject*	**Clone** ()
Subject::NameErrType	**SetName** (const string *s)
int	**GetClassType** () const

407

class **UCDiagram** : public ERDiagram

UML Use Case diagram class.

Inheritance



Public Members

	UCDiagram (Config *, UCWindow *, UCViewer *, UCGraph *)
virtual	**~UCDiagram** ()
Node*	**CreateNode** ()
Edge*	**CreateEdge** (Subject *n1, Subject *n2)
NodeShape*	**CreateNodeShape** (Node *node, int x, int y)
Line*	**CreateLine** (Edge *edge, GShape *fromShape, GShape *toShape, List<Point *> *line)
void	**UpdateNodeType** (int n)
void	**UpdateEdgeType** (int n)
bool	**SetText** (TextShape *t, const string *s)
void	**CheckDocument** () *check the soft constraints on a UCD.*

Protected Members

Thing* **CreateThing** (int classNr)

UCChecks*
GetUCChecks ()

bool **CheckEdgeConstraints** (Subject *subj1, Subject *subj2)

bool **CheckIsaLoop** (Edge *edge)

Private Members

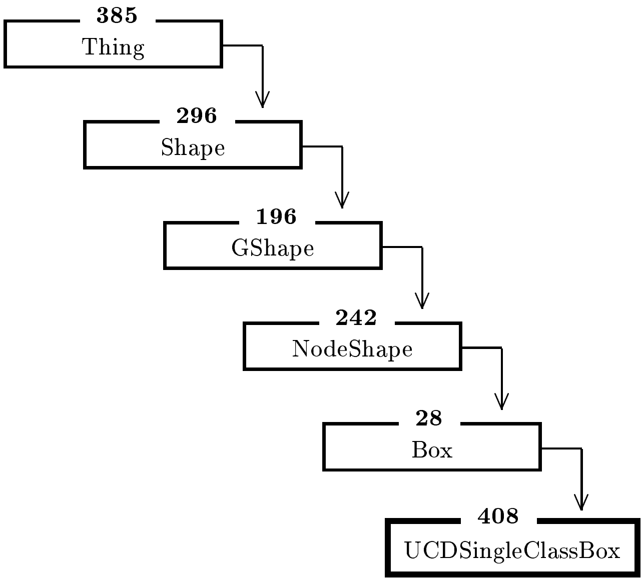
UCChecks*
ucChecks

408

class **UCDSingleClassBox** : public Box

Box for representing actor classes in a UCD.

Inheritance



Public Members

	UCDSingleClassBox (ShapeView *v, Grafport *g, int x, int y, int w = BOX_WIDTH, int h = BOX_HEIGHT)
	UCDSingleClassBox (const UCDSingleClassBox &s)
	UCDSingleClassBox (const StickMan &s)
	~UCDSingleClassBox ()
Shape*	**Clone** ()
int	**GetClassType** () const
void	**SetSize** (int w, int h)

void **SetPosition** (const Point *p, bool snap)

void **SetFont** (XFont *ft)

void **SetTextColor** (const string *color)

void **SetGrafport** (Grafport *g)

void **SetView** (ShapeView *v)

TextShape* **ChooseTextShape** (int x, int y)

void **AdjustSize** ()

bool **SetAssocSubject** (AssocList *)

Private Members

void **DrawTextShapes** ()

int **RequiredHeight** ()

int **RequiredWidth** ()

void **CalcPositionLabels** ()

void **CalcPositionStereotypeLabel** ()

static const int **TEXTMARGIN**

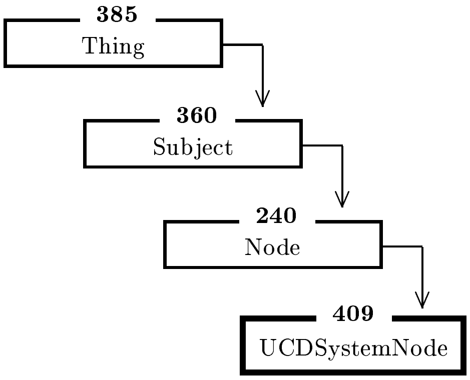
TextShape* **stereotypeLabel**

409

class **UCDSystemNode** : public Node

use case (UC) system node class.

Inheritance



Public Members

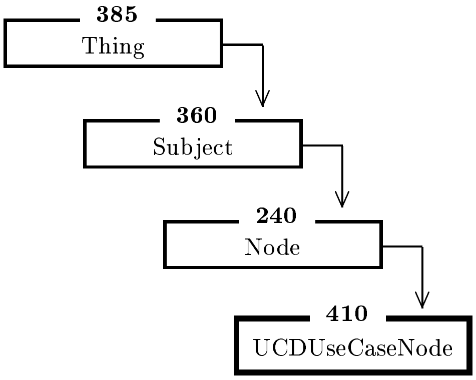
	UCDSystemNode (UCGraph *)
Subject*	**Clone** ()
int	**GetClassType** () const

410

class **UCDUseCaseNode** : public Node

use case (UC) node class.

Inheritance



Public Members

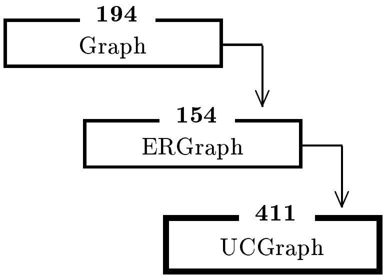
	UCDUseCaseNode (UCGraph *)
Subject*	**Clone** ()
Subject::NameErrType	**SetName** (const string *name)
int	**GetClassType** () const

411

class **UCGraph** : public ERGraph

UML use case graph class

Inheritance



Public Members

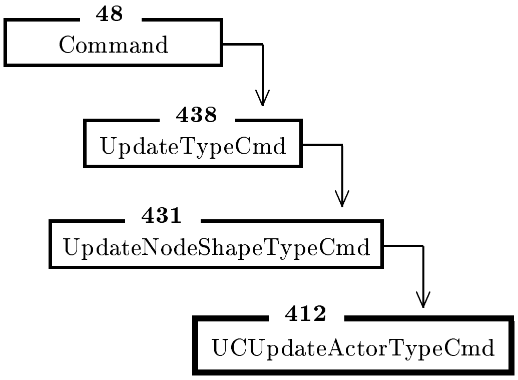
	UCGraph ()
void	**InitConnections** ()

412

class **UCUpdateActorTypeCmd** : public UpdateNodeShapeTypeCmd

command for changing actor types

Inheritance



Public Members

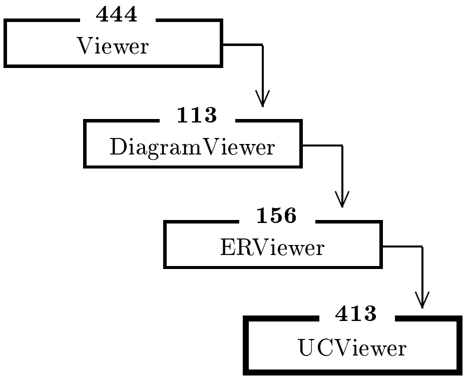
	UCUpdateActorTypeCmd (ShapeView *v, List<int *> *fromTypes, int toType)
const char*	**GetName** () const
Thing*	**Transform** (Thing *thing)

413

class **UCViewer** : public ERViewer

class-relationship viewer class.

