

XY Plot

4.1.0.1

Создано системой Doxygen 1.8.6

Чт 10 Июл 2014 11:45:42

Оглавление

1	Что нового?	1
1.1	Список изменений по версиям.	1
1.1.1	XY Plot v 4.4.1	1
1.1.1.1	Изменено	1
1.1.1.2	Добавлено	1
1.1.1.3	Исправлено	1
1.1.2	XY Plot v 3.4.1	1
1.1.2.1	Изменено	1
1.1.2.2	Добавлено	2
1.1.2.3	Исправлено	2
1.1.3	XY Plot v 2.3.1	2
1.1.3.1	Изменено	2
1.1.3.2	Добавлено	2
1.1.3.3	Исправлено	2
1.1.4	XY Plot v 2.2.1	2
1.1.4.1	Изменено	2
1.1.4.2	Добавлено	3
1.1.4.3	Исправлено	3
1.1.5	XY Plot v 2.1.1	3
1.1.5.1	Изменено	3
1.1.5.2	Добавлено	3
1.1.5.3	Исправлено	3
1.1.6	XY Plot v 1.0.1	3
1.1.6.1	Изменено	3
1.1.6.2	Добавлено	4
1.1.6.3	Исправлено	4
2	Страница 1	5
2.1	Это пример секции.	5
2.1.1	Раздел 1	5
2.1.2	Раздел 2	5

3	Страница 3	7
3.1	Это пример секции.	7
3.1.1	Раздел 2	7
3.1.2	Раздел 2	7
4	Алфавитный указатель групп	9
4.1	Группы	9
5	Алфавитный указатель пространств имен	11
5.1	Пространства имен	11
6	Иерархический список классов	13
6.1	Иерархия классов	13
7	Алфавитный указатель классов	15
7.1	Классы	15
8	Список файлов	17
8.1	Файлы	17
9	Группы	19
9.1	Функции инициализации и освобождения 2D графики	19
9.1.1	Подробное описание	19
9.1.2	Функции	19
9.1.2.1	CreatePlot	19
9.1.2.2	DestroyPlot	20
9.1.2.3	Initialize	20
9.2	Функции сохранения и загрузки свойств	21
9.2.1	Подробное описание	21
9.2.2	Функции	21
9.2.2.1	LoadProperties	21
9.2.2.2	RegLoadSettings	21
9.2.2.3	RegRestoreDefaults	21
9.2.2.4	RegSaveSettings	22
9.2.2.5	SaveProperties	22
9.3	Функции для работы с выделением	23
9.3.1	Подробное описание	23
9.3.2	Функции	23
9.3.2.1	GetSelectionMode	23
9.3.2.2	Select	23
9.3.2.3	Select	23
9.3.2.4	SetFlashSpeed	24

9.3.2.5	SetSelectionMode	24
9.4	Функции для работы с элементами графика	25
9.4.1	Подробное описание	25
9.4.2	Функции	25
9.4.2.1	EnableItem	25
9.4.2.2	GetAxisRange	25
9.4.2.3	GetColor	26
9.4.2.4	GetDouble	26
9.4.2.5	GetLong	26
9.4.2.6	GetString	27
9.4.2.7	IsEnabled	28
9.4.2.8	SetAxisRange	28
9.4.2.9	SetColor	28
9.4.2.10	SetDouble	29
9.4.2.11	SetFont	29
9.4.2.12	SetLong	29
9.4.2.13	SetString	30
9.5	Функции для работы с профилями и группами.	32
9.5.1	Подробное описание	32
9.5.2	Функции	32
9.5.2.1	AppendData	32
9.5.2.2	BindProfileToGroup	32
9.5.2.3	CreateGroup	32
9.5.2.4	CreateProfile	33
9.5.2.5	DeleteGroup	34
9.5.2.6	DeleteProfile	34
9.5.2.7	GetData	34
9.5.2.8	GetProfileList	35
9.5.2.9	SetData	36
10	Пространства имен	37
10.1	Пространство имен pyplot	37
10.1.1	Типы	40
10.1.1.1	FLASH_SPEED	40
10.1.1.2	LPFNADDLEVEL	40
10.1.1.3	LPFNADDPROFILE	40
10.1.1.4	LPFNADDREGION	40
10.1.1.5	LPFNAPPENDDATA	40
10.1.1.6	LPFNBINDPROFILE	40
10.1.1.7	LPFNCOPYTOCLIPBOARD	40

10.1.1.8	LPFNCREATEGROUP	40
10.1.1.9	LPFNDELETEGROUP	40
10.1.1.10	LPFNDELETELEVEL	40
10.1.1.11	LPFNDELETEPROFILE	40
10.1.1.12	LPFNDELETEREGION	40
10.1.1.13	LPFNDRAWTOBITMAP	40
10.1.1.14	LPFNENABLEITEM	41
10.1.1.15	LPFNFINALIZE	41
10.1.1.16	LPFNGETCOLOR	41
10.1.1.17	LPFNGETDOUBLE	41
10.1.1.18	LPFNGETLONG	41
10.1.1.19	LPFNGETPROFLIST	41
10.1.1.20	LPFNGETSELECTION	41
10.1.1.21	LPFNGETSTRING	41
10.1.1.22	LPFNINITIALIZE	41
10.1.1.23	LPFNISENABLED	41
10.1.1.24	LPFNLOADPROPERTIES	41
10.1.1.25	LPFNREGLOADSETTINGS	41
10.1.1.26	LPFNREGSTOREDEFAULTS	41
10.1.1.27	LPFNREGSAVESETTINGS	41
10.1.1.28	LPFNRUNDIALOG	41
10.1.1.29	LPFNSAVEPROPERTIES	41
10.1.1.30	LPFNSELECT	41
10.1.1.31	LPFNSETCOLOR	41
10.1.1.32	LPFNSETDOUBLE	41
10.1.1.33	LPFNSETFONT	41
10.1.1.34	LPFNSETLEVEL	41
10.1.1.35	LPFNSETLONG	42
10.1.1.36	LPFNSETREGION	42
10.1.1.37	LPFNSETSTRING	42
10.1.1.38	lProfileID	42
10.1.1.39	px	42
10.1.1.40	py	42
10.1.1.41	size	42
10.1.2	Перечисления	42
10.1.2.1	eAXISITEMS	42
10.1.2.2	eBACKGROUNDITEMS	43
10.1.2.3	eFLASHSPEED	43
10.1.2.4	eLEGENDITEMS	43
10.1.2.5	eLEVELLABELPOS	43

10.1.2.6	eMARKSTYPE	44
10.1.2.7	ePLOTELEMENT	44
10.1.2.8	ePLOTERRORS	45
10.1.2.9	ePROFILEITEMS	45
10.1.2.10	eSELECTIONITEMS	45
10.1.2.11	eSELECTIONTYPE	46
10.1.2.12	LINETYPE	46
10.1.3	Функции	46
10.1.3.1	long	46
11	Классы	47
11.1	Класс <code>хуplot::ALL_PLOTS_</code>	47
11.1.1	Подробное описание	48
11.1.2	Конструктор(ы)	48
11.1.2.1	ALL_PLOTS_	48
11.2	Класс <code>хуplot::ALL_PROFILES</code>	48
11.2.1	Конструктор(ы)	49
11.2.1.1	ALL_PROFILES	49
11.3	Класс <code>хуplot::ALL_REGIONS_</code>	50
11.3.1	Конструктор(ы)	50
11.3.1.1	ALL_REGIONS_	51
11.4	Класс <code>хуplot::HGROUP</code>	51
11.4.1	Конструктор(ы)	52
11.4.1.1	HGROUP	52
11.4.1.2	HGROUP	52
11.4.2	Документация по друзьям класса и функциям, относящимся к классу	52
11.4.2.1	<code>XYPlotManager::CreateGroup</code>	52
11.5	Класс <code>хуplot::HLEVEL</code>	52
11.5.1	Конструктор(ы)	53
11.5.1.1	HLEVEL	53
11.5.1.2	HLEVEL	53
11.5.2	Документация по друзьям класса и функциям, относящимся к классу	53
11.5.2.1	<code>XYPlotManager::CreateLevel</code>	53
11.6	Класс <code>хуplot::HPLOT</code>	53
11.6.1	Подробное описание	54
11.6.2	Конструктор(ы)	54
11.6.2.1	HPLOT	54
11.6.2.2	HPLOT	54
11.6.3	Документация по друзьям класса и функциям, относящимся к классу	54
11.6.3.1	<code>XYPlotManager::CreatePlot</code>	54

11.7 Класс <code>xyplot::HPLOTREGION</code>	55
11.7.1 Конструктор(ы)	56
11.7.1.1 <code>HPLOTREGION</code>	56
11.7.1.2 <code>HPLOTREGION</code>	56
11.7.2 Документация по друзьям класса и функциям, относящимся к классу	56
11.7.2.1 <code>XYPlotManager::CreateRegion</code>	56
11.8 Класс <code>xyplot::HPROFILE</code>	56
11.8.1 Подробное описание	58
11.8.2 Конструктор(ы)	58
11.8.2.1 <code>HPROFILE</code>	58
11.8.2.2 <code>HPROFILE</code>	58
11.8.3 Документация по друзьям класса и функциям, относящимся к классу	58
11.8.3.1 <code>XYPlotManager::CreateProfile</code>	58
11.8.3.2 <code>XYPlotManager::GetProfileList</code>	59
11.9 Класс <code>xyplot::INVALID_GROUP</code>	59
11.9.1 Конструктор(ы)	60
11.9.1.1 <code>INVALID_GROUP</code>	60
11.10 Класс <code>xyplot::INVALID_PROFILE</code>	60
11.10.1 Конструктор(ы)	61
11.10.1.1 <code>INVALID_PROFILE</code>	61
11.11 Класс <code>xyplot::PlotHandle</code>	62
11.11.1 Подробное описание	62
11.11.2 Конструктор(ы)	62
11.11.2.1 <code>~PlotHandle</code>	62
11.11.2.2 <code>PlotHandle</code>	62
11.11.2.3 <code>PlotHandle</code>	63
11.11.2.4 <code>PlotHandle</code>	64
11.11.3 Методы	64
11.11.3.1 <code>operator long</code>	64
11.11.3.2 <code>operator!</code>	64
11.11.3.3 <code>operator!=</code>	64
11.11.3.4 <code>operator&</code>	64
11.11.3.5 <code>operator=</code>	64
11.12 Класс <code>xyplot::XYPlotError</code>	65
11.12.1 Конструктор(ы)	65
11.12.1.1 <code>XYPlotError</code>	66
11.13 Класс <code>xyplot::XYPlotManager</code>	66
11.13.1 Подробное описание	68
11.13.2 Конструктор(ы)	68
11.13.2.1 <code>~XYPlotManager</code>	68

11.13.2.2 XYPlotManager	68
11.13.2.3 XYPlotManager	68
11.13.3 Методы	68
11.13.3.1 CopyToClipboard	68
11.13.3.2 CreateLevel	69
11.13.3.3 CreateRegion	69
11.13.3.4 DeleteLevel	69
11.13.3.5 DeleteRegion	69
11.13.3.6 DrawToBitmap	69
11.13.3.7 GetColor	69
11.13.3.8 GetLastError	69
11.13.3.9 GetSelectedProfile	69
11.13.3.10 Instance	69
11.13.3.11 RunDialog	70
11.13.3.12 SetColor	71
11.13.3.13 SetLevel	71
11.13.3.14 SetRedraw	71
11.13.3.15 SetRegion	71
11.13.3.16 TranslateError	71
11.13.4 Данные класса	71
11.13.4.1 m_hDllInstance	71
11.13.4.2 m_lpfnAddLevel	71
11.13.4.3 m_lpfnAddProfile	72
11.13.4.4 m_lpfnAddRegion	72
11.13.4.5 m_lpfnAppendData	72
11.13.4.6 m_lpfnBindProfile	72
11.13.4.7 m_lpfnCopyToClipboard	72
11.13.4.8 m_lpfnCreateGroup	72
11.13.4.9 m_lpfnDeleteGroup	72
11.13.4.10 m_lpfnDeleteLevel	72
11.13.4.11 m_lpfnDeleteProfile	72
11.13.4.12 m_lpfnDeleteRegion	72
11.13.4.13 m_lpfnDrawToBitmap	72
11.13.4.14 m_lpfnEnableItem	72
11.13.4.15 m_lpfnFinalize	72
11.13.4.16 m_lpfnGetColor	72
11.13.4.17 m_lpfnGetData	72
11.13.4.18 m_lpfnGetDouble	72
11.13.4.19 m_lpfnGetLong	72
11.13.4.20 m_lpfnGetProfileList	72

11.13.4.21	<code>lpfnGetSelection</code>	72
11.13.4.22	<code>lpfnGetString</code>	72
11.13.4.23	<code>lpfnInitialize</code>	72
11.13.4.24	<code>lpfnIsEnabled</code>	72
11.13.4.25	<code>lpfnLoadProperties</code>	72
11.13.4.26	<code>lpfnRegLoadSettings</code>	72
11.13.4.27	<code>lpfnRegRestoreDefaults</code>	72
11.13.4.28	<code>lpfnRegSaveSettings</code>	73
11.13.4.29	<code>lpfnRunDialog</code>	73
11.13.4.30	<code>lpfnSaveProperties</code>	73
11.13.4.31	<code>lpfnSelect</code>	73
11.13.4.32	<code>lpfnSetColor</code>	73
11.13.4.33	<code>lpfnSetData</code>	73
11.13.4.34	<code>lpfnSetDouble</code>	73
11.13.4.35	<code>lpfnSetFont</code>	73
11.13.4.36	<code>lpfnSetLevel</code>	73
11.13.4.37	<code>lpfnSetLong</code>	73
11.13.4.38	<code>lpfnSetRegion</code>	73
11.13.4.39	<code>lpfnSetString</code>	73
11.13.4.40	<code>nLastError</code>	73
11.14	Класс <code>xyplot::XYPlotRequestFailure</code>	73
11.14.1	Конструктор(ы)	74
11.14.1.1	<code>XYPlotRequestFailure</code>	74
12	Файлы	75
12.1	Файл <code>src/documentation_ru.h</code>	75
12.2	Файл <code>src/xyplotconst.h</code>	75
12.2.1	Подробное описание	77
12.2.2	Макросы	77
12.2.2.1	<code>_XY_PLOTCONST_H_</code>	77
12.3	Файл <code>src/XYPlotWrapper.cpp</code>	77
12.3.1	Макросы	79
12.3.1.1	<code>CHECK_FUNCTION_POINTER</code>	79
12.3.1.2	<code>XYFUNNAME_ADDLEVEL</code>	79
12.3.1.3	<code>XYFUNNAME_ADDPROFILE</code>	79
12.3.1.4	<code>XYFUNNAME_ADDREGION</code>	79
12.3.1.5	<code>XYFUNNAME_APPENDDATA</code>	79
12.3.1.6	<code>XYFUNNAME_BINDPROF</code>	79
12.3.1.7	<code>XYFUNNAME_CLIPCOPY</code>	79
12.3.1.8	<code>XYFUNNAME_CREATEGRP</code>	79

12.3.1.9 XYFUNNAME_DELETEGRP	79
12.3.1.10 XYFUNNAME_DELEVEL	79
12.3.1.11 XYFUNNAME_DELPROFILE	79
12.3.1.12 XYFUNNAME_DELREGION	79
12.3.1.13 XYFUNNAME_DRAWTOBITMAP	79
12.3.1.14 XYFUNNAME_ENABLEMARK	79
12.3.1.15 XYFUNNAME_ENITEM	79
12.3.1.16 XYFUNNAME_FINALIZE	79
12.3.1.17 XYFUNNAME_FLASHPROF	79
12.3.1.18 XYFUNNAME_GETAXRANGE	79
12.3.1.19 XYFUNNAME_GETCOLOR	79
12.3.1.20 XYFUNNAME_GETDATA	79
12.3.1.21 XYFUNNAME_GETDOUBLE	79
12.3.1.22 XYFUNNAME_GETLONG	79
12.3.1.23 XYFUNNAME_GETMARKPOS	79
12.3.1.24 XYFUNNAME_GETPROFCLR	79
12.3.1.25 XYFUNNAME_GETPROFLIST	79
12.3.1.26 XYFUNNAME_GETSELMODE	79
12.3.1.27 XYFUNNAME_GETSELPROF	79
12.3.1.28 XYFUNNAME_GETSTRING	79
12.3.1.29 XYFUNNAME_INITIALIZE	80
12.3.1.30 XYFUNNAME_IENABLED	80
12.3.1.31 XYFUNNAME_LOADPROP	80
12.3.1.32 XYFUNNAME_REGLOAD	80
12.3.1.33 XYFUNNAME_REGRESTORE	80
12.3.1.34 XYFUNNAME_REGSAVE	80
12.3.1.35 XYFUNNAME_RUNDIALOG	80
12.3.1.36 XYFUNNAME_SAVEPROP	80
12.3.1.37 XYFUNNAME_SETAXRANGE	80
12.3.1.38 XYFUNNAME_SETCOLOR	80
12.3.1.39 XYFUNNAME_SETDATA	80
12.3.1.40 XYFUNNAME_SETDOUBLE	80
12.3.1.41 XYFUNNAME_SETFLASH	80
12.3.1.42 XYFUNNAME_SETFONT	80
12.3.1.43 XYFUNNAME_SETLEVEL	80
12.3.1.44 XYFUNNAME_SETLONG	80
12.3.1.45 XYFUNNAME_SETMARKPOS	80
12.3.1.46 XYFUNNAME_SETPROFCLR	80
12.3.1.47 XYFUNNAME_SETPROFSTL	80
12.3.1.48 XYFUNNAME_SETREDRAW	80

12.3.1.49 XYFUNNAME_SETREGION	80
12.3.1.50 XYFUNNAME_SETSELMODE	80
12.3.1.51 XYFUNNAME_SETSTRING	80
12.4 Файл src/XYPlotWrapper.h	80
12.4.1 Макросы	83
12.4.1.1 ALL_PLOTS	83
12.4.1.2 ALL_REGIONS	83
12.4.1.3 XYPLOTFN	83

Глава 1

Что нового?