Inheritance



Public Members

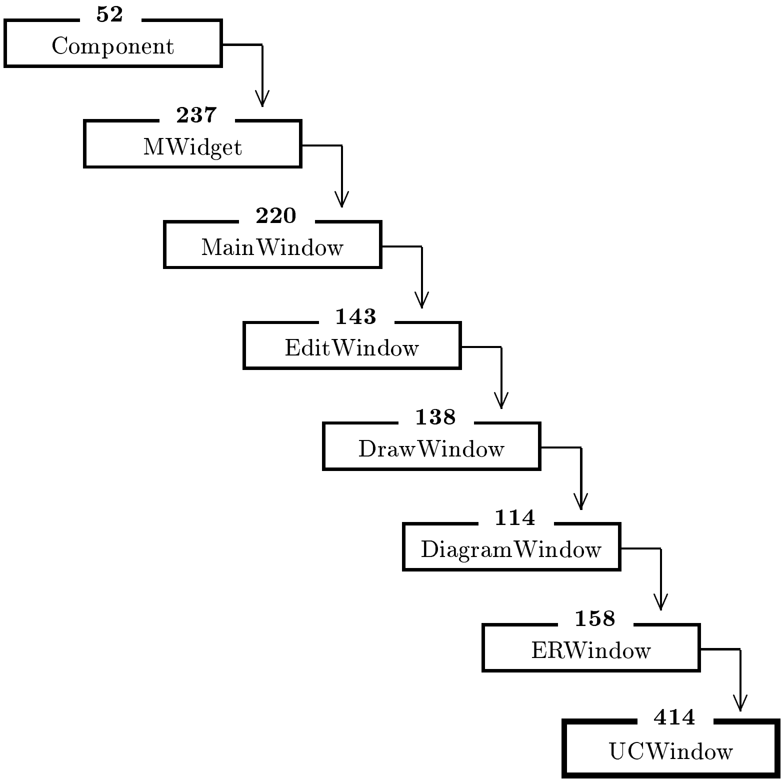
	UCViewer (Config *c, UCWindow * w)	
void	**UpdateNodeShapeType** (int)	*change actor type of selected object types.*

414

class **UCWindow** : public ERWindow

use case edit window class.

Inheritance



Public Members

	UCWindow (const char *name)
virtual	**~UCWindow** ()

Private Members

int	**GetTool** ()

DiagramViewer*
CreateViewer (Config *config)

Graph* **CreateGraph** ()

Diagram* **CreateDiagram** (Config *c, DiagramViewer *v, Graph *g)

void **DetermineIcons** ()

MenuItem*
diagramUCBoxItems

void **InitMenuItems** ()

static const char*
UC_NODE_NAMES []

static const Bitmap
UC_NODE_ICONS []

static const char*
UC_EDGE_NAMES []

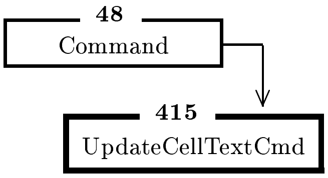
static const Bitmap
UC_EDGE_ICONS []

415

class **UpdateCellTextCmd** : public Command

command class for updating cell texts.

Inheritance



Public Members

UpdateCellTextCmd (Table *t, TableViewer *v,
Cell *c,
const string *newtext)

virtual **~UpdateCellTextCmd** ()

void **Execute** ()

void **UnExecute** ()

const char*
GetName () const

Private Members

Cell* **cell**

string* **newText**

string* **oldText**

416

class **UpdateCellTextFontCmd** : public Command

abstract command class for updating cell text fonts.

Inheritance

48
Command

416
UpdateCellTextFontCmd

Public Members

	UpdateCellTextFontCmd (Table *, TableViewer *v, int f, int s, int p)
virtual	**~UpdateCellTextFontCmd** ()
void	**UnExecute** ()
void	**Execute** ()
const char*	**GetName** () const

Protected Members

void	**SaveCell** (Cell *c) *save cell and its old font.*
void	**GetUpdateCells** ()

Private Members

TableViewer*	**tviewer**
List <Cell *> *	**cells**
List <XFont *> *	

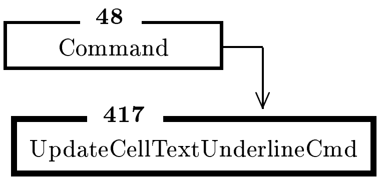
	oldFonts
XFont*	**GetNewFont** (Cell *cell)
int	**family**
int	**size**
int	**style**

417

class **UpdateCellTextUnderlineCmd** : public Command

command class for updating underlining of cell texts.

Inheritance



Public Members

	UpdateCellTextUnderlineCmd (Table *t, TableViewer *v)
virtual	**~UpdateCellTextUnderlineCmd** ()
void	**Execute** ()
void	**UnExecute** ()
const char*	**GetName** () const

Private Members

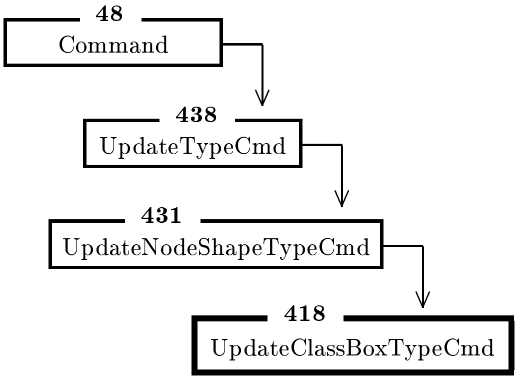
List <CellText *> *	**cellTexts**

418

class **UpdateClassBoxTypeCmd** : public UpdateNodeShapeTypeCmd

command for changing class box types

Inheritance



Public Members

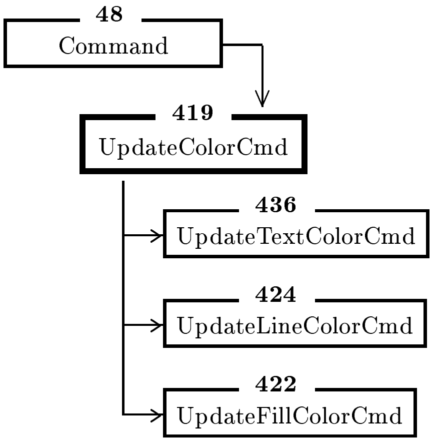
	UpdateClassBoxTypeCmd (ShapeView *v, List<int *> *fromTypes, int toType)
const char*	**GetName** () const
Thing*	**Transform** (Thing *thing)

419

class **UpdateColorCmd** : public Command

command class for updating colors of shapes.

Inheritance



Public Members

enum **Type**

UpdateColorCmd (ShapeView *view, const string *color, Type t)

virtual **~UpdateColorCmd** ()

void **Execute** ()

void **UnExecute** ()

const char* **GetName** () const

List <GShape *> * **GetShapes** () const

List <string *> * **GetOldColors** () const

Private Members

List <GShape *> *
shapes

List <string *> *
oldColors

ShapeView*
view

Type **type**

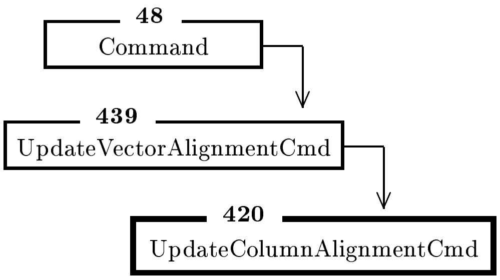
string **color**

420

class **UpdateColumnAlignmentCmd** : public UpdateVectorAlignmentCmd

command class for updating text alignments.

Inheritance



Public Members

UpdateColumnAlignmentCmd (Table *t, TableViewer *v, TextAlign::Type a)

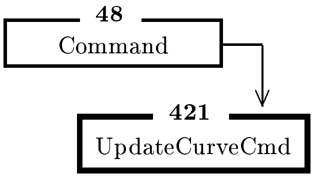
const char* **GetName** () const

421

class **UpdateCurveCmd** : public Command

command class for updating lines styles of shapes.

Inheritance



Public Members

	UpdateCurveCmd (ShapeView *view)
virtual	**~UpdateCurveCmd** ()
void	**Execute** ()
void	**UnExecute** ()
const char*	**GetName** () const

Private Members

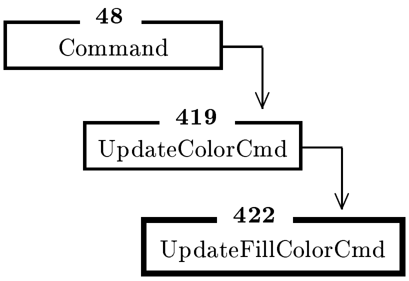
List <Line *> *	**lines**
ShapeView*	**view**

422

class **UpdateFillColorCmd** : public UpdateColorCmd

command class for updating fill colors of shapes.

Inheritance



Public Members

UpdateFillColorCmd (ShapeView *view,
const string *color,
FillStyle::Type f)

~UpdateFillColorCmd ()

void **Execute** ()

void **UnExecute** ()

const char*
GetName () const

Private Members

List <FillStyle::Type> *
oldFillStyles

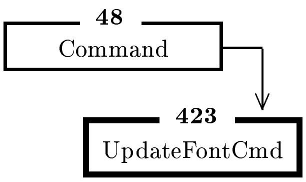
FillStyle::Type
fillStyle

423

class **UpdateFontCmd** : public Command

command class for updating fonts.

Inheritance



Public Members

UpdateFontCmd (ShapeView *view, int family, int style, int size)

virtual **~UpdateFontCmd** ()

void **UnExecute** ()

void **Execute** ()

const char*
GetName () const

Protected Members

void **SaveShape** (GShape *s)
save shape and its old font and size.

XFont* **GetNewFont** (GShape *s)

void **GetUpdateShapes** ()

Private Members

List <GShape *> *
shapes

List <Point *> *

oldSizes

List <XFont *> *
oldFonts

ShapeView*
view

int **family**

int **style**

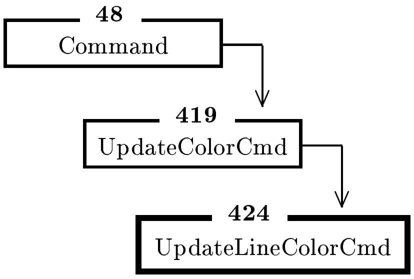
int **size**

424

class **UpdateLineColorCmd** : public UpdateColorCmd

command class for updating line colors of shapes.

Inheritance



Public Members

UpdateLineColorCmd (ShapeView *view, const string *color)

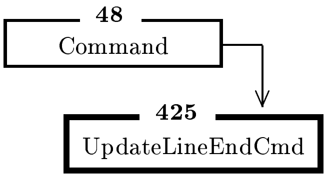
const char*
GetName () const

425

class **UpdateLineEndCmd** : public Command

command class for updating line ends of lines.

Inheritance



Public Members

UpdateLineEndCmd (ShapeView *view,
LineEnd::Type e1,
LineEnd::Type e2)

virtual **~UpdateLineEndCmd** ()

void **Execute** ()

void **UnExecute** ()

const char*
GetName () const

Private Members

LineEnd::Type
end1

LineEnd::Type
end2

List <Line *> *
lines

List <LineEnd::Type> *
oldEnds1

List <LineEnd::Type> *
oldEnds2

ShapeView*

view

426

class **UpdateLineStyleCmd** : public Command

command class for updating lines styles of shapes.

Inheritance

48
Command

426
UpdateLineStyleCmd

Public Members

UpdateLineStyleCmd (ShapeView *view, LineStyle::Type s)

virtual **~UpdateLineStyleCmd** ()

void **Execute** ()

void **UnExecute** ()

const char*
GetName () const

Private Members

LineStyle::Type
style

List <GShape *> *
shapes

List <LineStyle::Type> *
oldStyles

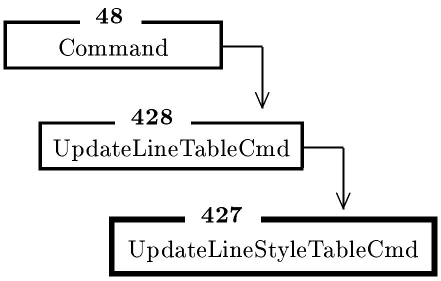
ShapeView*
view

427

class **UpdateLineStyleTableCmd** : public UpdateLineTableCmd

command class for updating line style in a table

Inheritance



Public Members

UpdateLineStyleTableCmd (Table *t, TableViewer *v, Side::Type side, LineStyle::Type style)

virtual **~UpdateLineStyleTableCmd** ()

void **Execute** ()

void **UnExecute** ()

const char* **GetName** () const

Private Members

List <LineStyle::Type> * **oldStyles**

LineStyle::Type **style**

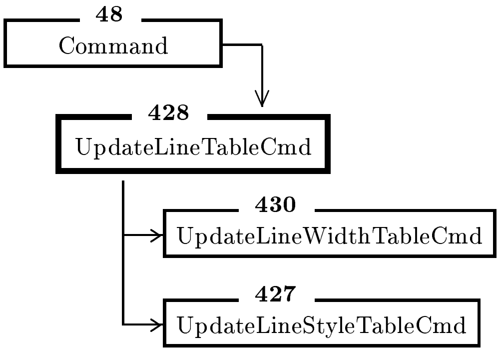
void **AddLine** (LinePiece *line)

428

class **UpdateLineTableCmd** : public Command

abstract command class for updating line attributes in a table

Inheritance



Public Members

UpdateLineTableCmd (Table *t, TableViewer *v, Side::Type side)

virtual **~UpdateLineTableCmd** ()

List <LinePiece *> * **GetLines** ()

Private Members

List <LinePiece *> * **lines**

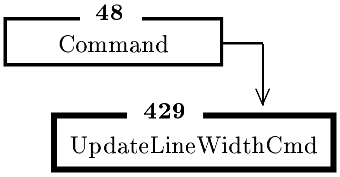
void **AddLine** (LinePiece *line)

429

class **UpdateLineWidthCmd** : public Command

command class for updating line widths of shapes.

Inheritance



Public Members

	UpdateLineWidthCmd (ShapeView *view, unsigned lineWidth)
virtual	**~UpdateLineWidthCmd** ()
void	**Execute** ()
void	**UnExecute** ()
const char*	**GetName** () const

Private Members

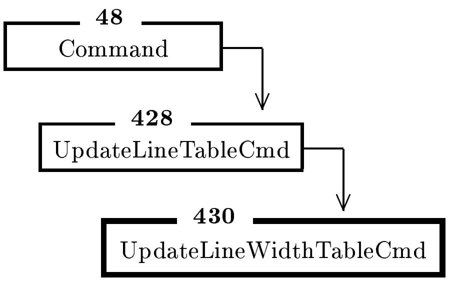
unsigned	**lineWidth**
List <GShape *> *	**shapes**
List <unsigned> *	**oldWidths**
ShapeView*	**view**