1.1 Список изменений по версиям.

1.1.1 XY Plot v 4.4.1

1.1.1.1 Изменено

- Изменение 001
- Изменение 002
- Изменение 003
- Изменение 004

1.1.1.2 Добавлено

- Добавление 001
- Добавление 002
- Добавление 003
- Добавление 004

1.1.1.3 Исправлено

- Исправление 001
- Исправление 002
- Исправление 003
- Исправление 004

1.1.2 XY Plot v 3.4.1

1.1.2.1 Изменено

- Изменение 001
- Изменение 002
- Изменение 003
- Изменение 004

1.1.2.2 Добавлено

- Добавление 001
- Добавление 002
- Добавление 003
- Добавление 004

1.1.2.3 Исправлено

- Исправление 001
- Исправление 002
- Исправление 003
- Исправление 004

1.1.3 XY Plot v 2.3.1

1.1.3.1 Изменено

- Изменение 001
- Изменение 002
- Изменение 003
- Изменение 004

1.1.3.2 Добавлено

- Добавление 001
- Добавление 002
- Добавление 003
- Добавление 004

1.1.3.3 Исправлено

- Исправление 001
- Исправление 002
- Исправление 003
- Исправление 004

1.1.4 XY Plot v 2.2.1

1.1.4.1 Изменено

- Изменение 001
- Изменение 002
- Изменение 003
- Изменение 004

1.1.4.2 Добавлено

- Добавление 001
- Добавление 002
- Добавление 003
- Добавление 004

1.1.4.3 Исправлено

- Исправление 001
- Исправление 002
- Исправление 003
- Исправление 004

1.1.5 XY Plot v 2.1.1

1.1.5.1 Изменено

- Изменение 001
- Изменение 002
- Изменение 003
- Изменение 004

1.1.5.2 Добавлено

- Добавление 001
- Добавление 002
- Добавление 003
- Добавление 004

1.1.5.3 Исправлено

- Исправление 001
- Исправление 002
- Исправление 003
- Исправление 004

1.1.6 XY Plot v 1.0.1

1.1.6.1 Изменено

- Изменение 001
- Изменение 002
- Изменение 003
- Изменение 004

1.1.6.2 Добавлено

- Добавление 001
- Добавление 002
- Добавление 003
- Добавление 004

1.1.6.3 Исправлено

- Исправление 001
- Исправление 002
- Исправление 003
- Исправление 004

Глава 2

Страница 1

Текст в начале страницы.

2.1 Это пример секции.

Эта страница содержит подсекции [Раздел 1](#) and [Раздел 2](#). Для дополнительной информации перейдите на [Страница 1](#).

2.1.1 Раздел 1

Это текст первой подсекции страницы 2.

2.1.2 Раздел 2

Это текст второй подсекции страницы 2. Это текст второй подсекции страницы 2.

Глава 3

Страница 3

Текст в начале страницы.

3.1 Это пример секции.

Эта страница содержит подсекции [Раздел 2](#) and [Раздел 2](#). Для дополнительной информации перейдите на страницу 1 [Что нового?](#).

3.1.1 Раздел 2

Это текст первой подсекции страницы 3.

3.1.2 Раздел 2

Это текст второй подсекции страницы 3. Это текст второй подсекции страницы 3.

Глава 4

Алфавитный указатель групп

4.1 Группы

Полный список групп.

Функции инициализации и освобождения 2D графики	19
Функции сохранения и загрузки свойств	21
Функции для работы с выделением	23
Функции для работы с элементами графика	25
Функции для работы с профилями и группами.	32

Глава 5

Алфавитный указатель пространств имен

5.1 Пространства имен

Полный список пространств имен.

[xuplot](#) 37

Глава 6

Иерархический список классов

6.1 Иерархия классов

Иерархия классов.

xyplot::PlotHandle	62
xyplot::HGROUP	51
xyplot::INVALID_GROUP	59
xyplot::HLEVEL	52
xyplot::HPLOT	53
xyplot::ALL_PLOTS_	47
xyplot::HPLOTREGION	55
xyplot::ALL_REGIONS_	50
xyplot::HPROFILE	56
xyplot::ALL_PROFILES	48
xyplot::INVALID_PROFILE	60
runtime_error	
xyplot::XYPlotError	65
xyplot::XYPlotRequestFailure	73
xyplot::XYPlotManager	66

Глава 7

Алфавитный указатель классов

7.1 Классы

Классы с их кратким описанием.

хуplot::ALL_PLOTS_	
Класс дескриптор для всех экземпляров 2D графики	47
хуplot::ALL_PROFILES	48
хуplot::ALL_REGIONS_	50
хуplot::HGROUP	51
хуplot::HLEVEL	52
хуplot::HPLOT	
Класс дескриптора 2D графики	53
хуplot::HPLOTREGION	55
хуplot::HPROFILE	
Класс дескриптора профиля	56
хуplot::INVALID_GROUP	59
хуplot::INVALID_PROFILE	60
хуplot::PlotHandle	
Базовый класс дескрипторов 2D графики	62
хуplot::XYPlotError	65
хуplot::XYPlotManager	66
хуplot::XYPlotRequestFailure	73

Глава 8

Список файлов

8.1 Файлы

Полный список файлов.

src/ documentation_ru.h	75
src/ xyplotconst.h	
Содержит константы и перечисляемые типы	75
src/ XYPlotWrapper.cpp	77
src/ XYPlotWrapper.h	80

Глава 9

Группы

9.1 Функции инициализации и освобождения 2D графики

Функции

- `BOOL xyplot::XYPlotManager::Initialize` (`const char *pszDllPath`, `HWND hwndParent=NULL`)
- `HPlot xyplot::XYPlotManager::CreatePlot` (`HWND hwndParent`)
- `BOOL xyplot::XYPlotManager::DestroyPlot` (`HPlot &hPlot`)

9.1.1 Подробное описание

Данный набор функций позволяет инициализировать класс-обертку для манипуляции всеми экземплярами 2D графики, а также создавать новые и уничтожать существующие экземпляры.

9.1.2 Функции

9.1.2.1 `HPlot XYPlotManager::CreatePlot (HWND hwndParent)`

Функция создания экземпляра 2D графики. В случае если не удалось создать экземпляр функция генерирует исключение `XYPlotRequestFailure`

Аргументы

<code>hwndParent</code>	- дескриптор окна в котором будет отображаться графика
-------------------------	--

Возвращает

Описатель экземпляра 2D графики

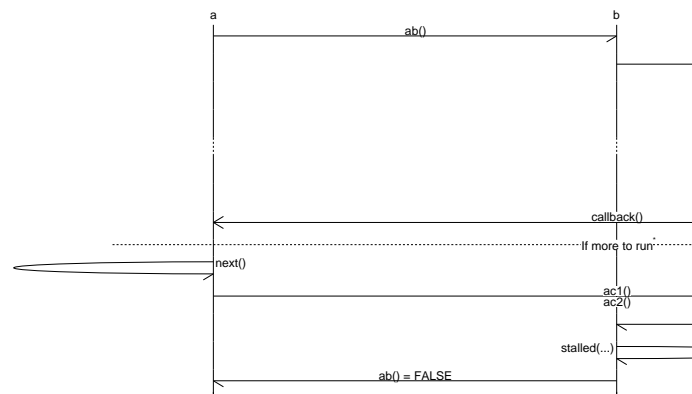
```
XYPlotManager& pm = XYPlotManager::Instance();
if ( !pm.Initialize("xyplot.dll", ::AfxGetMainWnd()->m_hWnd))
    return FALSE;
HPlot plot;
CChildView* pView = ...;//получить дескриптор окна в котором будет отображаться графика
try {
    plot = pm.CreatePlot(pView->m_hWnd)
}
catch (XYPlotRequestFailure& e)
{
    return FALSE;
}
```

9.1.2.2 BOOL XYPlotManager::DestroyPlot (HPLOT & hPlot)

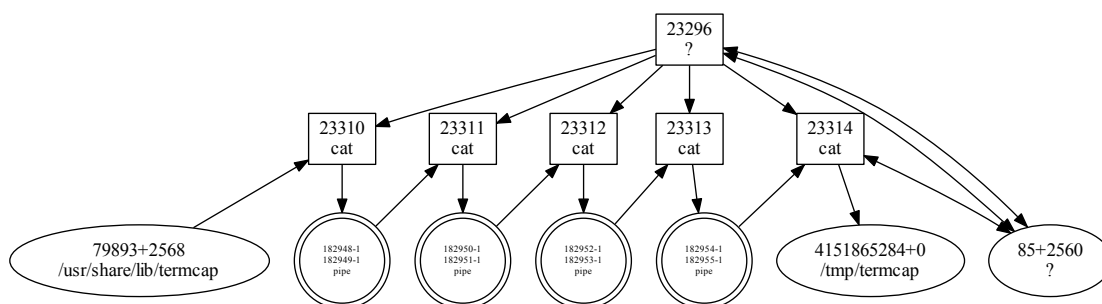
Функция удаления экземпляра 2D графики. Данную функцию необходимо вызвать перед уничтожением окна, указанного при создании экземпляра в функции CreatePlot

Возвращает

true - в случае если освобождение экземпляра графики прошло успешно;



false - в случае если не удалось освободить экземпляр.



9.1.2.3 BOOL XYPlotManager::Initialize (const char * pszDllPath, HWND hwndParent = NULL)

Данная функция инициализирует класса-обертку.

Аргументы

pszDllPath	- путь к динамически подключаемой библиотеке xyplot.dll
hwndParent	- опциональный параметр, дескриптор окна которому будут присылаться сообщения

Возвращает

true - в случае если инициализация прошла успешно;

false - в случае если инициализация не удалась.

9.2 Функции сохранения и загрузки свойств

Функции

- BOOL `xyplot::XYPlotManager::RegRestoreDefaults` (HPlot &hPlot)
- BOOL `xyplot::XYPlotManager::RegLoadSettings` (HPlot &hPlot, const char *pszKey, const char *pszValue)
- BOOL `xyplot::XYPlotManager::RegSaveSettings` (HPlot &hPlot, const char *pszKey, const char *pszValue)
- BOOL `xyplot::XYPlotManager::SaveProperties` (HPlot &hPlot, const char *pszFileName)
- BOOL `xyplot::XYPlotManager::LoadProperties` (HPlot &hPlot, const char *pszFileName)

9.2.1 Подробное описание

Данный набор функций позволяет сохранять и загружать свойства 2D графики из файла или реестра.

9.2.2 Функции

9.2.2.1 BOOL XYPlotManager::LoadProperties (HPlot & hPlot, const char * pszFileName)

Функция загружает настройки экземпляра 2D графики из файла.

Аргументы

<code>hPlot</code>	- дескриптор экземпляра 2D графики
<code>pszFileName</code>	- путь к файлу

Возвращает

`true` - в случае если удалось загрузить настройки;
`false` - в случае если не удалось загрузить настройки.

9.2.2.2 BOOL XYPlotManager::RegLoadSettings (HPlot & hPlot, const char * pszKey, const char * pszValue)

Функция загружает настройки экземпляра 2D графики из реестра.

Аргументы

<code>hPlot</code>	- дескриптор экземпляра 2D графики
<code>pszKey</code>	- ключ реестра
<code>pszValue</code>	- переменная ветки реестра

Возвращает

`true` - в случае если удалось загрузить настройки;
`false` - в случае если не удалось загрузить настройки.

9.2.2.3 BOOL XYPlotManager::RegRestoreDefaults (HPlot & hPlot)

Функция сбрасывает настройки экземпляра 2D графики в настройки по умолчанию.

Аргументы

hPlot	- дескриптор экземпляра 2D графики
-------	------------------------------------

Возвращает

true - в случае если удалось загрузить настройки;
false - в случае если не удалось загрузить настройки.

9.2.2.4 BOOL XYPlotManager::RegSaveSettings (HPLOT & hPlot, const char * pszKey, const char * pszValue)

Функция сохраняет настройки экземпляра 2D графики в реестре.

Аргументы

hPlot	- дескриптор экземпляра 2D графики
pszKey	- ключ реестра
pszValue	- переменная ветки реестра

Возвращает

true - в случае если удалось сохранить настройки;
false - в случае если не удалось сохранить настройки.

9.2.2.5 BOOL XYPlotManager::SaveProperties (HPLOT & hPlot, const char * pszFileName)

Функция сохраняет настройки экземпляра 2D графики в файле.

Аргументы

hPlot	- дескриптор экземпляра 2D графики
pszFileName	- путь к файлу

Возвращает

true - в случае если удалось сохранить настройки;
false - в случае если не удалось сохранить настройки.

9.3 Функции для работы с выделением

Функции

- **BOOL** `xyplot::XYPlotManager::Select` (**HPlot** &hPlot, **HPROFILE** &hProfile, unsigned nIndex)
- **BOOL** `xyplot::XYPlotManager::Select` (**HPlot** &hPlot, **HGROUP** &hGroup, unsigned nIndex)
- **BOOL** `xyplot::XYPlotManager::SetSelectionMode` (**HPlot** &hPlot, int mode)
- **long** `xyplot::XYPlotManager::GetSelectionMode` (**HPlot** &hPlot) const
- **BOOL** `xyplot::XYPlotManager::SetFlashSpeed` (**HPlot** &hPlot, int speed)

9.3.1 Подробное описание

Данный набор функций обеспечивает

9.3.2 Функции

9.3.2.1 `long XYPlotManager::GetSelectionMode (HPlot & hPlot) const`

Функция возвращает текущий режим выделения для профилей и групп профилей

Аргументы

<code>hPlot</code>	- дескриптор экземпляра 2D графики
--------------------	------------------------------------

Возвращает

`SM_NONE` = -1 - нет выделения
`SM_FLASH` = 0 - выделение мерцанием
`SM_VTRACE` = 1 - вертикальная трассировка
`SM_HTRACE` = 2 - горизонтальная трассировка
`SM_TRACE` = 3 - полная трассировка

9.3.2.2 `BOOL XYPlotManager::Select (HPlot & hPlot, HPROFILE & hProfile, unsigned nIndex)`

Функция устанавливает выделение указанного профиля.

Аргументы

<code>hPlot</code>	- дескриптор экземпляра 2D графики
<code>hProfile</code>	- дескриптор профиля
<code>nIndex</code>	- индекс в массиве данных

Возвращает

`true` - в случае если выделение профиля прошло успешно;
`false` - в случае если не удалось выделить профиль.

9.3.2.3 `BOOL XYPlotManager::Select (HPlot & hPlot, HGROUP & hGroup, unsigned nIndex)`

Функция устанавливает выделение указанной группы профилей.

Аргументы

hPlot	- дескриптор экземпляра 2D графики
hGroup	- дескриптор группы профилей
nIndex	- индекс в массиве данных

Возвращает

true - в случае если выделение группы профилей прошло успешно;
false - в случае если не удалось выделить группу профилей.

9.3.2.4 BOOL XYPlotManager::SetFlashSpeed (HPLOT & hPlot, int speed)

Функция задает частоту мерцания профиля либо группы профилей в режиме выделения SM_FLASH

Аргументы

hPlot	- дескриптор экземпляра 2D графики
speed	- режим выделения один из <ul style="list-style-type: none"> • FLS_NO_FLASH = 0 • FLS_VERY_SLOW = 1 • FLS_SLOW = 2 • FLS_NORMAL = 3 • FLS_FAST = 4 • FLS_VERY_FAST = 5 • FLS_EXTREME = 6

Возвращает

true - в случае если удалось установить частоту мерцания;
false - в случае если не удалось установить частоту мерцания.

9.3.2.5 BOOL XYPlotManager::SetSelectionMode (HPLOT & hPlot, int mode)

Функция задает режим выделения для профилей и групп профилей

Аргументы

hPlot	- дескриптор экземпляра 2D графики
mode	- режим выделения один из SM_NONE = -1 - нет выделения SM_FLASH = 0 - выделение мерцанием SM_VTRACE = 1 - вертикальная трассировка SM_HTRACE = 2 - горизонтальная трассировка SM_TRACE = 3 - полная трассировка

Возвращает

true - в случае если удалось установить режим выделения;
false - в случае если не удалось установить выделение.

9.4 Функции для работы с элементами графика

Функции

- **BOOL** `xyplot::XYPlotManager::SetAxisRange` (**HPLLOT** &hPlot, long axis, double dMin, double dMax)
- **BOOL** `xyplot::XYPlotManager::GetAxisRange` (**HPLLOT** &hPlot, long axis, double &dMin, double &dMax) const
- **std::string** `xyplot::XYPlotManager::GetString` (**HPLLOT** &hPlot, long item, long subitem) const
- **BOOL** `xyplot::XYPlotManager::SetString` (**HPLLOT** &hPlot, long item, long subitem, const char *pszString)
- **long** `xyplot::XYPlotManager::GetLong` (**HPLLOT** &hPlot, long item, long subitem) const
- **BOOL** `xyplot::XYPlotManager::SetLong` (**HPLLOT** &hPlot, long item, long subitem, long value)
- **double** `xyplot::XYPlotManager::GetDouble` (**HPLLOT** &hPlot, long item, long subitem) const
- **BOOL** `xyplot::XYPlotManager::SetDouble` (**HPLLOT** &hPlot, long item, long subitem, double value)
- **BOOL** `xyplot::XYPlotManager::EnableItem` (**HPLLOT** &hPlot, long item, long subitem, **BOOL** enabled)
- **BOOL** `xyplot::XYPlotManager::IsEnabled` (**HPLLOT** &hPlot, long item, long subitem) const
- **COLORREF** `xyplot::XYPlotManager::GetColor` (**HPLLOT** &hPlot, long item, long subitem) const
- **BOOL** `xyplot::XYPlotManager::SetColor` (**HPLLOT** &hPlot, long item, long subitem, **COLORREF** color)
- **BOOL** `xyplot::XYPlotManager::SetFont` (**HPLLOT** &hPlot, long item, long subitem, const **LOGFONT** *font)

9.4.1 Подробное описание

Данный набор функций обеспечивает работу с получением и заданием свойств элементов графика.

9.4.2 Функции

9.4.2.1 **BOOL** XYPlotManager::EnableItem (**HPLLOT** & hPlot, long item, long subitem, **BOOL** enabled)

Функция включения/отключения элемента 2D графики.

Аргументы

hPlot	- дескриптор экземпляра 2D графики
item	- идентификатор элемента 2D графики
subitem	- идентификатор подэлемента 2D графики
enabled	- флаг включения подэлемента 2D графики

Возвращает

true - в случае если удалось переключить состояние элемента;
false - в случае если не удалось установить значение с плавающей точкой элемента.

9.4.2.2 **BOOL** XYPlotManager::GetAxisRange (**HPLLOT** & hPlot, long axis, double & dMin, double & dMax) const

Функция получения диапазона значений оси экземпляра 2D графики.

Аргументы

	hPlot	- дескриптор экземпляра 2D графики
	axis	- идентификатор оси <ul style="list-style-type: none"> • PI_AXIS_LEFT = 6 • PI_AXIS_RIGHT = 7 • PI_AXIS_TOP = 8 • PI_AXIS_BOTTOM = 9
out	dMin	- начальное значение диапазона
out	dMax	- конечное значение диапазона

9.4.2.3 COLORREF XYPlotManager::GetColor (HPLOT & hPlot, long item, long subitem) const

Функция включения/отключения элемента 2D графики.

Аргументы

	hPlot	- дескриптор экземпляра 2D графики
	item	- идентификатор элемента 2D графики
	subitem	- идентификатор подэлемента 2D графики

Возвращает

текущий цвет элемента 2D графики.

9.4.2.4 double XYPlotManager::GetDouble (HPLOT & hPlot, long item, long subitem) const

Функция задания значения с плавающей точкой связанного с элементом 2D графики.

Аргументы

	hPlot	- дескриптор экземпляра 2D графики
	item	- идентификатор элемента 2D графики
	subitem	- идентификатор подэлемента 2D графики

Возвращает

значения с плавающей точкой связанное с подэлементом 2D графики.

9.4.2.5 long XYPlotManager::GetLong (HPLOT & hPlot, long item, long subitem) const

Функция получения целочисленного значения связанного с элементом 2D графики.

Аргументы

	hPlot	- дескриптор экземпляра 2D графики
	item	- идентификатор элемента 2D графики
	subitem	- идентификатор подэлемента 2D графики

Возвращает

целочисленные значения связанное с подэлементом 2D графики.

9.4.2.6 `std::string XYPlotManager::GetString (HPLOT & hPlot, long item, long subitem) const`

Функция получения строки элемента 2D графики.

Аргументы

hPlot	- дескриптор экземпляра 2D графики
item	- идентификатор элемента 2D графики
subitem	- идентификатор подэлемента 2D графики

Возвращает

строку связанную с подэлементом 2D графики.

9.4.2.7 `BOOL XYPlotManager::IsEnabled (HPLOT & hPlot, long item, long subitem) const`

Функция возвращает флаг состояния элемента 2D графики (включен/выключен).

Аргументы

hPlot	- дескриптор экземпляра 2D графики
item	- идентификатор элемента 2D графики
subitem	- идентификатор подэлемента 2D графики

Возвращает

`true` - в случае если элемент экземпляра графики включен;
`false` - в случае если элемент экземпляра графики выключен.

9.4.2.8 `BOOL XYPlotManager::SetAxisRange (HPLOT & hPlot, long axis, double dMin, double dMax)`

Функция задает диапазон значений оси экземпляра 2D графики. Данная функция работает только в случае если для оси не задан режим автоматического масштабирования.

Аргументы

hPlot	- дескриптор экземпляра 2D графики
axis	- идентификатор оси <ul style="list-style-type: none"> • <code>PI_AXIS_LEFT</code> = 6 • <code>PI_AXIS_RIGHT</code> = 7 • <code>PI_AXIS_TOP</code> = 8 • <code>PI_AXIS_BOTTOM</code> = 9
dMin	- начальное значение диапазона
dMax	- конечное значение диапазона

9.4.2.9 `BOOL XYPlotManager::SetColor (HPLOT & hPlot, long item, long subitem, COLORREF color)`

Функция задания цвета элемента 2D графики.

Аргументы

hPlot	- дескриптор экземпляра 2D графики
item	- идентификатор элемента 2D графики
subitem	- идентификатор подэлемента 2D графики
color	- новый цвет подэлемента 2D графики

Возвращает

true - в случае если удалось задать цвет элемента;
false - в случае если не удалось задать цвет элемента.

9.4.2.10 `BOOL XYPlotManager::SetDouble (HPLOT & hPlot, long item, long subitem, double value)`

Функция задания значения с плавающей точкой связанного с элементом 2D графики.

Аргументы

hPlot	- дескриптор экземпляра 2D графики
item	- идентификатор элемента 2D графики
subitem	- идентификатор подэлемента 2D графики
value	- с плавающей точкой значения связанное с подэлементом 2D графики.

Возвращает

true - в случае если удалось установить значение с плавающей точкой элемента;
false - в случае если не удалось установить значение с плавающей точкой элемента.

9.4.2.11 `BOOL XYPlotManager::SetFont (HPLOT & hPlot, long item, long subitem, const LOGFONT * font)`

Функция задания шрифта элемента 2D графики.

Аргументы

hPlot	- дескриптор экземпляра 2D графики
item	- идентификатор элемента 2D графики
subitem	- идентификатор подэлемента 2D графики
font	- указатель на новый шрифт подэлемента 2D графики

Возвращает

true - в случае если удалось задать шрифт элемента;
false - в случае если не удалось задать шрифт элемента.

9.4.2.12 `BOOL XYPlotManager::SetLong (HPLOT & hPlot, long item, long subitem, long value)`

Функция задания целочисленного значения связанного с элементом 2D графики.

Аргументы

hPlot	- дескриптор экземпляра 2D графики
item	- идентификатор элемента 2D графики
subitem	- идентификатор подэлемента 2D графики
value	- целочисленные значения связанное с подэлементом 2D графики.

Возвращает

true - в случае если удалось установить целочисленное значение элемента;
false - в случае если не удалось установить значение элемента.

9.4.2.13 `BOOL XYPlotManager::SetString (HPLOT & hPlot, long item, long subitem, const char * pszString)`

Функция задания строки элемента 2D графики.

Аргументы

hPlot	- дескриптор экземпляра 2D графики
item	- идентификатор элемента 2D графики
subitem	- идентификатор подэлемента 2D графики
pszString	- строка для подэлемента 2D графики.

Возвращает

true - в случае если удалось установить строку элемента;
false - в случае если не удалось установить строку элемента.

9.5 Функции для работы с профилями и группами.

Функции

- `HPROFILE xyplot::XYPlotManager::CreateProfile (HPLOT &hPlot, const char *name, COLORREF color, int width, int lineType=xyplot::PLS_SOLID, const char *szLineTemplate=0, BOOL visible=TRUE, BOOL showmarks=FALSE, const long xParentAxis=xyplot::PI_AXIS_BOTTOM, const long yParentAxis=xyplot::PI_AXIS_LEFT)`
- `BOOL xyplot::XYPlotManager::DeleteProfile (HPLOT &hPlot, HPROFILE &hProfile)`
- `HGROUP xyplot::XYPlotManager::CreateGroup (HPLOT &hPlot)`
- `BOOL xyplot::XYPlotManager::DeleteGroup (HPLOT &hPlot, HGROUP &hGroup)`
- `BOOL xyplot::XYPlotManager::BindProfileToGroup (HPLOT &hPlot, HGROUP &hGroup, HPROFILE &hProfile)`
- `BOOL xyplot::XYPlotManager::GetProfileList (HPLOT &hPlot, std::vector< HPROFILE > &hProfiles)`
- `BOOL xyplot::XYPlotManager::SetData (HPLOT &hPlot, HPROFILE &hProfile, const double *px, const double *py, unsigned size)`
- `BOOL xyplot::XYPlotManager::GetData (HPLOT &hPlot, HPROFILE &hProfile, double *px, double *py, unsigned *size)`
- `BOOL xyplot::XYPlotManager::AppendData (HPLOT &hPlot, HPROFILE &hProfile, const double *px, const double *py, unsigned size)`

9.5.1 Подробное описание

Данный набор функций обеспечивает создание, удаление профилей и групп профилей.

9.5.2 Функции

9.5.2.1 `BOOL XYPlotManager::AppendData (HPLOT & hPlot, HPROFILE & hProfile, const double * px, const double * py, unsigned size)`

9.5.2.2 `BOOL XYPlotManager::BindProfileToGroup (HPLOT & hPlot, HGROUP & hGroup, HPROFILE & hProfile)`

Добавление профиля к группе

Аргументы

<code>hPlot</code>	дескриптор экземпляра 2D графика содержащий группу профилей
<code>hProfile</code>	дескриптор профиля
<code>hGroup</code>	дескриптор группы

Возвращает

`true` если группа профилей удалена
`false` если группа профилей не удалена

9.5.2.3 `HGROUP XYPlotManager::CreateGroup (HPLOT & hPlot)`

Создание группы профилей

Аргументы

hPlot	дескриптор экземпляра 2D графика содержащий группу профилей
-------	---

Возвращает

дескриптор группы

См. также

[CreateProfile\(\)](#)
[BindProfileToGroup\(\)](#)

9.5.2.4 HPROFILE XYPlotManager::CreateProfile (HPLOT & hPlot, const char * name, COLORREF color, int width, int lineType = xyplot::PLS_SOLID, const char * szLineTemplate = 0, BOOL visible = TRUE, BOOL showmarks = FALSE, const long xParentAxis = xyplot::PI_AXIS_BOTTOM, const long yParentAxis = xyplot::PI_AXIS_LEFT)

Создания экземпляра профиля

Аргументы

hPlot	дескриптор экземпляра 2D графика для которого создается профиль
name	имя профиля (отображается в легенде)
color	цвет профиля
width	толщина линии профиля в пикселях
lineType	тип линии профиля <ul style="list-style-type: none"> • PLS_INVISIBLE - невидимая • PLS_SOLID - сплошная • PLS_DASH - штриховая • PLS_DOT - точечная • PLS_DASHDOT - осевая • PLS_DASHDOTDOT - штрих-точка-точка • PLS_CUSTOM - пользовательская

szLineTemplate	пользовательский шаблон линии. Данный аргумент используется только в случае если выбран пользовательский тип линии. Шаблон представляет собой последовательность целочисленных значений для задания интервалов заполнения и промежутков в пикселях. Например значение шаблона "5 2" означает, что 5px отрисовываются сплошной линией затем следует промежуток 2px.
visible	будет ли отображаться линия данного профиля
showmarks	отображать ли метки данных профиля
xParent Axis	ось X, к которой прикрепляется профиль
yParent Axis	ось Y, к которой прикрепляется профиль

Возвращает

дескриптор экземпляра профиля

```
XYPlotManager& pm = XYPlotManager::Instance();
if ( !pm.Initialize("xyplot.dll", ::Afx GetMainWnd()->m_hWnd))
    return FALSE;
HPLOT plot;
CChildView* pView = ...; //получить дескриптор окна в котором будет отображаться графика
try {
    plot = pm.CreatePlot(pView->m_hWnd)
}
catch (XYPlotRequestFailure& e)
{
    return FALSE;
}
```

9.5.2.5 BOOL XYPlotManager::DeleteGroup (HPLOT & hPlot, HGROUP & hGroup)

Удаление группы профилей

Аргументы

hPlot	дескриптор экземпляра 2D графика содержащий группу профилей
hGroup	дескриптор группы

Возвращает

true если группа профилей удалена
false если группа профилей не удалена

9.5.2.6 BOOL XYPlotManager::DeleteProfile (HPLOT & hPlot, HPROFILE & hProfile)

Удаление экземпляра профиля

Аргументы

hPlot	дескриптор экземпляра 2D графика содержащий профиль
hProfile	дескриптор профиля

Возвращает

true если профиль удален
false если профиль не удален.

9.5.2.7 BOOL XYPlotManager::GetData (HPLOT & hPlot, HPROFILE & hProfile, double * px, double * py, unsigned * size)

```
9.5.2.8  BOOL XYPlotManager::GetProfileList (  HPLOT & hPlot,  std::vector< HPROFILE > &
        hProfiles  )
```

Получение списка профилей экземпляра 2D графика

Аргументы

hPlot	дескриптор экземпляра 2D графика содержащий группу профилей
hProfiles	контейнер содержащий список всех профилей

Возвращает

true если удалось получить список профилей
false если не удалось получить список профилей

9.5.2.9 `BOOL XYPlotManager::SetData (HPLOT & hPlot, HPROFILE & hProfile, const double * px, const double * py, unsigned size)`

Глава 10

Пространства имен

10.1 Пространство имен хуplot

Классы

- class [XYPlotManager](#)
- class [PlotHandle](#)
Базовый класс дескрипторов 2D графики
- class [HPLOT](#)
Класс дескриптора 2D графики
- class [ALL_PLOTS_](#)
Класс дескриптор для всех экземпляров 2D графики
- class [HPROFILE](#)
Класс дескриптора профиля
- class [ALL_PROFILES](#)
- class [INVALID_PROFILE](#)
- class [HGROUP](#)
- class [INVALID_GROUP](#)
- class [HPLOTREGION](#)
- class [ALL_REGIONS_](#)
- class [HLEVEL](#)
- class [XYPlotError](#)
- class [XYPlotRequestFailure](#)

Определения типов

- typedef int [FLASH_SPEED](#)
- typedef long([XYPLOTFN](#) * [LPFNINITIALIZE](#))(HWND hwndMain, long *lPlotID)
- typedef long([XYPLOTFN](#) * [LPFNFINALIZE](#))(long lPlotID)
- typedef long([XYPLOTFN](#) * [LPFNADDPROFILE](#))(long lPlotID, const char *name, COLORREF color, int width, int linetype, const char *pszTemplate, BOOL bVisible, BOOL bDataMarks, long XAxis, long YAxis, long *lProfileID)
- typedef long([XYPLOTFN](#) * [LPFNDELETEPROFILE](#))(long lPlotID, long lProfileID)
- typedef long lProfileID
- typedef long const double * [px](#)
- typedef long const double
const double * [py](#)
- typedef long const double
const double unsigned [size](#)