430

class **UpdateLineWidthTableCmd** : public UpdateLineTableCmd

command class for updating line widths in a table

Inheritance



Public Members

UpdateLineWidthTableCmd (Table *t, TableViewer *v, Side::Type side, unsigned width)

virtual **~UpdateLineWidthTableCmd** ()

void **Execute** ()

void **UnExecute** ()

const char* **GetName** () const

Private Members

List <unsigned> * **oldWidths**

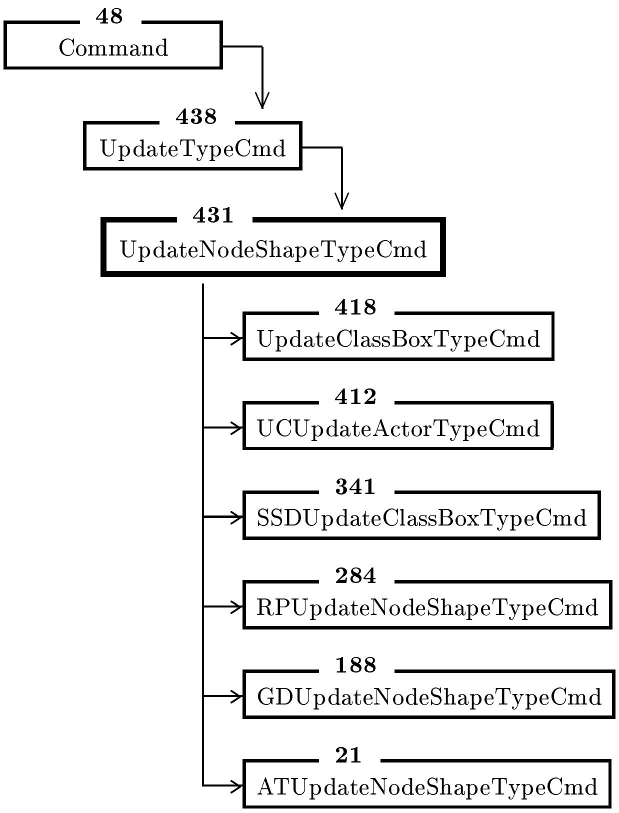
unsigned **width**

431

class **UpdateNodeShapeTypeCmd** : public UpdateTypeCmd

command for changing node shape types.

Inheritance



Public Members

UpdateNodeShapeTypeCmd (ShapeView *v, List<int *> *fromTypes, int toType)

const char*

GetName () const

Protected Members

void **Reverse** (Thing *oldThing, Thing *newThing)

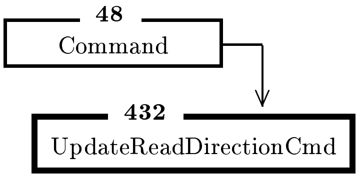
void **UpdateLines** (GShape *oldShape, GShape *newShape)

432

class **UpdateReadDirectionCmd** : public Command

command class for updating read direction arrows in lines.

Inheritance



Public Members

	UpdateReadDirectionCmd (ShapeView *view, ReadDirection::Type direction)
virtual	**~UpdateReadDirectionCmd** ()
void	**Execute** ()
void	**UnExecute** ()
const char*	**GetName** () const

Private Members

ReadDirection::Type
direction

List <ReadDirection::Type> *
oldDirections

List <Line *> *
lines

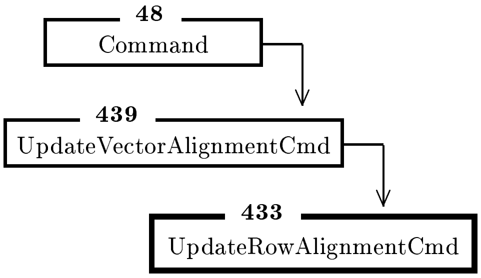
ShapeView*
view

433

class **UpdateRowAlignmentCmd** : public UpdateVectorAlignmentCmd

command class for updating text alignments.

Inheritance



Public Members

UpdateRowAlignmentCmd (Table *t,
TableViewer *v,
TextAlign::Type a)

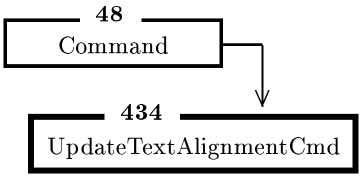
const char*
GetName () const

434

class **UpdateTextAlignmentCmd** : public Command

command class for updating text alignments.

Inheritance



Public Members

	UpdateTextAlignmentCmd (ShapeView *view, TextAlign::Type a)
virtual	**~UpdateTextAlignmentCmd** ()
void	**Execute** ()
void	**UnExecute** ()
const char*	**GetName** () const

Private Members

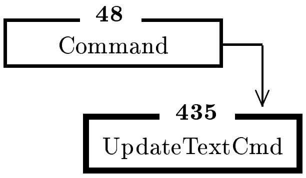
TextAlign::Type	**alignment**
List <GShape *> *	**shapes**
List <TextAlign::Type> *	**oldAlignments**
ShapeView*	**view**

435

class **UpdateTextCmd** : public Command

command class for updating cell texts.

Inheritance



Public Members

	UpdateTextCmd (TextShape *t, const string *newtext, const string *oldtext)
virtual	**~UpdateTextCmd** ()
void	**Execute** ()
void	**UnExecute** ()
void	**ReExecute** ()
const char*	**GetName** () const

Private Members

Diagram*	**diagram**
GShape*	**parent**
TextShape*	**textShape**
string*	**newText**
string*	**oldText**
int	**oldWidth**
int	**oldHeight**

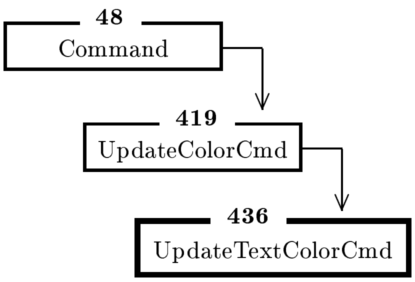
Point	**oldParentPosition**
Point	**textShapePosition**
void	**Executable** ()

436

class **UpdateTextColorCmd** : public UpdateColor-Cmd

command class for updating text colors of shapes.

Inheritance



Public Members

UpdateTextColorCmd (ShapeView *view, const string *color)

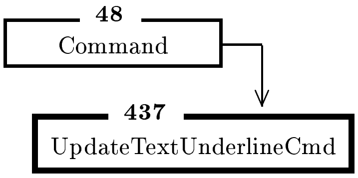
const char*
GetName () const

437

class **UpdateTextUnderlineCmd** : public Command

command class for updating underlining of text shapes.

Inheritance



Public Members

	UpdateTextUnderlineCmd (ShapeView *view)
virtual	**~UpdateTextUnderlineCmd** ()
void	**Execute** ()
void	**UnExecute** ()
const char*	**GetName** () const

Private Members

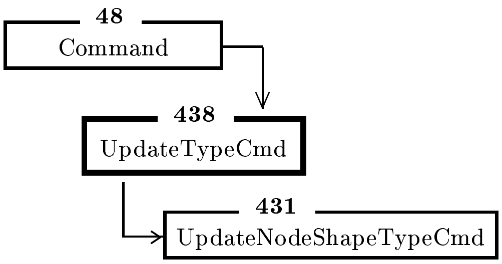
List <TextShape *> *
textShapes

438

class **UpdateTypeCmd** : public Command

command for converting things to a different type

Inheritance



Public Members

UpdateTypeCmd (ShapeView *v, List<int *> *fromTypes, int toType)

virtual **~UpdateTypeCmd** ()

void **Execute** ()

void **UnExecute** ()

const char* **GetName** () const

Protected Members

virtual Thing* **Transform** (Thing *thing)

virtual void **Reverse** (Thing *oldThing, Thing *newThing)

List <Thing *> * **oldThings**

List <Thing *> *

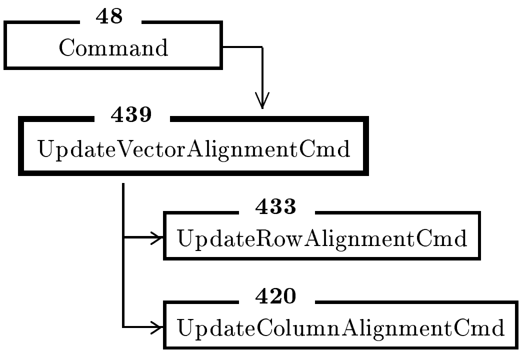
	newThings
int	**toType**
List <int*>	**fromTypes**
ShapeView*	**view**
Diagram*	**diagram**
void	**TransformAll** ()

439

class **UpdateVectorAlignmentCmd** : public Command

abstract command class for updating text alignments.