- typedef long(XYPLOTFN * LPFNAPPENDDATA)(long lPlotID, long lProfileID, const double *px, const double *py, unsigned size)
- typedef long(XYPLOTFN * LPFNSETSTRING)(long lPlotID, long item, long subitem, const char *pszString)
- typedef long(XYPLOTFN * LPFNGETSTRING)(long lPlotID, long item, long subitem, long *buflen, char *pszBuffer)
- typedef long(XYPLOTFN * LPFNSETCOLOR)(long lPlotID, long item, long subitem, COLORREF color)
- typedef long(XYPLOTFN * LPFNGETCOLOR)(long lPlotID, long item, long subitem, COLORREF *pColor)
- typedef long(XYPLOTFN * LPFNSETLONG)(long lPlotID, long item, long subitem, long value)
- typedef long(XYPLOTFN * LPFNGETLONG)(long lPlotID, long item, long subitem, long *value)
- typedef long(XYPLOTFN * LPFNSETDOUBLE)(long lPlotID, long item, long subitem, double value)
- typedef long(XYPLOTFN * LPFNGETDOUBLE)(long lPlotID, long item, long subitem, double *value)
- typedef long(XYPLOTFN * LPFNSETFONT)(long lPlotID, long item, long subitem, const LOGFONT *plf)
- typedef long(XYPLOTFN * LPFNISENABLED)(long lPlotID, long item, long subitem, BOOL *enabled)
- typedef long(XYPLOTFN * LPFNENABLEITEM)(long lPlotID, long item, long subitem, BOOL enable)
- typedef long(XYPLOTFN * LPFNSELECT)(long lPlotID, long lProfileID, long index)
- typedef long(XYPLOTFN * LPFNGETSELECTION)(long lPlotID, long *pID)
- typedef long(XYPLOTFN * LPFNSAVEPROPERTIES)(long lPlotID, const char *pszFileName)
- typedef long(XYPLOTFN * LPFNLOADPROPERTIES)(long lPlotID, const char *pszFileName)
- typedef long(XYPLOTFN * LPFNCOPYTOCLIPBOARD)(long lPlotID, unsigned width, unsigned height, unsigned dpi)
- typedef long(XYPLOTFN * LPFNDRAWTOBITMAP)(long lPlotID, HBITMAP &hBitmap, bool bDrawWindowBG, bool bDrawClientBG)
- typedef long(XYPLOTFN * LPFNCREATEGROUP)(long lPlotID, long *plGroupID)
- typedef long(XYPLOTFN * LPFNDELETEGROUP)(long lPlotID, long lGroupID)
- typedef long(XYPLOTFN * LPFNBINDPROFILE)(long lPlotID, long lGroupID, long lProfileID)
- typedef long(XYPLOTFN * LPFNGETPROFLIST)(long lPlotID, long *lSize, long *lProfileIDs)
- typedef long(XYPLOTFN * LPFNREGRESTOREDEFAULTS)(long lPlotID)
- typedef long(XYPLOTFN * LPFNREGLOADSETTINGS)(long lPlotID, const char *pszKeyName, const char *pszValue)
- typedef long(XYPLOTFN * LPFNREGSAVESETTINGS)(long lPlotID, const char *pszKeyName, const char *pszValue)
- typedef long(XYPLOTFN * LPFNRUNDIALOG)(long lPlotID)
- typedef long(XYPLOTFN * LPFNADDREGION)(long lPlotID, long AxisX, long AxisY, double xFrom, double xTo, double yFrom, double yTo, const char *pszTitle, long *lRegionID)
- typedef long(XYPLOTFN * LPFNDELETEREGION)(long lPlotID, long lRegionID)
- typedef long(XYPLOTFN * LPFNSETREGION)(long lPlotID, long lRegionID, double x1, double x2, double y1, double y2)
- typedef long(XYPLOTFN * LPFNADDLEVEL)(long lPlotID, long axis, double value, const char *pszTitle, COLORREF clr, int width, long *lLevelID)
- typedef long(XYPLOTFN * LPFNDELETELEVEL)(long lPlotID, long lLevelID)
- typedef long(XYPLOTFN * LPFNSETLEVEL)(long lPlotID, long lLevelID, double value)

Перечисления

- enum `ePLOTERRORS` {
`PE_NOERROR`, `PE_UNKNOWN`, `PE_UNEXPECTED`, `PE_INVALID_PLOT`,
`PE_INVALID_PROFILE`, `PE_INVALID_WINDOW`, `PE_ALREADY_HOOKED`, `PE_INSUFFICIENT_MEMORY`,
`PE_INVALID_RANGE`, `PE_INVALID_ARG`, `PE_INSUFFICIENT_STORAGE`, `PE_OUT_OF_RANGE`,
`PE_INVALID_GROUP`, `PE_RTFTTEXT`, `PE_INVALID_DIALOG_DLL`, `PE_DIALOG_DLL_NOT_FOUND` }

Коды ошибок библиотеки

- enum `ePLETELEMENT` {
`PI_NOTHING`, `PI_BACKGROUND`, `PI_BACKGROUND_CLIENT`, `PI_BORDER`,
`PI_TITLE`, `PI_LEGEND`, `PI_AXIS_LEFT`, `PI_AXIS_RIGHT`,
`PI_AXIS_TOP`, `PI_AXIS_BOTTOM`, `PI_SELECTION`, `PI_CANVAS` }

Идентификаторы элементов 2D графика

- enum `eSELECTIONITEMS` { `SP_MODE`, `SP_FLASH_SPEED`, `SP_NOTIFY_MESSAGE` }

Подэлементы выделения

- enum `eSELECTIONTYPE` {
`SM_NONE` = -1, `SM_FLASH`, `SM_VTRACE`, `SM_HTRACE`,
`SM_TRACE` }

Тип выделения

- enum `eFLASHSPEED` {
`FLS_NO_FLASH`, `FLS_VERY_SLOW`, `FLS_SLOW`, `FLS_NORMAL`,
`FLS_FAST`, `FLS_VERY_FAST`, `FLS_EXTREME` }

Скорость мерцания профиля или группы профилей в режиме выделения `SM_FLASH`.

- enum `eAXISITEMS` {
`PAI_TITLE` = 200, `PAI_MAJOR_TICKS`, `PAI_MINOR_TICKS`, `PAI_MAJOR_TICKS_LENGTH`,
`PAI_MINOR_TICKS_LENGTH`, `PAI_MAJOR_TICKS_COUNT`, `PAI_MINOR_TICKS_COUNT`, `PAI_MAJOR_TICKS_WIDTH`,
`PAI_MINOR_TICKS_WIDTH`, `PAI_LINE`, `PAI_LINE_WIDTH`, `PAI_LINE_STYLE`,
`PAI_MAJOR_DIGITS`, `PAI_MINOR_DIGITS`, `PAI_MAJOR_DIGITS_FORMAT`, `PAI_MINOR_DIGITS_FORMAT`,
`PAI_MAJOR_GRID_LINES`, `PAI_MINOR_GRID_LINES`, `PAI_LOWER_LIMIT`, `PAI_UPPER_LIMIT`,
`PAI_AUTOSCALE`, `PAI_LOG10`, `PAI_STYLE`, `PAI_TIME_HISTORY` }

Элементы осей

- enum `eLEGENDITEMS` {
`PLP_ALIGN` = 220, `PLP_LINELENGTH` = 221, `PLP_LEFT` = -1, `PLP_RIGHT` = 1,
`PLP_BOTTOM` = 0 }

legend properties

- enum `eLEVELLABELPOS` {
`LLP_CENTER` = 0x001, `LLP_LEFT` = 0x002, `LLP_RIGHT` = 0x004, `LLP_ABOVE` = 0x008,
`LLP_UNDER` = 0x010, `LLP_ATLINE` = `LLP_ABOVE`|`LLP_UNDER`, `LLP_ABOVELEFT` =
`LLP_LEFT`|`LLP_ABOVE`, `LLP_ABOVERIGHT` = `LLP_RIGHT`|`LLP_ABOVE`,
`LLP_ABOVECENTER` = `LLP_CENTER`|`LLP_ABOVE`, `LLP_UNDERLEFT` = `LLP_LEFT`|
`LLP_UNDER`, `LLP_UNDERRIGHT` = `LLP_RIGHT`|`LLP_UNDER`, `LLP_UNDERCENTER` =
`LLP_CENTER`|`LLP_UNDER`,
`LLP_ATLINELEFT` = `LLP_LEFT`|`LLP_ATLINE`, `LLP_ATLINERIGHT` = `LLP_RIGHT`|
`LLP_ATLINE`, `LLP_ATLINECENTER` = `LLP_CENTER`|`LLP_ATLINE` }

Level label position - положение надписи уровня

- enum `LINETYPE` {
`PLS_INVISIBLE`, `PLS_SOLID`, `PLS_DASH`, `PLS_DOT`,
`PLS_DASHDOT`, `PLS_DASHDOTDOT`, `PLS_CUSTOM` }

Тип линии

- enum `ePROFILEITEMS` {
`PRP_COLOR = 1, PRP_WIDTH, PRP_STYLE, PRP_TEMPLATE,`
`PRP_NAME, PRP_MARKS, PRP_MARKSFREQ, PRP_MARKSTYPE,`
`PRP_MARKSIZE, PRP_MARKFILLCOLOR, PRP_MARKSTROKECOLOR` }
 Profile properties.
- enum `eMARKSTYPE` {
`PMT_CIRCLE, PMT_SQUARE, PMT_DIAMOND, PMT_TRIANGLEUP,`
`PMT_TRIANGLEDOWN, PMT_CROSS0, PMT_CROSS45` }
 Типы меток данных
- enum `eBACKGROUNDITEMS` {
`BGP_COLOR_START = 330, BGP_COLOR_END, BGP_DIRECTION, BGP_GRADIENT-`
`_VERT = 1,`
`BGP_GRADIENT_HORZ = 0` }
 Background properties.

Функции

- typedef `long` (`XYPLOTFN *LPFNSETDATA`)(`long lPlotID`

10.1.1.1 Типы

10.1.1.1 typedef int `xyplot::FLASH_SPEED`

10.1.1.2 typedef long(`XYPLOTFN * xyplot::LPFNADDLEVEL`)(`long lPlotID, long axis, double value,`
`const char *pszTitle, COLORREF clr, int width, long *lLevelID`)

10.1.1.3 typedef long(`XYPLOTFN * xyplot::LPFNADDPROFILE`)(`long lPlotID, const char *name,`
`COLORREF color, int width, int linetype, const char *pszTemplate, BOOL bVisible, BOOL`
`bDataMarks, long XAxis, long YAxis, long *lProfileID`)

10.1.1.4 typedef long(`XYPLOTFN * xyplot::LPFNADDREGION`)(`long lPlotID, long AxisX, long`
`AxisY, double xFrom, double xTo, double yFrom, double yTo, const char *pszTitle, long`
`*lRegionID`)

10.1.1.5 typedef long(`XYPLOTFN * xyplot::LPFNAPPENDDATA`)(`long lPlotID, long lProfileID,`
`const double *px, const double *py, unsigned size`)

10.1.1.6 typedef long(`XYPLOTFN * xyplot::LPFNBINDPROFILE`)(`long lPlotID, long lGroupID,`
`long lProfileID`)

10.1.1.7 typedef long(`XYPLOTFN * xyplot::LPFNCOPYTOCLIPBOARD`)(`long lPlotID, unsigned`
`width, unsigned height, unsigned dpi`)

10.1.1.8 typedef long(`XYPLOTFN * xyplot::LPFNCREATEGROUP`)(`long lPlotID, long *plGroupID`)

10.1.1.9 typedef long(`XYPLOTFN * xyplot::LPFNDELETETEGROUP`)(`long lPlotID, long lGroupID`)

10.1.1.10 typedef long(`XYPLOTFN * xyplot::LPFNDELETELEVEL`)(`long lPlotID, long lLevelID`)

10.1.1.11 typedef long(`XYPLOTFN * xyplot::LPFNDELETEPROFILE`)(`long lPlotID, long`
`lProfileID`)

10.1.1.12 typedef long(`XYPLOTFN * xyplot::LPFNDELETETEREGION`)(`long lPlotID, long lRegionID`)

10.1.1.13 typedef long(`XYPLOTFN * xyplot::LPFNDRAWTOBITMAP`)(`long lPlotID, HBITMAP`
`&hBitmap, bool bDrawWindowBG, bool bDrawClientBG`)

- 10.1.1.14 `typedef long(XYPLOTFN * xyplot::LPFNENABLEITEM)(long lPlotID, long item, long subitem, BOOL enable)`
- 10.1.1.15 `typedef long(XYPLOTFN * xyplot::LPFNFINALIZE)(long lPlotID)`
- 10.1.1.16 `typedef long(XYPLOTFN * xyplot::LPFNGETCOLOR)(long lPlotID, long item, long subitem, COLORREF *pColor)`
- 10.1.1.17 `typedef long(XYPLOTFN * xyplot::LPFNGETDOUBLE)(long lPlotID, long item, long subitem, double *value)`
- 10.1.1.18 `typedef long(XYPLOTFN * xyplot::LPFNGETLONG)(long lPlotID, long item, long subitem, long *value)`
- 10.1.1.19 `typedef long(XYPLOTFN * xyplot::LPFNGETPROFLIST)(long lPlotID, long *lSize, long *lProfileIDs)`
- 10.1.1.20 `typedef long(XYPLOTFN * xyplot::LPFNGETSELECTION)(long lPlotID, long *pID)`
- 10.1.1.21 `typedef long(XYPLOTFN * xyplot::LPFNGETSTRING)(long lPlotID, long item, long subitem, long *buflen, char *pszBuffer)`
- 10.1.1.22 `typedef long(XYPLOTFN * xyplot::LPFNINITIALIZE)(HWND hwndMain, long *lPlotID)`
- 10.1.1.23 `typedef long(XYPLOTFN * xyplot::LPFNISENABLED)(long lPlotID, long item, long subitem, BOOL *enabled)`
- 10.1.1.24 `typedef long(XYPLOTFN * xyplot::LPFNLOADPROPERTIES)(long lPlotID, const char *pszFileName)`
- 10.1.1.25 `typedef long(XYPLOTFN * xyplot::LPFNREGLOADSETTINGS)(long lPlotID, const char *pszKeyName, const char *pszValue)`
- 10.1.1.26 `typedef long(XYPLOTFN * xyplot::LPFNREGRESTOREDEFAULTS)(long lPlotID)`
- 10.1.1.27 `typedef long(XYPLOTFN * xyplot::LPFNREGSAVESETTINGS)(long lPlotID, const char *pszKeyName, const char *pszValue)`
- 10.1.1.28 `typedef long(XYPLOTFN * xyplot::LPFNRUNDIALOG)(long lPlotID)`
- 10.1.1.29 `typedef long(XYPLOTFN * xyplot::LPFNSAVEPROPERTIES)(long lPlotID, const char *pszFileName)`
- 10.1.1.30 `typedef long(XYPLOTFN * xyplot::LPFNSELECT)(long lPlotID, long lProfileID, long index)`
- 10.1.1.31 `typedef long(XYPLOTFN * xyplot::LPFNSETCOLOR)(long lPlotID, long item, long subitem, COLORREF color)`
- 10.1.1.32 `typedef long(XYPLOTFN * xyplot::LPFNSETDOUBLE)(long lPlotID, long item, long subitem, double value)`
- 10.1.1.33 `typedef long(XYPLOTFN * xyplot::LPFNSETFONT)(long lPlotID, long item, long subitem, const LOGFONT *plf)`
- 10.1.1.34 `typedef long(XYPLOTFN * xyplot::LPFNSETLEVEL)(long lPlotID, long lLevelID, double value)`

- 10.1.1.35 `typedef long(XYPLOTFN * xyplot::LPFNSETLONG)(long lPlotID, long item, long subitem, long value)`
- 10.1.1.36 `typedef long(XYPLOTFN * xyplot::LPFNSETREGION)(long lPlotID, long lRegionID, double x1, double x2, double y1, double y2)`
- 10.1.1.37 `typedef long(XYPLOTFN * xyplot::LPFNSETSTRING)(long lPlotID, long item, long subitem, const char *pszString)`
- 10.1.1.38 `typedef long xyplot::lProfileID`
- 10.1.1.39 `typedef long double * xyplot::px`
- 10.1.1.40 `typedef long double double * xyplot::py`
- 10.1.1.41 `typedef long double double unsigned * xyplot::size`

10.1.2 Перечисления

- 10.1.2.1 `enum xyplot::eAXISITEMS`

Элементы осей

Элементы перечислений

- `PAI_TITLE` Заголовок оси
- `PAI_MAJOR_TICKS` Основные метки оси
- `PAI_MINOR_TICKS` Вспомогательные метки оси
- `PAI_MAJOR_TICKS_LENGTH` Длина основных меток оси
- `PAI_MINOR_TICKS_LENGTH` Длина дополнительных меток оси
- `PAI_MAJOR_TICKS_COUNT` Количество основных меток оси
- `PAI_MINOR_TICKS_COUNT` Количество вспомогательных меток оси
- `PAI_MAJOR_TICKS_WIDTH` Толщина линии основных меток оси
- `PAI_MINOR_TICKS_WIDTH` Толщина линии вспомогательных меток оси
- `PAI_LINE` Тип линии оси
- `PAI_LINE_WIDTH` Толщина линии оси
- `PAI_LINE_STYLE` Стилль линии
- `PAI_MAJOR_DIGITS` Подпись к основным меткам оси
- `PAI_MINOR_DIGITS` Подпись к дополнительным меткам оси
- `PAI_MAJOR_DIGITS_FORMAT` Формат подписей к основным меткам оси
- `PAI_MINOR_DIGITS_FORMAT` Формат подписи к дополнительным меткам оси
- `PAI_MAJOR_GRID_LINES` Линии сетки к основным меткам оси
- `PAI_MINOR_GRID_LINES` Линии сетки к дополнительным меткам оси
- `PAI_LOWER_LIMIT` Нижняя граница оси
- `PAI_UPPER_LIMIT` Верхняя граница оси
- `PAI_AUTOSCALE` Режим автоматического масштабирования оси
- `PAI_LOG10` Режим логарифмического отображения оси
- `PAI_STYLE` Текущий стиль оси
- `PAI_TIME_HISTORY` Режим отображения временной шкалы

10.1.2.2 `enum xyplot::eBACKGROUNDITEMS`

Background properties.

Элементы перечислений

`BGP_COLOR_START` Начальный цвет заливки
`BGP_COLOR_END` Конечный цвет заливки
`BGP_DIRECTION` Направление заливки
`BGP_GRADIENT_VERT` Вертикальная заливка
`BGP_GRADIENT_HORZ` Горизонтальная заливка

10.1.2.3 `enum xyplot::eFLASHSPEED`

Скорость мерцания профиля или группы профилей в режиме выделения `SM_FLASH`.