Inheritance



Public Members

UpdateVectorAlignmentCmd (Table *t, TableViewer *v, TextAlign::Type a)

virtual **~UpdateVectorAlignmentCmd** ()

void **Execute** ()

void **UnExecute** ()

Protected Members

List <TextAlign::Type> * **GetOldAlignments** ()

List <CellVector *> * **GetVectors** ()

Private Members

TextAlign::Type
alignment

List <CellVector *> *
vectors

List <TextAlign::Type> *
oldAlignments

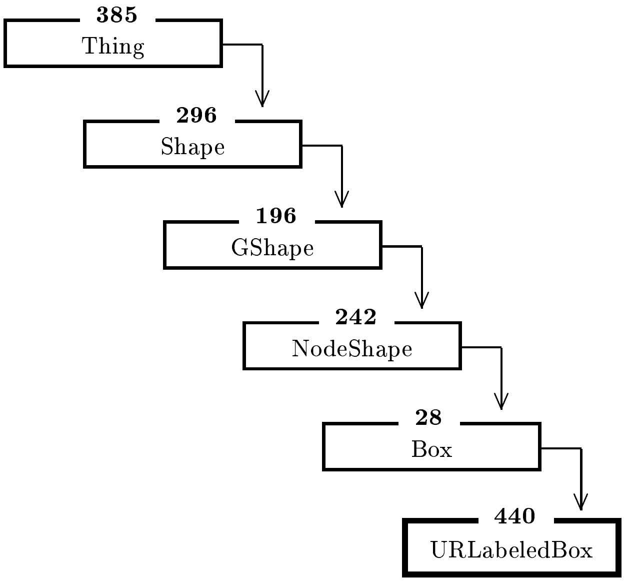
TableViewer*
tviewer

440

class **URLabeledBox** : public Box

box with extra label in Upper Right corner (PSD).

Inheritance



Public Members

	URLabeledBox (ShapeView *v, Grafport *g, int x, int y, int w = BOX_WIDTH, int h = BOX_HEIGHT)
	URLabeledBox (const URLabeledBox &c)
virtual	**~URLabeledBox** ()
Shape*	**Clone** ()
int	**GetClassType** () const
void	**SetPosition** (const Point *p, bool b) *adjusts the label to a new place*
void	**SetSize** (int w, int h)

void **SetFont** (XFont *ft)

void **SetTextColor** (const string *color)

TextShape*
ChooseTextShape (int x, int y)
chooses the name or the UR label.

bool **HasTextShape** (TextShape *t) const

void **UpdateURLabel** (const string *s)
update string of UR label.

void **SetURLabel** (const string *s)

bool **SetAssocSubject** (AssocList *)

void **SetTextShape** ()

void **SetGrafport** (Grafport *g)

void **SetView** (ShapeView *v)

Private Members

void **DrawShape** ()

void **CalcPositionLabels** ()

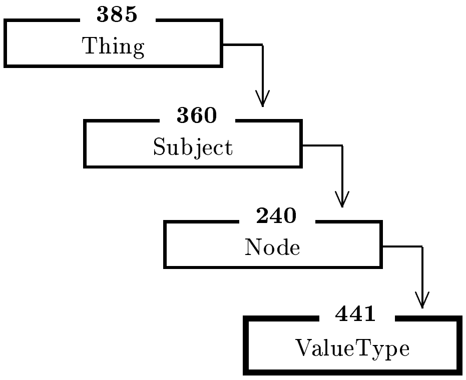
TextShape*
urLabel

441

class **ValueType** : public Node

value type (ER) node class.

Inheritance



Public Members

	ValueType (ERGraph *g)
int	**GetClassType** () const
Subject*	**Clone** ()

442

class **Version**

Version information of the software.

Public Members

static const char
TOOLKIT_VERSION []
the current toolkit version as string.

static const double
FILE_FORMAT *the version of the (default) generated file format.*

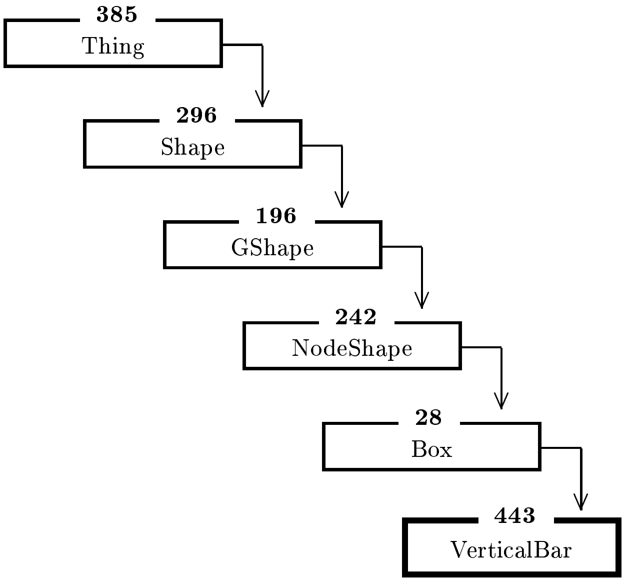
static const char
ABOUT_TEXT []
the text displayed "about" the software.

443

class **VerticalBar** : public Box

box class with undrawn top and bottom sides.

Inheritance



Public Members

static const int
BAR_WIDTH

static const int
BAR_HEIGHT

VerticalBar (ShapeView *v, Grafport *g, int x, int y, int w = BAR_WIDTH, int h = BAR_HEIGHT)

VerticalBar (const NodeShape &c)

Shape* **Clone** ()

void **DrawOutLine** (const Point *center)

		draw bar-form dotted outline. snaps to the grid.
int	**GetClassType** () const	
static const int	**BAR_HANDLES**	

Protected Members

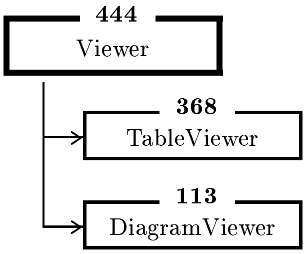
void	**DrawShape** ()	*draws a vertical bar.*

444

class **Viewer**

(abstract) viewer class, responsible for the contents of the drawing area.

Inheritance



Public Members

Viewer (Config *c, DrawWindow *d)

virtual **~Viewer** ()

virtual void
Initialize ()

Scaler* **GetScaler** () const

Command*
GetLastCmd ()

DrawWindow*
GetMainWindow () const

Grafport* **GetGrafport** () const

Printer* **GetPrinter** () const

void **SetPrinter** (Printer *p)

virtual void
SetGrafport (Grafport *g)

NToggleListDialog*
GetLineWidthDialog ()

NToggleListDialog*
GetLineStyleDialog ()

NToggleListDialog*

GetTextAlignmentDialog ()

FontChooserDialog*
GetFontChooserDialog ()

FileSelectionDialog*
GetPrintFileSelectionDialog ()

ColorChooserDialog*
GetColorChooserDialog ()

int **Scale** (int x) const
forwarding to scaler object.