Элементы перечислений

`FLS_NO_FLASH` Нет мерцания
`FLS_VERY_SLOW` Очень медленно (1200 ms - отображение профиля, 600 ms - профиль не отображается)
`FLS_SLOW` Медленно (600 ms - отображение профиля, 400 ms - профиль не отображается)
`FLS_NORMAL` Нормально (400 ms - отображение профиля, 300 ms - профиль не отображается)
`FLS_FAST` Быстро (300 ms - отображение профиля, 150 ms - профиль не отображается)
`FLS_VERY_FAST` Очень быстро (150 ms - отображение профиля, 75 ms - профиль не отображается)
`FLS_EXTREME` Экстремально (10 ms - отображение профиля, 10 ms - профиль не отображается)

10.1.2.4 `enum xyplot::eLEGENDITEMS`

legend properties

Элементы перечислений

`PLP_ALIGN` Положение легенды
`PLP_LINELENGTH` Длина линии в легенде
`PLP_LEFT` Положение легенды в левой части
`PLP_RIGHT` Положение легенды в правой части
`PLP_BOTTOM` Положение легенды в нижней части

10.1.2.5 `enum xyplot::eLEVELLABELPOS`

Level label position - положение надписи уровня

Элементы перечислений

`LLP_CENTER` Горизонтальное выравнивание по центру
`LLP_LEFT` Горизонтальное выравнивание по левому краю

LLP_RIGHT Горизонтальное выравнивание по правому краю
 LLP_ABOVE Вертикальное выравнивание над линией
 LLP_UNDER Вертикальное выравнивание под линией
 LLP_ATLINE Вертикальное выравнивание по линии
 LLP_ABOVELEFT Положение слева над линией
 LLP_ABOVERIGHT Положение справа над линией
 LLP_ABOVECENTER Положение по середине над линией
 LLP_UNDERLEFT Положение слева под линией
 LLP_UNDERRIGHT Положение справа под линией
 LLP_UNDERCENTER Положение по середине под линией
 LLP_ATLINELEFT Положение слева по линии
 LLP_ATLINERIGHT Положение справа по линии
 LLP_ATLINECENTER Положение по середине по линии

10.1.2.6 enum xyplot::eMARKSTYPE

Типы меток данных

Элементы перечислений

PMT_CIRCLE Круглая
 PMT_SQUARE Квадратная
 PMT_DIAMOND Ромб
 PMT_TRIANGLEUP Треугольник направленный вверх
 PMT_TRIANGLEDOWN Треугольник направленный вниз
 PMT_CROSS0 Крест
 PMT_CROSS45 Повернутый крест

10.1.2.7 enum xyplot::ePLOTELEMENT

Идентификаторы элементов 2D графика

Элементы перечислений

PI_NOTHING Пустой элемент
 PI_BACKGROUND Заливка фона ркна
 PI_BACKGROUND_CLIENT Заливка клиентской области графика
 PI_BORDER Граница окна
 PI_TITLE Заголовок графика
 PI_LEGEND Легенда графика
 PI_AXIS_LEFT Левая ось
 PI_AXIS_RIGHT Правая ось
 PI_AXIS_TOP Верхняя ось
 PI_AXIS_BOTTOM Нижняя ось
 PI_SELECTION Текущее выделение
 PI_CANVAS Холст

10.1.2.8 enum xyplot::ePLOTERRORS

Коды ошибок библиотеки

Элементы перечислений

PE_NOERROR
PE_UNKNOWN
PE_UNEXPECTED
PE_INVALID_PLOT
PE_INVALID_PROFILE
PE_INVALID_WINDOW
PE_ALREADY_HOOKED
PE_INSUFFICIENT_MEMORY
PE_INVALID_RANGE
PE_INVALID_ARG
PE_INSUFFICIENT_STORAGE
PE_OUT_OF_RANGE
PE_INVALID_GROUP
PE_RTFTEXT
PE_INVALID_DIALOG_DLL
PE_DIALOG_DLL_NOT_FOUND

10.1.2.9 enum xyplot::ePROFILEITEMS

Profile properties.

Элементы перечислений

PRP_COLOR Цвет линии
PRP_WIDTH Толщина линии
PRP_STYLE Тип линии
PRP_TEMPLATE Пользовательский шаблон линии
PRP_NAME Имя профиля
PRP_MARKS Метки данных профиля
PRP_MARKSFREQ Частота меток данных
PRP_MARKSTYPE Тип меток
PRP_MARKSIZE Размер меток
PRP_MARKFILLCOLOR Цвет заполнения меток
PRP_MARKSTROKECOLOR Цвет линий меток

10.1.2.10 enum xyplot::eSELECTIONITEMS

Подэлементы выделения

Элементы перечислений

SP_MODE Режим выделения
SP_FLASH_SPEED Скорость мерцания в режиме выделения SM_FLASH.
SP_NOTIFY_MESSAGE Сообщение о выделении элемента графика

10.1.2.11 enum xyplot::eSELECTIONTYPE

Тип выделения

Элементы перечислений

SM_NONE Нет выделения

SM_FLASH Выделение мерцанием

SM_VTRACE Вертикальная трассировка профилей и групп

SM_HTRACE Горизонтальная трассировка профилей и групп

SM_TRACE Полная трассировка профилей и групп

10.1.2.12 enum xyplot::LINETYPE

Тип линии

Элементы перечислений

PLS_INVISIBLE Невидимая

PLS_SOLID Сплошная

PLS_DASH Штриховая

PLS_DOT Точечная

PLS_DASHDOT Осевая

PLS_DASHDOTDOT Штрих-точка-точка

PLS_CUSTOM Пользовательская

10.1.3 Функции

10.1.3.1 typedef xyplot::long (XYPLOTFN * LPFNSETDATA)

Глава 11

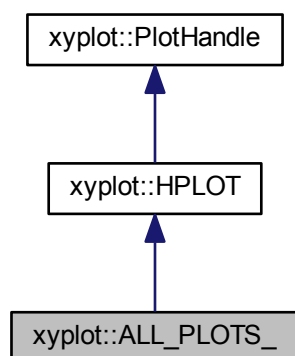
Классы

11.1 Класс `xyplot::ALL_PLOTS_`

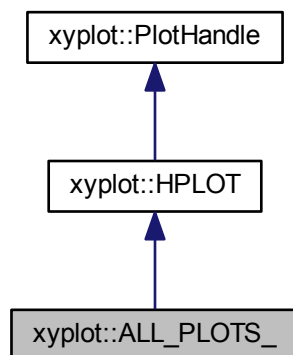
Класс дескриптор для всех экземпляров 2D графики

```
#include <XYPlotWrapper.h>
```

Граф наследования: `xyplot::ALL_PLOTS_`:



Граф связей класса `xyplot::ALL_PLOTS_`:



Открытые члены

- [ALL_PLOTS_\(\)](#)

Дополнительные унаследованные члены

11.1.1 Подробное описание

Класс дескриптор для всех экземпляров 2D графики

11.1.2 Конструктор(ы)

11.1.2.1 `xyplot::ALL_PLOTS_::ALL_PLOTS_ ()` [inline]

Конструктор по умолчанию.

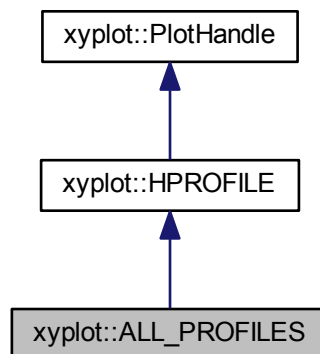
Объявления и описания членов класса находятся в файле:

- [src/XYPlotWrapper.h](#)

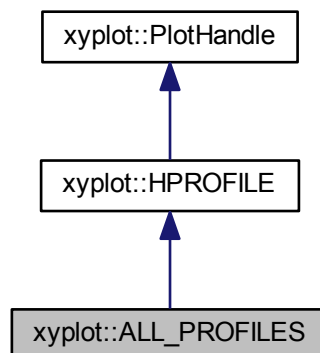
11.2 Класс `xyplot::ALL_PROFILES`

```
#include <XYPlotWrapper.h>
```

Граф наследования: `xyplot::ALL_PROFILES`:



Граф связей класса `xyplot::ALL_PROFILES`:



Открытые члены

- [ALL_PROFILES](#) ()

Дополнительные унаследованные члены

11.2.1 Конструктор(ы)

11.2.1.1 `xyplot::ALL_PROFILES::ALL_PROFILES ()` [inline]

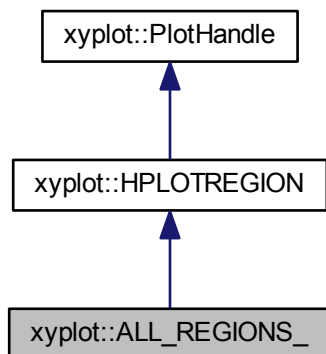
Объявления и описания членов класса находятся в файле:

- [src/XYPlotWrapper.h](#)

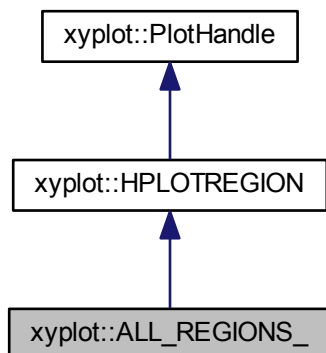
11.3 Класс `xyplot::ALL_REGIONS_`

```
#include <XYPlotWrapper.h>
```

Граф наследования: `xyplot::ALL_REGIONS_`:



Граф связей класса `xyplot::ALL_REGIONS_`:



Открытые члены

- [ALL_REGIONS_\(\)](#)

Дополнительные унаследованные члены

11.3.1 Конструктор(ы)

11.3.1.1 `xyplot::ALL_REGIONS_::ALL_REGIONS_ ()` [inline]

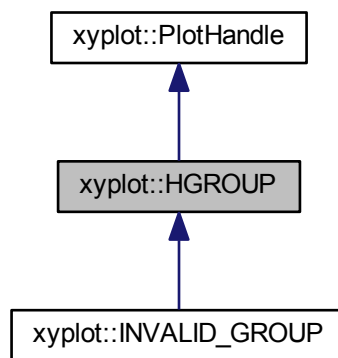
Объявления и описания членов класса находятся в файле:

- [src/XYPlotWrapper.h](#)

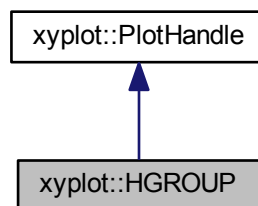
11.4 Класс `xyplot::HGROUP`

```
#include <XYPlotWrapper.h>
```

Граф наследования:`xyplot::HGROUP`:



Граф связей класса `xyplot::HGROUP`:



Открытые члены

- [HGROUP \(\)](#)

Защищенные члены

- [HGROUP \(long n\)](#)

Друзья

- [HGROUP XYPlotManager::CreateGroup \(HPLOT &\)](#)

11.4.1 Конструктор(ы)

11.4.1.1 `xyplot::HGROUP::HGROUP ()` [inline]

11.4.1.2 `xyplot::HGROUP::HGROUP (long n)` [inline], [explicit], [protected]

11.4.2 Документация по друзьям класса и функциям, относящимся к классу

11.4.2.1 `HGROUP XYPlotManager::CreateGroup (HPLOT &)` [friend]

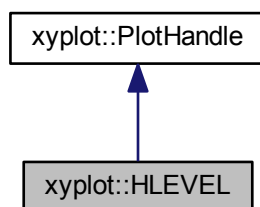
Объявления и описания членов класса находятся в файле:

- [src/XYPlotWrapper.h](#)

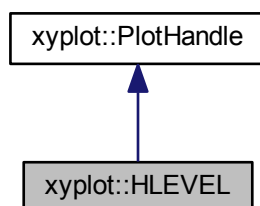
11.5 Класс `xyplot::HLEVEL`

```
#include <XYPlotWrapper.h>
```

Граф наследования:`xyplot::HLEVEL`:



Граф связей класса `xyplot::HLEVEL`:



Открытые члены

- [HLEVEL](#) ()

Защищенные члены

- [HLEVEL](#) (long n)

Друзья

- [HLEVEL XYPlotManager::CreateLevel](#) (HPLOT &, long, double, const char *, COLORREF, int)

11.5.1 Конструктор(ы)

11.5.1.1 `xyplot::HLEVEL::HLEVEL ()` [inline]

11.5.1.2 `xyplot::HLEVEL::HLEVEL (long n)` [inline], [explicit], [protected]

11.5.2 Документация по друзьям класса и функциям, относящимся к классу

11.5.2.1 `HLEVEL XYPlotManager::CreateLevel (HPLOT & , long , double , const char * , COLORREF , int)` [friend]

Объявления и описания членов класса находятся в файле:

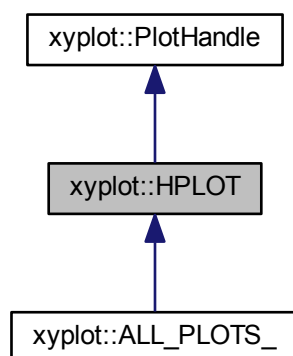
- `src/XYPlotWrapper.h`

11.6 Класс `xyplot::HPLOT`

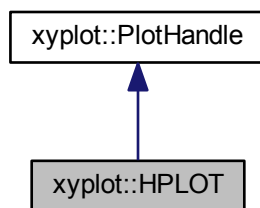
Класс дескриптора 2D графики

```
#include <XYPlotWrapper.h>
```

Граф наследования: `xyplot::HPLOT`:



Граф связей класса `xyplot::HPLOT`:



Открытые члены

- [HPLOT](#) ()

Защищенные члены

- [HPLOT](#) (long n)

Друзья

- [HPLOT](#) [XYPlotManager::CreatePlot](#) (HWND hwnd)

11.6.1 Подробное описание

Класс дескриптора 2D графики

11.6.2 Конструктор(ы)

11.6.2.1 `xyplot::HPLOT::HPLOT ()` [inline]

Конструктор по умолчанию. Инициализирует пустой дескриптор (= 0)

11.6.2.2 `xyplot::HPLOT::HPLOT (long n)` [inline], [explicit], [protected]

Конструктор с параметром.

Аргументы

n	- ???
---	-------

11.6.3 Документация по друзьям класса и функциям, относящимся к классу

11.6.3.1 `HPLOT` [XYPlotManager::CreatePlot](#) (HWND hwnd) [friend]

Дружественная функция из класса [XYPlotManager](#)

Аргументы

hwnd	дескриптор окна для которого создается экземпляр 2D графики
------	---

Возвращает

дескриптор экземпляра 2D графики

```

XYPlotManager& pm = XYPlotManager::Instance();
if ( !pm.Initialize("xyplot.dll", ::Afx GetMainWnd()->m_hWnd))
    return FALSE;
HPLOT plot;
CChildView* pView = ...; //получить дескриптор окна в котором будет отображаться графика
try {
    plot = pm.CreatePlot(pView->m_hWnd)
}
catch (XYPlotRequestFailure& e)
{
    return FALSE;
}

```

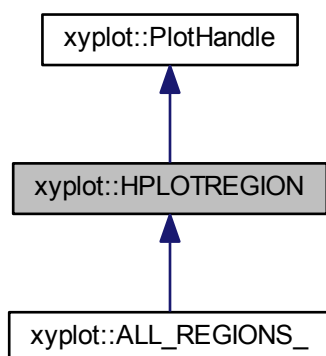
Объявления и описания членов класса находятся в файле:

- [src/XYPlotWrapper.h](#)

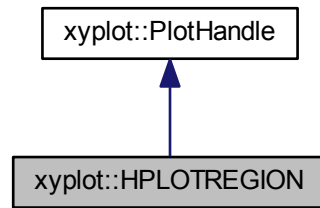
11.7 Класс xyplot::HPLOTREGION

```
#include <XYPlotWrapper.h>
```

Граф наследования:xyplot::HPLOTREGION:



Граф связей класса `xyplot::HPLOTREGION`:



Открытые члены

- [HPLOTREGION](#) ()

Защищенные члены

- [HPLOTREGION](#) (long n)

Друзья

- [HPLOTREGION XYPlotManager::CreateRegion](#) (HPLOT &, long, long, double, double, double, double, const char *)

11.7.1 Конструктор(ы)

11.7.1.1 `xyplot::HPLOTREGION::HPLOTREGION ()` [inline]

11.7.1.2 `xyplot::HPLOTREGION::HPLOTREGION (long n)` [inline], [explicit], [protected]

11.7.2 Документация по друзьям класса и функциям, относящимся к классу

11.7.2.1 `HPLOTREGION XYPlotManager::CreateRegion (HPLOT &, long, long, double, double, double, double, const char *)` [friend]

Объявления и описания членов класса находятся в файле:

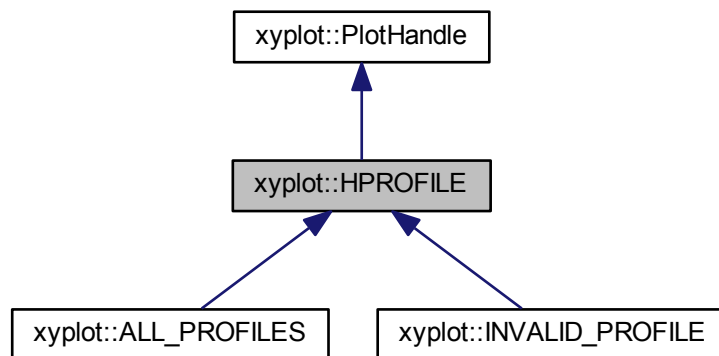
- [src/XYPlotWrapper.h](#)

11.8 Класс `xyplot::HPROFILE`

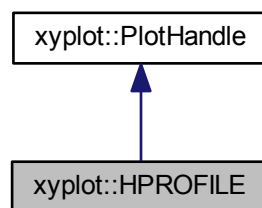
Класс дескриптора профиля

```
#include <XYPlotWrapper.h>
```

Граф наследования: `xyplot::HPROFILE`:



Граф связей класса `xyplot::HPROFILE`:



Открытые члены

- [HPROFILE](#) ()

Защищенные члены

- [HPROFILE](#) ([long](#) n)

Друзья

- [HPROFILE](#) [XYPlotManager::CreateProfile](#) ([HPLOT](#) &hPlot, const char *name, [COLORREF](#) color, int width, int lineType, const char *szLineTemplate, [BOOL](#) visible, [BOOL](#) showmarks, [long](#) xParentAxis, [long](#) yParentAxis)
- [BOOL](#) [XYPlotManager::GetProfileList](#) ([HPLOT](#) &, std::vector< [HPROFILE](#) > &)

11.8.1 Подробное описание

Класс дескриптора профиля

11.8.2 Конструктор(ы)

11.8.2.1 `xypLOT::HPROFILE::HPROFILE () [inline]`

Конструктор по умолчанию.