Point **Scale** (const Point *p) const

int **ScaleCorrect** (int x) const

Point **ScaleCorrect** (const Point *p) const

virtual void
Draw () *Draw all elements of the viewer in grafport.*

virtual void
Undraw () *Erase all elements from grafport.*

void **Undo** () *Undo last command.*

void **Redo** () *Redo last command.*

void **Refresh** () *Clear grafport contents, and redraw all viewer elements.*

virtual void
Redraw () *redraw contents of grafport.*

virtual void
DeselectAll () *deselect all elements in viewer;*

virtual void
Select (int x, int y)
Perform the function of the 1st button function of the mouse

virtual void
Adjust (int x, int y)
Perform the function of the 2nd button function of the mouse.

virtual void
Activate () *show that viewer is active.*

virtual void

Deactivate () *show that viewer is not active.*

virtual void
TextModeOff () *stop being in editing mode.*

void **SetDefaultLineWidth** (unsigned d)

int **GetDefaultLineWidth** ()

void **UpdateLineWidth** ()

void **DefaultLineWidth** ()

void **ShowDefaultLineWidth** ()

void **ShowConfigLineWidth** ()

virtual void
UpdateLineWidthSelection (unsigned i)

void **SetDefaultLineStyle** (LineStyle::Type l)

LineStyle::Type
GetDefaultLineStyle ()

void **UpdateLineStyle** ()

void **DefaultLineStyle** ()

void **ShowDefaultLineStyle** ()

void **ShowConfigLineStyle** ()

virtual void
UpdateLineStyleSelection (LineStyle::Type s)

void **SetDefaultLineColor** (const string *s)

const string*
GetDefaultLineColor ()

void **SetDefaultTextColor** (const string *s)

const string*
GetDefaultTextColor ()

void **SetDefaultFillColor** (const string *s)

const string*
GetDefaultFillColor ()

FillStyle::Type
GetDefaultFillStyle () const

void **SetDefaultFillStyle** (FillStyle::Type l)

virtual void

UpdateLineColorSelection (const string *s)

virtual void **UpdateTextColorSelection** (const string *s)

virtual void **UpdateFillColorSelection** (const string *s, FillStyle::Type f)

void **UpdateLineColor** ()

void **DefaultLineColor** ()

void **ShowDefaultLineColor** ()

void **ShowConfigLineColor** ()

void **UpdateTextColor** ()

void **DefaultTextColor** ()

void **ShowDefaultTextColor** ()

void **ShowConfigTextColor** ()

void **UpdateFillColor** ()

void **DefaultFillColor** ()

void **ShowDefaultFillColor** ()

void **ShowConfigFillColor** ()

XFont* **GetDefaultFont** () const

void **SetDefaultFontAttributes** (int f, int s, int p)

virtual void **SetDefaultFont** (XFont *ft)

void **UpdateFont** ()

void **DefaultFont** ()

void **ShowDefaultFont** ()

void **ShowConfigFont** ()

virtual void **UpdateFontSelection** (int family, int style, int size)

XFont* **GetFont** (int family, int style, int size)

XFont* **GetFont** (const char *description)

void **UpdateTextAlignment** ()

void **DefaultTextAlignment** ()

void **ShowDefaultTextAlignment** ()

void **ShowConfigTextAlignment** ()

TextAlign::Type **GetDefaultTextAlignment** () const

void **SetDefaultTextAlignment** (TextAlign::Type a)

virtual void **UpdateTextAlignmentSelection** (TextAlign::Type a)

void **EditText** (int key)

bool **IsAutoResize** () const

void **SetAutoResize** (bool s)

bool **IsInlineEdit** () const

void **SetInlineEdit** (bool s)

bool **IsPasting** () const

void **SetPasting** (bool b)

bool **IsZigZag** () const

void **SetZigZag** (bool b)

bool **IsEditing** () const

virtual void **MovingPointer** (int x, int y)
change cursor to show which command can be performed.

void **DoubleClick** () *not yet implemented (double clicked button).*

void **DoFunctionKey** (int)
not yet implemented (function keys).

void **NewCommand** (Command *newCmd)
enable execution of last command as newCmd.

void **AbortCommand** ()
abort execution of last command.

void **ExecuteCommand** ()
execute last command.

void **UndoCommand** ()

undo execution of last command.

void **RedoCommand** ()
redo execution of last command.

void **DeleteCommands** ()
delete command history, make undo impossible.

virtual Command*
Drag (int x, int y)
Perform the function of 1st button drag.

virtual Command*
Connect (int x, int y)
Perform the function of 2nd button drag.

virtual void
KeyTyped (int x, int y, int c)
Perform the function of keypress 'c'.

virtual void
CalcSizeElements (Point &topLeft, Point &bottomRight)
calculate covered area of elements in grafport.

virtual void
CalcSizeSelection (Point &topLeft, Point &bottomRight)
calculate covered area of selected elements in grafport.

void **GetPageSize** (double &width, double &height)
return page size (pixels).

void **Save** (OutputFile *f)
Save view attributes to file.

bool **Load** (InputFile *f, double format)
Load view attributes from file.

bool **Check** (InputFile *f, double format)
Check but not load view attributes from file.

static const int
MIN_LINE_WIDTH

static const int

MAX_LINE_WIDTH

Protected Members

void	**ClearGrafport** ()	*Clear contents of grafport.*
InlineEditor*	**GetInlineEditor** () const	
void	**SetInlineEditor** (InlineEditor *i)	
void	**SetEditing** (bool b)	

Private Members

DrawWindow*	**mainwindow**	*the main edit window.*
Grafport*	**grafport**	*the grafport.*
Scaler*	**scaler**	*for scaling coordinates.*
enum		*the commands that were executed.*
Command*	**cmdHist** [MAXHISTSIZE]	
int	**cmdHistSize**	
int	**lastCmd**	*the index of the last command that is executed.*
bool	**lastCmdIllegal**	*when NewCommand(0) is called, execute can not take place.*
Config*	**config**	
XFont*	**defaultFont**	
List <XFont *> *	**fontList**	
bool	**editing**	*being in editing mode.*
bool	**pasting**	*busy pasting.*
bool	**zigZag**	*being zigzagging (busy drawing lines).*
InlineEditor*	**edit**	
Printer*	**printer**	

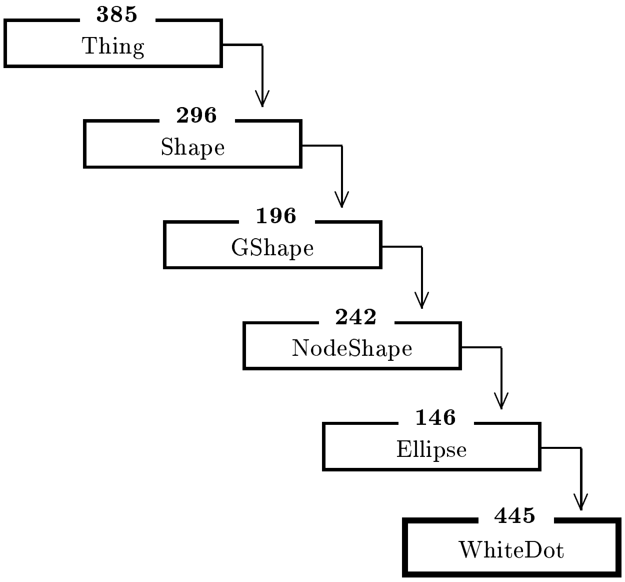
bool	**autoResizing**	*if true, shapes automatically adapted to fit a text.*
bool	**inlineEditor**	*text editing happens in drawing area (True) or in separate window.*
unsigned	**defaultLineWidth**	*newly created figures get this line width.*
LineStyle::Type	**defaultLineStyle**	*newly created figures get this line type.*
TextAlign::Type	**defaultTextAlignment**	*newly created text get this text alignment.*
string	**defaultLineColor**	*newly created lines get this color*
string	**defaultTextColor**	*newly created texts get this color*
string	**defaultFillColor**	*newly created filled items get this color*
FillStyle::Type	**defaultFillStyle**	
NToggleListDialog*	**lineWidthDialog**	
NToggleListDialog*	**lineStyleDialog**	
NToggleListDialog*	**textAlignmentDialog**	
FontChooserDialog*	**fontChooserDialog**	
ColorChooserDialog*	**colorChooserDialog**	

445

class **WhiteDot** : public Ellipse

Mini ellipse class, drawn as a white dot.

Inheritance



Public Members

static const int
WHITE_DOT_WIDTH

WhiteDot (ShapeView *v, Grafport *g, int x, int y, int w = WHITE_DOT_WIDTH, int h = WHITE_DOT_WIDTH)

WhiteDot (const NodeShape &c)

Shape* **Clone** ()

int **GetClassType** () const

Private Members

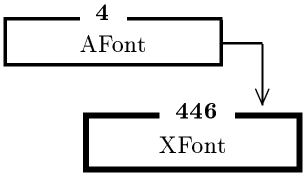
void **Init** ()

446

class **XFont** : public AFont

A class for X fonts.

Inheritance



Public Members

	XFont (const string *foundry, int name, int style, int size, bool scalable)	*Creates new X font with the specified name, style and point size.*
	~XFont ()	
void	**Scale** (double factor)	*scale the X font (if scalable) with factor.*
static int	**FindFamily** (const char *desc)	*Gets family name from X font description.*
static int	**FindStyle** (const char *desc)	*Gets font style from X font description.*
static int	**FindSize** (const char *desc)	*Gets point size from X font description.*
int	**GetLeading** ()	*Gets the standard leading, or line spacing, for the font.*
int	**GetAscent** ()	*Gets font ascent (distance from base line to top of characters).*

int **GetDescent** () *Gets font descent (distance from base line to bottom of characters).*

int **GetHeight** () *Gets total height of font (= is leading + ascent + descent)*

int **GetMaxAscent** () *Gets the maximum ascent of all characters in this Font.*

int **GetMaxDescent** () *Gets the maximum descent of all characters.*

int **GetMaxAdvance** () *Gets maximum advance width of any character (-1 if unknown).*

int **StringWidth** (const string *str) *Returns width of specified String in Font.*

int **StringWidth** (const char *str) *see above.*

void **Box** (const string *s, int &wd, int &ht) *calculates height and width of multiline string in grafport.*

void **Box** (const char *s, int &wd, int &ht) *see above.*

long **GetXFid** () *Return x font id of stored X font.*

void **GetXLFD** (char *xd) *Get XLFD description.*

void **GetXLFD** (string *xd) *Get XLFD description.*

void **GetPostScript** (char *ps) *Get PostScript name of this Font.*

XFontStruct* **GetXFontStruct** ()

Protected Members

void **CalcXLFD** (string *fn, bool scalable)

Calculate X logical font description.

static bool
IsScalableFont (const string *name)

static XFontStruct*
LoadQueryScalableFont (Display *d,
const char *name,
int size)

Private Members

XFontStruct*
xFont *X font struct corresponding to this font.*

Display* **display**

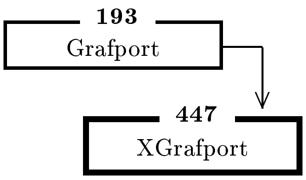
string **fontName** *the full XLFD fontname.*

447

class **XGrafport** : public Grafport

X11 grafport for Xlib drawing.

Inheritance



Public Members

	XGrafport (Display *d, Window win, int wd, int ht)
virtual	**~XGrafport** ()
bool	**DrawsXor** ()
void	**ClearArea** (int x, int y, int wd, int ht)
int	**GetWidth** ()
int	**GetHeight** ()
void	**Redraw** (int fromX, int fromY, int wd, int ht, int toX, int toY) *redraw using rdPixmap.*
void	**UpdateSize** (int wd, int ht)
void	**SetBackgroundColor** (const string *colorName)
void	**SetBackgroundColor** (const char *colorName)
void	**SetForegroundColor** (const string *colorName)
void	**SetForegroundColor** (const char *colorName)
void	**SetFont** (XFont *font)
void	**DrawRectangle** (int x, int y, int wd, int ht)
void	**FillRectangle** (int x, int y, int wd, int ht)
void	**DrawRoundedRectangle** (int x, int y, int wd, int ht, int rd)

void **FillRoundedRectangle** (int x, int y, int wd, int ht, int rd)

void **DrawEllipsedRectangle** (int x, int y, int wd, int ht, int rd)

void **FillEllipsedRectangle** (int x, int y, int wd, int ht, int rd)

void **DrawDisk** (int x, int y, int wd, int ht, int ellipse_ht)

void **FillDisk** (int x, int y, int wd, int ht, int ellipse_ht)

void **DrawEditDashedRectangle** (int x, int y, int wt, int ht)
not zoomed, not printed.

void **DrawUnzoomedRectangle** (int x, int y, int wt, int ht)

void **DrawStringLeft** (int x, int y, const char *str)

void **DrawStringRight** (int x, int y, const char *str)

void **DrawStringCentered** (int x, int y, const char *str)

void **DrawStringLeftUnderlined** (int x, int y, const char *str)

void **DrawStringRightUnderlined** (int x, int y, const char *str)

void **DrawStringCenteredUnderlined** (int x, int y, const char *str)

void **DrawUnzoomedStringLeft** (int, int, const char *)
not zoomed

void **DrawUnzoomedStringCentered** (int, int, const char *)

void **DrawLine** (int x1, int y1, int x2, int y2)

void **DrawLine** (const Point *p1, const Point *p2) void **DrawCurve** (const Point *p1, const Point *p2, const Point *p3, const Point *p4)

void **DrawEditDottedGridLine** (int x1, int y1, int x2, int y2)
not zoomed. Uses different dot pattern.

void **DrawEditDottedLine** (int x1, int y1, int x2, int y2)

void **DrawEditLine** (int x1, int y1, int x2, int y2)
not zoomed and not printed

void **DrawEditPoint** (int x1, int y2)
not zoomed and not printed

void **DrawEditLine** (const Point *p1, const Point *p2)
not zoomed and not printed

void **DrawEditPoint** (const Point *p)
not zoomed and not printed

void **DrawUnzoomedLine** (int, int, int, int)

void **DrawPoint** (int x, int y)

void **DrawPoint** (const Point *p)

void **DrawEllipse** (int x, int y, int wd, int ht)

void **FillEllipse** (int x, int y, int wd, int ht)

void **DrawArc** (int x, int y, int wd, int ht, int arc1, int arc2)

void **FillArc** (int x, int y, int wd, int ht, int arc1, int arc2)

void **DrawPolygon** (const Point *points, int n)

void **FillPolygon** (const Point *points, int n)

void **DrawActiveCaret** (int x, int y)

void **DrawDeactiveCaret** (int x, int y)