11.8.2.2 `xypLOT::HPROFILE::HPROFILE (long n) [inline], [explicit], [protected]`

11.8.3 Документация по друзьям класса и функциям, относящимся к классу

11.8.3.1 `HPROFILE XYPlotManager::CreateProfile (HPLOT & hPlot, const char * name, COLORREF color, int width, int lineType, const char * szLineTemplate, BOOL visible, BOOL showmarks, long xParentAxis, long yParentAxis) [friend]`

Дружественная функция из класса [XYPlotManager](#) для создания экземпляра профиля

Аргументы

<code>hPlot</code>	дескриптор экземпляра 2D графика для которого создается профиль
<code>name</code>	имя профиля (отображается в легенде)
<code>color</code>	цвет профиля
<code>width</code>	толщина линии профиля в пикселях
<code>lineType</code>	тип линии профиля <ul style="list-style-type: none"> • <code>PLS_INVISIBLE</code> - невидимая • <code>PLS_SOLID</code> - сплошная • <code>PLS_DASH</code> - штриховая • <code>PLS_DOT</code> - точечная • <code>PLS_DASHDOT</code> - осевая • <code>PLS_DASHDOTDOT</code> - штрих-точка-точка • <code>PLS_CUSTOM</code> - пользовательская

szLineTemplate	пользовательский шаблон линии. Данный аргумент используется только в случае если выбран пользовательский тип линии. Шаблон представляет собой последовательность целочисленных значений для задания интервалов заполнения и промежутков в пикселях. Например значение шаблона "5 2" означает, что 5px отрисовываются сплошной линией затем следует промежуток 2px.
visible	будет ли отображаться линия данного профиля
showmarks	отображать ли метки данных профиля
xParentAxis	ось X, к которой прикрепляется профиль
yParentAxis	ось Y, к которой прикрепляется профиль

Возвращает

дескриптор экземпляра профиля

```

XYPlotManager& pm = XYPlotManager::Instance();
if ( !pm.Initialize("xyplot.dll", ::AfxGetMainWnd()->m_hWnd))
    return FALSE;
HPLOT plot;
CChildView* pView = ...//получить дескриптор окна в котором будет отображаться графика
try {
    plot = pm.CreatePlot(pView->m_hWnd)
}
catch (XYPlotRequestFailure& e)
{
    return FALSE;
}

```

11.8.3.2 BOOL XYPlotManager::GetProfileList (HPLOT & , std::vector< HPROFILE > &)
[friend]

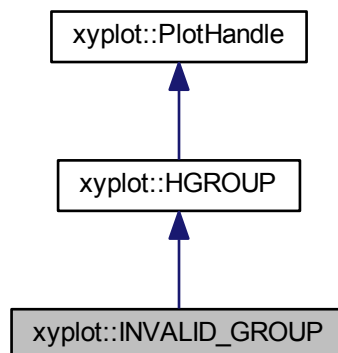
Объявления и описания членов класса находятся в файле:

- [src/XYPlotWrapper.h](#)

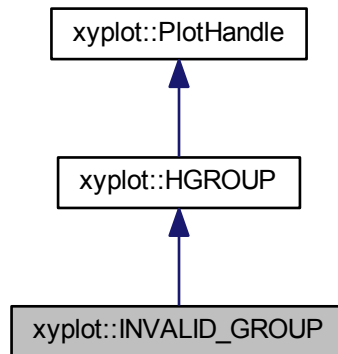
11.9 Класс xyplot::INVALID_GROUP

```
#include <XYPlotWrapper.h>
```

Граф наследования:xyplot::INVALID_GROUP:



Граф связей класса `xyplot::INVALID_GROUP`:



Открытые члены

- [`INVALID_GROUP\(\)`](#)

Дополнительные унаследованные члены

11.9.1 Конструктор(ы)

11.9.1.1 `xyplot::INVALID_GROUP::INVALID_GROUP () [inline]`

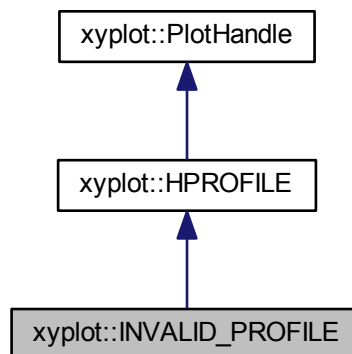
Объявления и описания членов класса находятся в файле:

- [src/XYPlotWrapper.h](#)

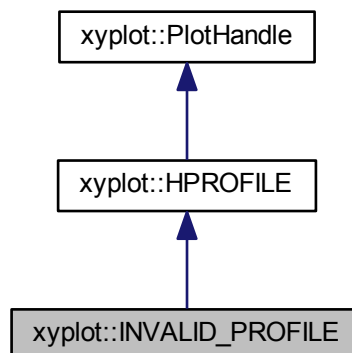
11.10 Класс `xyplot::INVALID_PROFILE`

```
#include <XYPlotWrapper.h>
```


Граф наследования: `xyplot::INVALID_PROFILE`:



Граф связей класса `xyplot::INVALID_PROFILE`:



Открытые члены

- [`INVALID_PROFILE\(\)`](#)

Дополнительные унаследованные члены

11.10.1 Конструктор(ы)

11.10.1.1 `xyplot::INVALID_PROFILE::INVALID_PROFILE ()` [inline]

Объявления и описания членов класса находятся в файле:

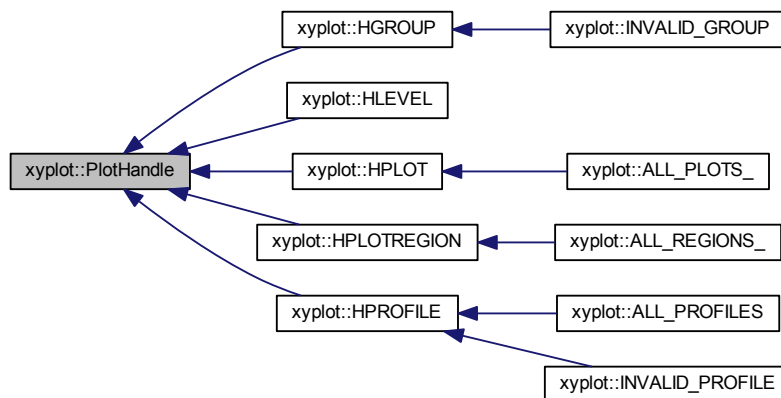
- [src/XYPlotWrapper.h](#)

11.11 Класс xyplot::PlotHandle

Базовый класс дескрипторов 2D графики

```
#include <XYPlotWrapper.h>
```

Граф наследования:xyplot::PlotHandle:



Открытые члены

- virtual `~PlotHandle ()`
- const `PlotHandle & operator= (const PlotHandle &rhs)`
- bool `operator! () const`
- bool `operator!= (const PlotHandle &rhs) const`
- `operator long () const`
- `long * operator& ()`
- `PlotHandle (long n)`
- `PlotHandle (const PlotHandle &rhs)`

Защищенные члены

- `PlotHandle ()`

11.11.1 Подробное описание

Базовый класс дескрипторов 2D графики

11.11.2 Конструктор(ы)

11.11.2.1 `virtual xyplot::PlotHandle::~~PlotHandle () [inline], [virtual]`

Деструктор

11.11.2.2 `xyplot::PlotHandle::PlotHandle () [inline], [protected]`

Конструктор по умолчанию. Инициализирует пустой дескриптор (= 0)

11.11.2.3 `xypLOT::PlotHandle::PlotHandle (long n)` [inline]

Конструктор с параметром.

Аргументы

n	- ???
---	-------

11.11.2.4 `xypLOT::PlotHandle::PlotHandle (const PlotHandle & rhs) [inline]`

Конструктор копирования.

Аргументы

rhs	ссылка на существующий дескриптор
-----	-----------------------------------

11.11.3 Методы

11.11.3.1 `xypLOT::PlotHandle::operator long () const [inline]`

Оператор приведения к long.

Возвращает

дескриптор в виде long

11.11.3.2 `bool xypLOT::PlotHandle::operator! () const [inline]`

Оператор логическое нет

Возвращает

true если дескриптор пустой (= 0)
false если дескриптор не нулевой

11.11.3.3 `bool xypLOT::PlotHandle::operator!= (const PlotHandle & rhs) const [inline]`

Оператор не равно. Сравнивает с существующим дескриптором

Аргументы

rhs	ссылка на существующий дескриптор
-----	-----------------------------------

Возвращает

true если дескрипторы не равны
false если дескрипторы равны

11.11.3.4 `long* xypLOT::PlotHandle::operator& () [inline]`

Унарный оператор & возвращает адрес дескриптора.

Возвращает

возвращает адрес дескриптора

11.11.3.5 `const PlotHandle& xypLOT::PlotHandle::operator= (const PlotHandle & rhs) [inline]`

Оператор присваивания

Аргументы

rhs	ссылка на существующий дескриптор
-----	-----------------------------------

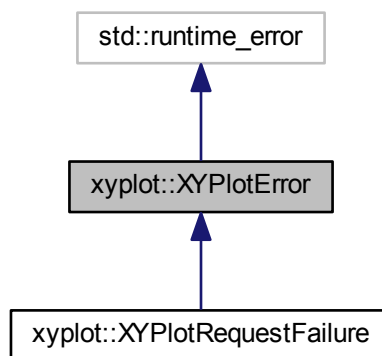
Объявления и описания членов класса находятся в файле:

- `src/XYPlotWrapper.h`

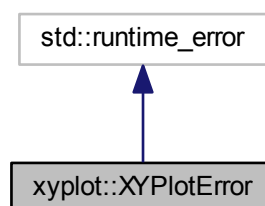
11.12 Класс `xyplot::XYPlotError`

```
#include <XYPlotWrapper.h>
```

Граф наследования:`xyplot::XYPlotError`:



Граф связей класса `xyplot::XYPlotError`:



Открытые члены

- [XYPlotError](#) (`const std::string &message`)

11.12.1 Конструктор(ы)

11.12.1.1 `xyplot::XYPlotError::XYPlotError (const std::string & message) [inline]`

Объявления и описания членов класса находятся в файле:

- `src/XYPlotWrapper.h`

11.13 Класс `xyplot::XYPlotManager`

`#include <XYPlotWrapper.h>`

Открытые члены

- `int GetLastError () const`
- `std::string TranslateError (int code) const`
- `BOOL Initialize (const char *pszDllPath, HWND hwndParent=NULL)`
- `HPlot CreatePlot (HWND hwndParent)`
- `BOOL DestroyPlot (HPlot &hPlot)`
- `BOOL CopyToClipboard (HPlot &hPlot, unsigned width, unsigned height, unsigned dpi)`
- `BOOL DrawToBitmap (HPlot &hPlot, HBITMAP &hBitmap, bool bDrawWindowBG, bool bDrawClientBG)`
- `BOOL RunDialog (HPlot &hPlot)`
- `BOOL RegRestoreDefaults (HPlot &hPlot)`
- `BOOL RegLoadSettings (HPlot &hPlot, const char *pszKey, const char *pszValue)`
- `BOOL RegSaveSettings (HPlot &hPlot, const char *pszKey, const char *pszValue)`
- `BOOL SaveProperties (HPlot &hPlot, const char *pszFileName)`
- `BOOL LoadProperties (HPlot &hPlot, const char *pszFileName)`
- `BOOL Select (HPlot &hPlot, HPROFILE &hProfile, unsigned nIndex)`
- `BOOL Select (HPlot &hPlot, HGROUП &hGroup, unsigned nIndex)`
- `BOOL SetSelectionMode (HPlot &hPlot, int mode)`
- `long GetSelectionMode (HPlot &hPlot) const`
- `BOOL SetFlashSpeed (HPlot &hPlot, int speed)`
- `BOOL SetRedraw (HPlot &hPlot, BOOL bRedraw=TRUE)`
- `BOOL SetAxisRange (HPlot &hPlot, long axis, double dMin, double dMax)`
- `BOOL GetAxisRange (HPlot &hPlot, long axis, double &dMin, double &dMax) const`
- `std::string GetString (HPlot &hPlot, long item, long subitem) const`
- `BOOL SetString (HPlot &hPlot, long item, long subitem, const char *pszString)`
- `long GetLong (HPlot &hPlot, long item, long subitem) const`
- `BOOL SetLong (HPlot &hPlot, long item, long subitem, long value)`
- `double GetDouble (HPlot &hPlot, long item, long subitem) const`
- `BOOL SetDouble (HPlot &hPlot, long item, long subitem, double value)`
- `BOOL EnableItem (HPlot &hPlot, long item, long subitem, BOOL enabled)`
- `BOOL IsEnabled (HPlot &hPlot, long item, long subitem) const`
- `COLORREF GetColor (HPlot &hPlot, long item, long subitem) const`
- `BOOL SetColor (HPlot &hPlot, long item, long subitem, COLORREF color)`
- `BOOL SetFont (HPlot &hPlot, long item, long subitem, const LOGFONT *font)`
- `HPROFILE CreateProfile (HPlot &hPlot, const char *name, COLORREF color, int width, int lineType=xyplot::PLS_SOLID, const char *szLineTemplate=0, BOOL visible=TRUE, BOOL showmarks=FALSE, const long xParentAxis=xyplot::PI_AXIS_BOTTOM, const long yParentAxis=xyplot::PI_AXIS_LEFT)`
- `BOOL DeleteProfile (HPlot &hPlot, HPROFILE &hProfile)`
- `HGROUP CreateGroup (HPlot &hPlot)`
- `BOOL DeleteGroup (HPlot &hPlot, HGROUП &hGroup)`
- `BOOL BindProfileToGroup (HPlot &hPlot, HGROUП &hGroup, HPROFILE &hProfile)`

- **BOOL** [GetProfileList](#) (**HPlot** &hPlot, std::vector< **HProfile** > &hProfiles)
- **BOOL** [SetData](#) (**HPlot** &hPlot, **HProfile** &hProfile, const double *px, const double *py, unsigned size)
- **BOOL** [GetData](#) (**HPlot** &hPlot, **HProfile** &hProfile, double *px, double *py, unsigned *size)
- **BOOL** [AppendData](#) (**HPlot** &hPlot, **HProfile** &hProfile, const double *px, const double *py, unsigned size)
- **HProfile** [GetSelectedProfile](#) (**HPlot** &hPlot) const
- **COLORREF** [GetColor](#) (**HPlot** &hPlot, **HProfile** &hProfile) const
- **BOOL** [SetColor](#) (**HPlot** &hPlot, **HProfile** &hProfile, **COLORREF** color)
- **HLEVEL** [CreateLevel](#) (**HPlot** &hPlot, long axis, double val, const char *pszTitle, **COLORREF** clr, int width)
- **LEVELS**
- **BOOL** [DeleteLevel](#) (**HPlot** &hPlot, **HLEVEL** &hLevel)
- **BOOL** [SetLevel](#) (**HPlot** &hPlot, **HLEVEL** &hLevel, double value)
- **HPlotRegion** [CreateRegion](#) (**HPlot** &hPlot, long AxisX, long AxisY, double xFrom, double xTo, double yFrom, double yTo, const char *pszTitle)
- **BOOL** [DeleteRegion](#) (**HPlot** &hPlot, **HPlotRegion** &hRegion)
- **BOOL** [SetRegion](#) (**HPlot** &hPlot, **HPlotRegion** &hRegion, double x1, double x2, double y1, double y2)
- [~XYPlotManager](#) ()

Открытые статические члены

- static [XYPlotManager](#) & [Instance](#) ()

Функция доступа к единственному экземпляру класса Класс [XYPlotManager](#) является синглтоном т.е. существует единственный экземпляр объекта данного класса. Доступ к нему осуществляется при помощи данной функции.