Protected Members

void **DrawXRectangle** (GC gc, int x, int y, int wd, int ht)

void	**FillXRectangle** (GC gc, int x, int y, int wd, int ht)
void	**DrawXRoundedRectangle** (GC gc, int x, int y, int wd, int ht, int radius)
void	**FillXRoundedRectangle** (GC gc, int x, int y, int wd, int ht, int radius)
void	**DrawXEllipsedRectangle** (GC gc, int x, int y, int wd, int ht, int radius)
void	**FillXEllipsedRectangle** (GC gc, int x, int y, int wd, int ht, int radius)
void	**DrawXDisk** (GC gc, int x, int y, int wd, int ht, int ellipse_ht)
void	**FillXDisk** (GC gc, int x, int y, int wd, int ht, int ellipse_ht)
void	**DrawXLine** (GC gc, int x1, int y1, int x2, int y2)
void	**DrawXArc** (GC gc, int x, int y, int wd, int ht, int arc1, int arc2)
void	**FillXArc** (GC gc, int x, int y, int wd, int ht, int arc1, int arc2)
void	**DrawXPolygon** (GC gc, const Point *points, int n)
void	**FillXPolygon** (GC gc, const Point *points, int n)
GC	**CreateXorGC** () *X graphics context for xor drawing.*
GC	**CreateCopyGC** ()

		X graphics context for copy drawing.
GC	**CreateDotGC** ()	*X gc for drawing xor dotted patterns.*
GC	**CreateWideDotGC** ()	*X gc for drawing xor wide dotted patterns.*
GC	**CreateDashGC** ()	*X gc for drawing xor dashed patterns.*
GC	**GetCopyGC** ()	
GC	**GetXORGC** ()	
GC	**GetDotGC** ()	
GC	**GetWideDotGC** ()	
GC	**GetDashGC** ()	
void	**ColorInfo** ()	
void	**SetDots** (GC gc)	
Pixel	**GetWhitePixel** ()	
Pixel	**GetBlackPixel** ()	
Pixel	**GetForegroundPixel** ()	
int	**Px** (double t)	
int	**Py** (double t)	
void	**CalcCurvePoints** (const Point *p1, const Point *p2, const Point *p3, const Point *p4)	

Private Members

Window	**window**	*The X window where the drawing area is.*
Display*	**display**	*The X display.*
int	**width**	*The width of the drawing area.*
int	**height**	*The height of the drawing area.*

Pixmap	**rdPixmap**	*pixmap for redrawing on exposure*
GC	**copyGC**	*copy GC (Graphic Context). (see Xlib reference man.)*
GC	**xorGC**	*solid xor GC.*
GC	**dotGC**	*dotted xor GC*
GC	**wideDotGC**	*wide dotted xor GC*
GC	**dashGC**	*dashed xor GC*
enum		*number of points for approximating a curve.*
Colormap	**colormap**	*X colormap.*
enum		
char	**grayPattern** [PATTERNLEN]	*dot pattern (for EditText)*
bool	**whitePixelWarned**	
bool	**blackPixelWarned**	
int	**px0**	
int	**py0**	
int	**xx** [CURVE_POINTS]	
int	**yy** [CURVE_POINTS]	

448

class **XResources**

Fallback X resources.

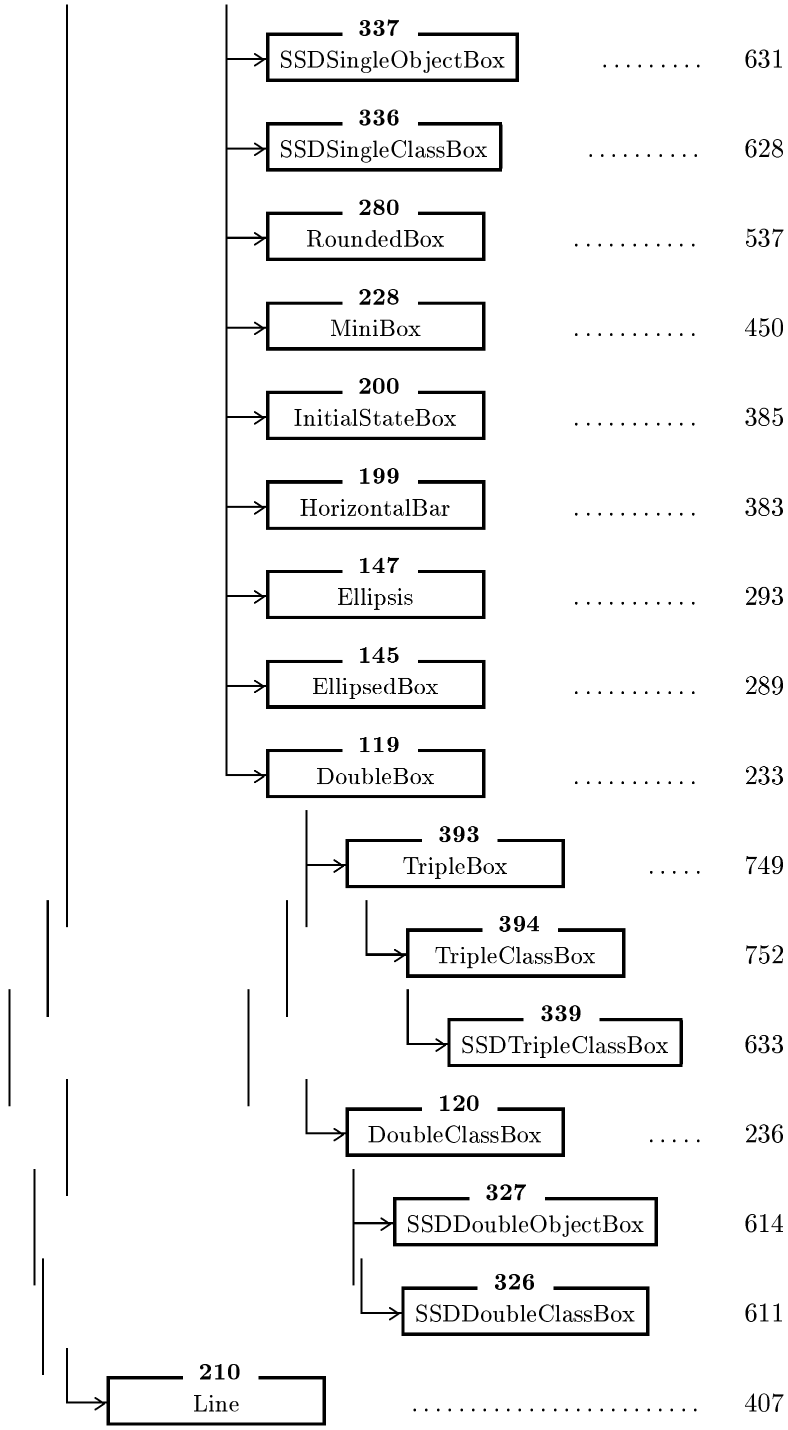
Public Members

static const char*
classicColorResources []

static const char*
modernColorResources []

static const char*
monoResources []

Class Graph


337 SSDSingleObjectBox 631
336 SSDSingleClassBox 628
280 RoundedBox 537
228 MiniBox 450
200 InitialStateBox 385
199 HorizontalBar 383
147 Ellipsis 293
145 EllipsedBox 289
119 DoubleBox 233
393 TripleBox 749
394 TripleClassBox 752
339 SSDTripleClassBox 633
120 DoubleClassBox 236
327 SSDDoubleObjectBox 614
326 SSDDoubleClassBox 611
210 Line 407