Защищенные члены

- [XYPlotManager](#) ()
- [XYPlotManager](#) (const [XYPlotManager](#) &rhs)

Защищенные данные

- **HMODULE** [m_hDllInstance](#)
- **int** [m_nLastError](#)
- [xyplot::LPFNINITIALIZE](#) [m_lpfInitialize](#)
- [xyplot::LPFNFINALIZE](#) [m_lpfFinalize](#)
- [xyplot::LPFNADDPROFILE](#) [m_lpfAddProfile](#)
- [xyplot::LPFNDELETEPROFILE](#) [m_lpfDeleteProfile](#)
- [xyplot::LPFNSETDATA](#) [m_lpfSetData](#)
- [xyplot::LPFNGETDATA](#) [m_lpfGetData](#)
- [xyplot::LPFNAPPENDDATA](#) [m_lpfAppendData](#)
- [xyplot::LPFNSETSTRING](#) [m_lpfSetString](#)
- [xyplot::LPFNGETSTRING](#) [m_lpfGetString](#)
- [xyplot::LPFNSETCOLOR](#) [m_lpfSetColor](#)
- [xyplot::LPFNGETCOLOR](#) [m_lpfGetColor](#)
- [xyplot::LPFNSETFONT](#) [m_lpfSetFont](#)
- [xyplot::LPFNSETLONG](#) [m_lpfSetLong](#)
- [xyplot::LPFNGETLONG](#) [m_lpfGetLong](#)
- [xyplot::LPFNSETDOUBLE](#) [m_lpfSetDouble](#)
- [xyplot::LPFNGETDOUBLE](#) [m_lpfGetDouble](#)
- [xyplot::LPFNISENABLED](#) [m_lpfIsEnabled](#)

- `xyplot::LPFNENABLEITEM m_lpfmEnableItem`
- `xyplot::LPFNSAVEPROPERTIES m_lpfmSaveProperties`
- `xyplot::LPFNLOADPROPERTIES m_lpfmLoadProperties`
- `xyplot::LPFNSELECT m_lpfmSelect`
- `xyplot::LPFNGETSELECTION m_lpfmGetSelection`
- `xyplot::LPFNCOPYTOCLIPBOARD m_lpfmCopyToClipboard`
- `xyplot::LPFNDRAWTOBITMAP m_lpfmDrawToBitmap`
- `xyplot::LPFNCREATEGROUP m_lpfmCreateGroup`
- `xyplot::LPFNDELETEDGROUP m_lpfmDeleteGroup`
- `xyplot::LPFNBINDPROFILE m_lpfmBindProfile`
- `xyplot::LPFNGETPROFLIST m_lpfmGetProfileList`
- `xyplot::LPFNREGSTOREDEFAULTS m_lpfmRegRestoreDefaults`
- `xyplot::LPFNREGLOADSETTINGS m_lpfmRegLoadSettings`
- `xyplot::LPFNREGSAVESETTINGS m_lpfmRegSaveSettings`
- `xyplot::LPFNRUNDIALOG m_lpfmRunDialog`
- `xyplot::LPFNADDREGION m_lpfmAddRegion`
- `xyplot::LPFNDELETEREGION m_lpfmDeleteRegion`
- `xyplot::LPFNSETREGION m_lpfmSetRegion`
- `xyplot::LPFNADDLEVEL m_lpfmAddLevel`
- `xyplot::LPFNDELETELEVEL m_lpfmDeleteLevel`
- `xyplot::LPFNSETLEVEL m_lpfmSetLevel`

11.13.1 Подробное описание

Класс-обертка для работы с объектами отображения 2D графики. Данный класс предоставляет инструменты создания, манипуляции и уничтожение объектов 2D графики. Класс является синглтоном, т.е. существует только один экземпляр данного класса, доступ к которому осуществляется через статическую функцию `Instance`.

11.13.2 Конструктор(ы)

11.13.2.1 `xyplot::XYPlotManager::~~XYPlotManager () [inline]`

11.13.2.2 `xyplot::XYPlotManager::XYPlotManager () [inline], [protected]`

11.13.2.3 `xyplot::XYPlotManager::XYPlotManager (const XYPlotManager & rhs) [protected]`

11.13.3 Методы

11.13.3.1 `BOOL XYPlotManager::CopyToClipboard (HPLOT & hPlot, unsigned width, unsigned height, unsigned dpi)`

Функция копирует текущее содержимое экземпляра 2D графики в буфер обмена в виде изображения.

Аргументы

<code>hPlot</code>	- дескриптор экземпляра 2D графики
<code>width</code>	- ширина результирующего изображения в пикселях
<code>height</code>	- высота результирующего изображения в пикселях
<code>dpi</code>	- разрешение изображения

Возвращает

`true` - в случае если копирование прошло успешно;

`false` - в случае если не удалось скопировать экземпляр в буфер обмена.

11.13.3.2 HLEVEL XYPlotManager::CreateLevel (HPLOT & hPlot, long axis, double val, const char * pszTitle, COLORREF clr, int width)

LEVELS.

11.13.3.3 HPLOTREGION XYPlotManager::CreateRegion (HPLOT & hPlot, long AxisX, long AxisY, double xFrom, double xTo, double yFrom, double yTo, const char * pszTitle)

11.13.3.4 BOOL XYPlotManager::DeleteLevel (HPLOT & hPlot, HLEVEL & hLevel)

11.13.3.5 BOOL XYPlotManager::DeleteRegion (HPLOT & hPlot, HPLOTREGION & hRegion)

11.13.3.6 BOOL XYPlotManager::DrawToBitmap (HPLOT & hPlot, HBITMAP & hBitmap, bool bDrawWindowBG, bool bDrawClientBG)

Функция копирует текущее содержимое экземпляра 2D в буфер битовую карту.

Аргументы

hPlot	- дескриптор экземпляра 2D графики
hBitmap	- дескриптор битовой карты
bDrawWindowBG	- флаг отрисовки фона окна
bDrawClientBG	- флаг отрисовки фона клиентской части

Возвращает

true - в случае если копирование прошло успешно;

false - в случае если не удалось скопировать экземпляр в буфер обмена.

11.13.3.7 COLORREF xyplot::XYPlotManager::GetColor (HPLOT & hPlot, HPROFILE & hProfile) const

11.13.3.8 int xyplot::XYPlotManager::GetLastError () const [inline]

Функция получения кода последней ошибки

Возвращает

целочисленный код последней ошибки

11.13.3.9 HPROFILE XYPlotManager::GetSelectedProfile (HPLOT & hPlot) const

11.13.3.10 XYPlotManager & XYPlotManager::Instance () [static]

Функция доступа к единственному экземпляру класса Класс [XYPlotManager](#) является синглетном т.е. существует единственный экземпляр объекта данного класса. Доступ к нему осуществляется при помощи данной функции.

[XYPlotManager](#)& pm = [XYPlotManager::Instance](#)();

Возвращает

Ссылка на экземпляр

11.13.3.11 `BOOL XYPlotManager::RunDialog (HPLOT & hPlot)`

Функция вызывает диалог настроек экземпляра 2D графики .

Аргументы

hPlot	- дескриптор экземпляра 2D графики
-------	------------------------------------

Возвращает

true - в случае если запуск диалога прошел успешно;
false - в случае если не удалось запустить диалог настроек.

11.13.3.12 `BOOL xyplot::XYPlotManager::SetColor (HPLOT & hPlot, HPROFILE & hProfile, COLORREF color)`

11.13.3.13 `BOOL XYPlotManager::SetLevel (HPLOT & hPlot, HLEVEL & hLevel, double value)`

11.13.3.14 `BOOL XYPlotManager::SetRedraw (HPLOT & hPlot, BOOL bRedraw = TRUE)`

Функция разрешает либо запрещает перерисовку экземпляра 2D графики

Аргументы

hPlot	- дескриптор экземпляра 2D графики
bRedraw	- флаг разрешения отрисовки если bRedraw = TRUE окно обновляется после каждой операции. В случае если bRedraw = FALSE окно не обновляется до того момента вызова функции с параметром bRedraw = TRUE <pre> XYPlotManager& pm = XYPlotManager::Instance(); double* pdX = NULL; double* pdY = NULL; HPLOT hPlot = ...; //Получить дескриптор 2D графики pm.SetRedraw(hPlot, FALSE); for (int i = 0; i < nProfileCount; i++) { //получить данные для профиля ... HPROFILE hProf = ...; //получить дескриптор профиля //установить данные профиля pm.SetData(hPlot, hProf, pdX, pdY, nSize); ...//прочие операции } pm.SetRedraw(hPlot, TRUE); </pre>

11.13.3.15 `BOOL XYPlotManager::SetRegion (HPLOT & hPlot, HPLOTREGION & hRegion, double x1, double x2, double y1, double y2)`

11.13.3.16 `std::string XYPlotManager::TranslateError (int code) const`

Расшифровывает код ошибки

Аргументы

code	- Код ошибки
------	--------------

Возвращает

Возвращает строковое описание ошибки

11.13.4 Данные класса

11.13.4.1 `HMODULE xyplot::XYPlotManager::m_hDllInstance` [protected]

11.13.4.2 `xyplot::LPFNADDLEVEL xyplot::XYPlotManager::m_lpfAddLevel` [protected]

- 11.13.4.3 xyplot::LPFNADDPFILE xyplot::XYPlotManager::m_lpfnAddProfile [protected]
- 11.13.4.4 xyplot::LPFNADDREGION xyplot::XYPlotManager::m_lpfnAddRegion [protected]
- 11.13.4.5 xyplot::LPFNAPPENDDATA xyplot::XYPlotManager::m_lpfnAppendData [protected]
- 11.13.4.6 xyplot::LPFNBINDPROFILE xyplot::XYPlotManager::m_lpfnBindProfile [protected]
- 11.13.4.7 xyplot::LPFNCOPYTOCLIPBOARD xyplot::XYPlotManager::m_lpfnCopyToClipboard [protected]
- 11.13.4.8 xyplot::LPFNCREATEGROUP xyplot::XYPlotManager::m_lpfnCreateGroup [protected]
- 11.13.4.9 xyplot::LPFNDELETEGROUP xyplot::XYPlotManager::m_lpfnDeleteGroup [protected]
- 11.13.4.10 xyplot::LPFNDELETELEVEL xyplot::XYPlotManager::m_lpfnDeleteLevel [protected]
- 11.13.4.11 xyplot::LPFNDELETEPROFILE xyplot::XYPlotManager::m_lpfnDeleteProfile [protected]
- 11.13.4.12 xyplot::LPFNDELETEREGION xyplot::XYPlotManager::m_lpfnDeleteRegion [protected]
- 11.13.4.13 xyplot::LPFNDRAWTOBITMAP xyplot::XYPlotManager::m_lpfnDrawToBitmap [protected]
- 11.13.4.14 xyplot::LPFNENABLEITEM xyplot::XYPlotManager::m_lpfnEnableItem [protected]
- 11.13.4.15 xyplot::LPFNFINALIZE xyplot::XYPlotManager::m_lpfnFinalize [protected]
- 11.13.4.16 xyplot::LPFNGETCOLOR xyplot::XYPlotManager::m_lpfnGetColor [protected]
- 11.13.4.17 xyplot::LPFNGETDATA xyplot::XYPlotManager::m_lpfnGetData [protected]
- 11.13.4.18 xyplot::LPFNGETDOUBLE xyplot::XYPlotManager::m_lpfnGetDouble [protected]
- 11.13.4.19 xyplot::LPFNGETLONG xyplot::XYPlotManager::m_lpfnGetLong [protected]
- 11.13.4.20 xyplot::LPFNGETPROFLIST xyplot::XYPlotManager::m_lpfnGetProfileList [protected]
- 11.13.4.21 xyplot::LPFNGETSELECTION xyplot::XYPlotManager::m_lpfnGetSelection [protected]
- 11.13.4.22 xyplot::LPFNGETSTRING xyplot::XYPlotManager::m_lpfnGetString [protected]
- 11.13.4.23 xyplot::LPFNINITIALIZE xyplot::XYPlotManager::m_lpfnInitialize [protected]
- 11.13.4.24 xyplot::LPFNISENABLED xyplot::XYPlotManager::m_lpfnIsEnabled [protected]
- 11.13.4.25 xyplot::LPFNLOADPROPERTIES xyplot::XYPlotManager::m_lpfnLoadProperties [protected]
- 11.13.4.26 xyplot::LPFNREGLOADSETTINGS xyplot::XYPlotManager::m_lpfnRegLoadSettings [protected]
- 11.13.4.27 xyplot::LPFNREGRESTOREDEFAULTS xyplot::XYPlotManager::m_lpfnRegRestore-Defaults [protected]

- 11.13.4.28 xyplot::LPFNREGSAVESETTINGS xyplot::XYPlotManager::m_lpfnRegSaveSettings [protected]
- 11.13.4.29 xyplot::LPFNRRUNDIALOG xyplot::XYPlotManager::m_lpfnRunDialog [protected]
- 11.13.4.30 xyplot::LPFNSAVEPROPERTIES xyplot::XYPlotManager::m_lpfnSaveProperties [protected]
- 11.13.4.31 xyplot::LPFNSELECT xyplot::XYPlotManager::m_lpfnSelect [protected]
- 11.13.4.32 xyplot::LPFNSETCOLOR xyplot::XYPlotManager::m_lpfnSetColor [protected]
- 11.13.4.33 xyplot::LPFNSETDATA xyplot::XYPlotManager::m_lpfnSetData [protected]
- 11.13.4.34 xyplot::LPFNSETDOUBLE xyplot::XYPlotManager::m_lpfnSetDouble [protected]
- 11.13.4.35 xyplot::LPFNSETFONT xyplot::XYPlotManager::m_lpfnSetFont [protected]
- 11.13.4.36 xyplot::LPFNSETLEVEL xyplot::XYPlotManager::m_lpfnSetLevel [protected]
- 11.13.4.37 xyplot::LPFNSETLONG xyplot::XYPlotManager::m_lpfnSetLong [protected]
- 11.13.4.38 xyplot::LPFNSETREGION xyplot::XYPlotManager::m_lpfnSetRegion [protected]
- 11.13.4.39 xyplot::LPFNSETSTRING xyplot::XYPlotManager::m_lpfnSetString [protected]
- 11.13.4.40 int xyplot::XYPlotManager::m_nLastError [mutable], [protected]

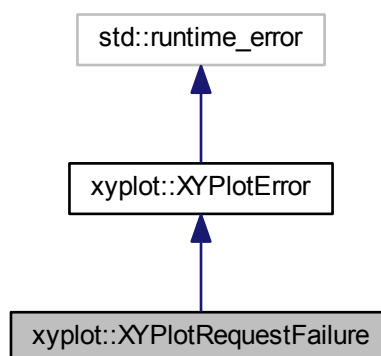
Объявления и описания членов классов находятся в файлах:

- src/[XYPlotWrapper.h](#)
- src/[XYPlotWrapper.cpp](#)

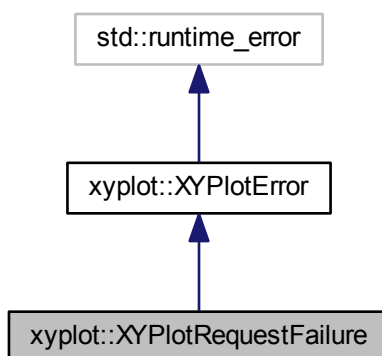
11.14 Класс xyplot::XYPlotRequestFailure

```
#include <XYPlotWrapper.h>
```

Граф наследования:xyplot::XYPlotRequestFailure:



Граф связей класса `xyplot::XYPlotRequestFailure`:



Открытые члены

- [XYPlotRequestFailure](#) (const std::string &msg)

11.14.1 Конструктор(ы)

11.14.1.1 `xyplot::XYPlotRequestFailure::XYPlotRequestFailure (const std::string & msg) [inline]`

Объявления и описания членов класса находятся в файле:

- [src/XYPlotWrapper.h](#)

Глава 12

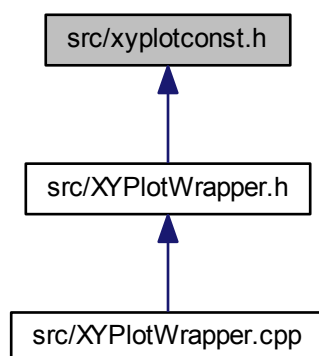
Файлы

12.1 Файл `src/documentation_ru.h`

12.2 Файл `src/xyplotconst.h`

Содержит константы и перечисляемые типы .

Граф файлов, в которые включается этот файл:



Пространства имен

- `xyplot`

Макросы

- `#define _XY_PLOTCONST_H_`

Определения типов

- `typedef int xyplot::FLASH_SPEED`

Перечисления

- enum `xyplot::ePLOTERRORS` {
`xyplot::PE_NOERROR`, `xyplot::PE_UNKNOWN`, `xyplot::PE_UNEXPECTED`, `xyplot::PE_INVALID_PLOT`,
`xyplot::PE_INVALID_PROFILE`, `xyplot::PE_INVALID_WINDOW`, `xyplot::PE_ALREADY_HOOKED`, `xyplot::PE_INSUFFICIENT_MEMORY`,
`xyplot::PE_INVALID_RANGE`, `xyplot::PE_INVALID_ARG`, `xyplot::PE_INSUFFICIENT_STORAGE`, `xyplot::PE_OUT_OF_RANGE`,
`xyplot::PE_INVALID_GROUP`, `xyplot::PE_RTFTTEXT`, `xyplot::PE_INVALID_DIALOG_DLL`, `xyplot::PE_DIALOG_DLL_NOT_FOUND` }

Коды ошибок библиотеки

- enum `xyplot::ePLOTLEMENT` {
`xyplot::PI_NOTHING`, `xyplot::PI_BACKGROUND`, `xyplot::PI_BACKGROUND_CLIENT`,
`xyplot::PI_BORDER`,
`xyplot::PI_TITLE`, `xyplot::PI_LEGEND`, `xyplot::PI_AXIS_LEFT`, `xyplot::PI_AXIS_RIGHT`,
`xyplot::PI_AXIS_TOP`, `xyplot::PI_AXIS_BOTTOM`, `xyplot::PI_SELECTION`, `xyplot::PI_CANVAS` }

Идентификаторы элементов 2D графика

- enum `xyplot::eSELECTIONITEMS` { `xyplot::SP_MODE`, `xyplot::SP_FLASH_SPEED`, `xyplot::SP_NOTIFY_MESSAGE` }

Подэлементы выделения

- enum `xyplot::eSELECTIONTYPE` {
`xyplot::SM_NONE` = -1, `xyplot::SM_FLASH`, `xyplot::SM_VTRACE`, `xyplot::SM_HTRACE`,
`xyplot::SM_TRACE` }

Тип выделения

- enum `xyplot::eFLASHSPEED` {
`xyplot::FLS_NO_FLASH`, `xyplot::FLS_VERY_SLOW`, `xyplot::FLS_SLOW`, `xyplot::FLS_NORMAL`,
`xyplot::FLS_FAST`, `xyplot::FLS_VERY_FAST`, `xyplot::FLS_EXTREME` }

Скорость мерцания профиля или группы профилейв режиме выделения `SM_FLASH`.

- enum `xyplot::eAXISITEMS` {
`xyplot::PAI_TITLE` = 200, `xyplot::PAI_MAJOR_TICKS`, `xyplot::PAI_MINOR_TICKS`,
`xyplot::PAI_MAJOR_TICKS_LENGTH`,
`xyplot::PAI_MINOR_TICKS_LENGTH`, `xyplot::PAI_MAJOR_TICKS_COUNT`, `xyplot::PAI_MINOR_TICKS_COUNT`, `xyplot::PAI_MAJOR_TICKS_WIDTH`,
`xyplot::PAI_MINOR_TICKS_WIDTH`, `xyplot::PAI_LINE`, `xyplot::PAI_LINE_WIDTH`,
`xyplot::PAI_LINE_STYLE`,
`xyplot::PAI_MAJOR_DIGITS`, `xyplot::PAI_MINOR_DIGITS`, `xyplot::PAI_MAJOR_DIGITS_FORMAT`, `xyplot::PAI_MINOR_DIGITS_FORMAT`,
`xyplot::PAI_MAJOR_GRID_LINES`, `xyplot::PAI_MINOR_GRID_LINES`, `xyplot::PAI_LOWER_LIMIT`, `xyplot::PAI_UPPER_LIMIT`,
`xyplot::PAI_AUTOSCALE`, `xyplot::PAI_LOG10`, `xyplot::PAI_STYLE`, `xyplot::PAI_TIME_HISTORY` }

Элементы осей

- enum `xyplot::eLEGENDITEMS` {
`xyplot::PLP_ALIGN` = 220, `xyplot::PLP_LINELENGTH` = 221, `xyplot::PLP_LEFT` = -1,
`xyplot::PLP_RIGHT` = 1,
`xyplot::PLP_BOTTOM` = 0 }

legend properties

- enum `xyplot::eLEVELLABELPOS` {
`xyplot::LLP_CENTER` = 0x001, `xyplot::LLP_LEFT` = 0x002, `xyplot::LLP_RIGHT` = 0x004,
`xyplot::LLP_ABOVE` = 0x008,
`xyplot::LLP_UNDER` = 0x010, `xyplot::LLP_ATLINE` = `LLP_ABOVE`|`LLP_UNDER`, `xyplot::LLP_ABOVELEFT` = `LLP_LEFT`|`LLP_ABOVE`, `xyplot::LLP_ABOVERIGHT` = `LLP_RIGHT`|`LLP_ABOVE`,
`xyplot::LLP_ABOVECENTER` = `LLP_CENTER`|`LLP_ABOVE`, `xyplot::LLP_UNDERLEFT`


```
T = LLP_LEFT|LLP_UNDER, xyplot::LLP_UNDERRIGHT = LLP_RIGHT|LLP_UNDER,
xyplot::LLP_UNDERCENTER = LLP_CENTER|LLP_UNDER,
xyplot::LLP_ATLINELEFT = LLP_LEFT|LLP_ATLINE, xyplot::LLP_ATLINERIGHT = LLP_RIGHT|LLP_ATLINE,
xyplot::LLP_ATLINECENTER = LLP_CENTER|LLP_ATLINE
}
```

Level label position - положение надписи уровня

- enum xyplot::LINETYPE {
 xyplot::PLS_INVISIBLE, xyplot::PLS_SOLID, xyplot::PLS_DASH, xyplot::PLS_DOT,
 xyplot::PLS_DASHDOT, xyplot::PLS_DASHDOTDOT, xyplot::PLS_CUSTOM }

Тип линии

- enum xyplot::ePROFILEITEMS {
 xyplot::PRP_COLOR = 1, xyplot::PRP_WIDTH, xyplot::PRP_STYLE, xyplot::PRP_TEMPLATE,
 xyplot::PRP_NAME, xyplot::PRP_MARKS, xyplot::PRP_MARKSFREQ, xyplot::PRP_MARKSTYPE,
 xyplot::PRP_MARKSIZE, xyplot::PRP_MARKFILLCOLOR, xyplot::PRP_MARKSTROKECOLOR }

Profile properties.

- enum xyplot::eMARKSTYPE {
 xyplot::PMT_CIRCLE, xyplot::PMT_SQUARE, xyplot::PMT_DIAMOND, xyplot::PMT_TRIANGLEUP,
 xyplot::PMT_TRIANGLEDOWN, xyplot::PMT_CROSS0, xyplot::PMT_CROSS45 }

Типы меток данных

- enum xyplot::eBACKGROUNDITEMS {
 xyplot::BGP_COLOR_START = 330, xyplot::BGP_COLOR_END, xyplot::BGP_DIRECTION,
 xyplot::BGP_GRADIENT_VERT = 1,
 xyplot::BGP_GRADIENT_HORZ = 0 }

Background properties.

12.2.1 Подробное описание

Содержит константы и перечисляемые типы .

Автор

Johnny A. Matveichik
Alexei N. Migoun

12.2.2 Макросы

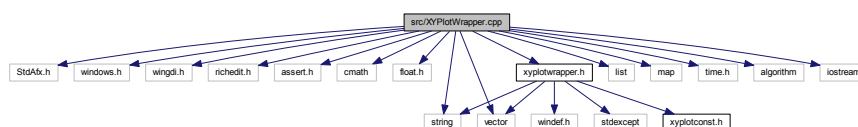
12.2.2.1 #define _XY_PLOTCONST_H_

12.3 Файл src/XYPlotWrapper.cpp

```
#include "StdAfx.h"
```

```
#include "xyplotwrapper.h"
```

Граф включаемых заголовочных файлов для XYPlotWrapper.cpp:



Макросы

```

• #define XYFUNNAME_INITIALIZE "_Initialize@8"
• #define XYFUNNAME_FINALIZE "_Finalize@4"
• #define XYFUNNAME_ADDPROFILE "_AddProfile@44"
• #define XYFUNNAME_DELPROFILE "_DeleteProfile@8"
• #define XYFUNNAME_SETDATA "_SetData@20"
• #define XYFUNNAME_GETDATA "_GetData@20"
• #define XYFUNNAME_APPENDDATA "_AppendData@20"
• #define XYFUNNAME_SETPROFCLR "_SetProfileColor@12"
• #define XYFUNNAME_GETPROFCLR "_GetProfileColor@12"
• #define XYFUNNAME_SETPROFSTL "_ChangeProfileStyle@24"
• #define XYFUNNAME_ENABLEMARK "_EnablePointsMarker@12"
• #define XYFUNNAME_SETSTRING "_SetString@16"
• #define XYFUNNAME_GETSTRING "_GetString@20"
• #define XYFUNNAME_SETCOLOR "_SetColor@16"
• #define XYFUNNAME_GETCOLOR "_GetColor@16"
• #define XYFUNNAME_SETLONG "_SetLong@16"
• #define XYFUNNAME_GETLONG "_GetLong@16"
• #define XYFUNNAME_SETDOUBLE "_SetDouble@20"
• #define XYFUNNAME_GETDOUBLE "_GetDouble@16"
• #define XYFUNNAME_SETSELMODE "_SetSelectionMode@8"
• #define XYFUNNAME_GETSELMODE "_GetSelectionMode@8"
• #define XYFUNNAME_ISENABLED "_IsEnabled@16"
• #define XYFUNNAME_SETAXRANGE "_SetAxisRange@24"
• #define XYFUNNAME_GETAXRANGE "_GetAxisRange@16"
• #define XYFUNNAME_GETMARKPOS "_GetMarkerPos@12"
• #define XYFUNNAME_SETMARKPOS "_SetMarkerPos@20"
• #define XYFUNNAME_GETSELPROF "_GetSelection@8"
• #define XYFUNNAME_SETFLASH "_SetFlashSpeed@8"
• #define XYFUNNAME_CLIPCOPY "_CopyToClipboard@16"
• #define XYFUNNAME_SETREDRAW "_SetRedraw@8"
• #define XYFUNNAME_CREATEGRP "_CreateGroup@8"
• #define XYFUNNAME_DELETEGRP "_DeleteGroup@8"
• #define XYFUNNAME_BINDPROF "_BindProfile@12"
• #define XYFUNNAME_GETPROFLIST "_GetProfileList@12"
• #define XYFUNNAME_SAVEPROP "_SaveProperties@8"
• #define XYFUNNAME_LOADPROP "_LoadProperties@8"
• #define XYFUNNAME_REGRESTORE "_RegRestoreDefaults@4"
• #define XYFUNNAME_REGLOAD "_RegLoadSettings@12"
• #define XYFUNNAME_REGSAVE "_RegSaveSettings@12"
• #define XYFUNNAME_RUNDIALOG "_RunDialog@4"
• #define XYFUNNAME_ADDLEVEL "_AddLevel@32"
• #define XYFUNNAME_DELEVEL "_DeleteLevel@8"
• #define XYFUNNAME_SETLEVEL "_SetLevel@16"
• #define XYFUNNAME_ADDREGION "_AddRegion@52"
• #define XYFUNNAME_DEREGION "_DeleteRegion@8"
• #define XYFUNNAME_SETREGION "_SetRegion@40"
• #define XYFUNNAME_SETFONT "_SetFont@16"
• #define XYFUNNAME_ENITEM "_EnableItem@16"
• #define XYFUNNAME_FLASHPROF "_Select@12"
• #define XYFUNNAME_DRAWTOBITMAP "_DrawToBitmap@16"
• #define CHECK_FUNCTION_POINTER(a) if (!(a)) return FALSE;

```

12.3.1 Макросы

12.3.1.1 `#define CHECK_FUNCTION_POINTER(a) if (!(a)) return FALSE;`

12.3.1.2 `#define XYFUNNAME_ADDLEVEL "_AddLevel@32"`

12.3.1.3 `#define XYFUNNAME_ADDPROFILE "_AddProfile@44"`

12.3.1.4 `#define XYFUNNAME_ADDREGION "_AddRegion@52"`

12.3.1.5 `#define XYFUNNAME_APPENDDATA "_AppendData@20"`

12.3.1.6 `#define XYFUNNAME_BINDPROF "_BindProfile@12"`

12.3.1.7 `#define XYFUNNAME_CLIPCOPY "_CopyToClipboard@16"`

12.3.1.8 `#define XYFUNNAME_CREATEGRP "_CreateGroup@8"`

12.3.1.9 `#define XYFUNNAME_DELETEGRP "_DeleteGroup@8"`

12.3.1.10 `#define XYFUNNAME_DELLEVEL "_DeleteLevel@8"`

12.3.1.11 `#define XYFUNNAME_DELPROFILE "_DeleteProfile@8"`

12.3.1.12 `#define XYFUNNAME_DELREGION "_DeleteRegion@8"`

12.3.1.13 `#define XYFUNNAME_DRAWTOBITMAP "_DrawToBitmap@16"`

12.3.1.14 `#define XYFUNNAME_ENABLEMARK "_EnablePointsMarker@12"`

12.3.1.15 `#define XYFUNNAME_ENITEM "_EnableItem@16"`

12.3.1.16 `#define XYFUNNAME_FINALIZE "_Finalize@4"`

12.3.1.17 `#define XYFUNNAME_FLASHPROF "_Select@12"`

12.3.1.18 `#define XYFUNNAME_GETAXRANGE "_GetAxisRange@16"`

12.3.1.19 `#define XYFUNNAME_GETCOLOR "_GetColor@16"`

12.3.1.20 `#define XYFUNNAME_GETDATA "_GetData@20"`

12.3.1.21 `#define XYFUNNAME_GETDOUBLE "_GetDouble@16"`

12.3.1.22 `#define XYFUNNAME_GETLONG "_GetLong@16"`

12.3.1.23 `#define XYFUNNAME_GETMARKPOS "_GetMarkerPos@12"`

12.3.1.24 `#define XYFUNNAME_GETPROFCLR "_GetProfileColor@12"`

12.3.1.25 `#define XYFUNNAME_GETPROFLIST "_GetProfileList@12"`

12.3.1.26 `#define XYFUNNAME_GETSELMODE "_GetSelectionMode@8"`

12.3.1.27 `#define XYFUNNAME_GETSELPROF "_GetSelection@8"`

12.3.1.28 `#define XYFUNNAME_GETSTRING "_GetString@20"`

```

12.3.1.29 #define XYFUNNAME__INITIALIZE "__Initialize@8"
12.3.1.30 #define XYFUNNAME__ISENABLED "__IsEnabled@16"
12.3.1.31 #define XYFUNNAME__LOADPROP "__LoadProperties@8"
12.3.1.32 #define XYFUNNAME__REGLOAD "__RegLoadSettings@12"
12.3.1.33 #define XYFUNNAME__REGRESTORE "__RegRestoreDefaults@4"
12.3.1.34 #define XYFUNNAME__REGSAVE "__RegSaveSettings@12"
12.3.1.35 #define XYFUNNAME__RUNDIALOG "__RunDialog@4"
12.3.1.36 #define XYFUNNAME__SAVEPROP "__SaveProperties@8"
12.3.1.37 #define XYFUNNAME__SETAXRANGE "__SetAxisRange@24"
12.3.1.38 #define XYFUNNAME__SETCOLOR "__SetColor@16"
12.3.1.39 #define XYFUNNAME__SETDATA "__SetData@20"
12.3.1.40 #define XYFUNNAME__SETDOUBLE "__SetDouble@20"
12.3.1.41 #define XYFUNNAME__SETFLASH "__SetFlashSpeed@8"
12.3.1.42 #define XYFUNNAME__SETFONT "__SetFont@16"
12.3.1.43 #define XYFUNNAME__SETLEVEL "__SetLevel@16"
12.3.1.44 #define XYFUNNAME__SETLONG "__SetLong@16"
12.3.1.45 #define XYFUNNAME__SETMARKPOS "__SetMarkerPos@20"
12.3.1.46 #define XYFUNNAME__SETPROFCLR "__SetProfileColor@12"
12.3.1.47 #define XYFUNNAME__SETPROFSTL "__ChangeProfileStyle@24"
12.3.1.48 #define XYFUNNAME__SETREDRAW "__SetRedraw@8"
12.3.1.49 #define XYFUNNAME__SETREGION "__SetRegion@40"
12.3.1.50 #define XYFUNNAME__SETSELMODE "__SetSelectionMode@8"
12.3.1.51 #define XYFUNNAME__SETSTRING "__SetString@16"

```

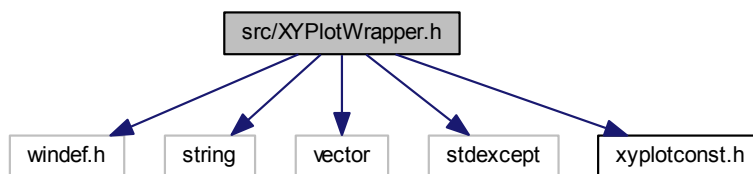
12.4 Файл src/XYPlotWrapper.h

```

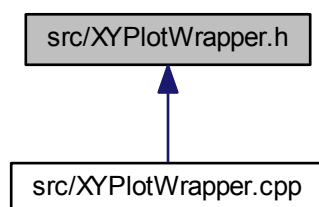
#include <windef.h>
#include <string>
#include <vector>
#include <stdexcept>
#include "xyplotconst.h"

```

Граф включаемых заголовочных файлов для XYPlotWrapper.h:



Граф файлов, в которые включается этот файл:



Классы

- class [xyplot::XYPlotManager](#)
- class [xyplot::PlotHandle](#)
Базовый класс дескрипторов 2D графики
- class [xyplot::HPLOT](#)
Класс дескриптора 2D графики
- class [xyplot::ALL_PLOTS_](#)
Класс дескриптор для всех экземпляров 2D графики
- class [xyplot::HPROFILE](#)
Класс дескриптора профиля
- class [xyplot::ALL_PROFILES](#)
- class [xyplot::INVALID_PROFILE](#)
- class [xyplot::HGROUP](#)
- class [xyplot::INVALID_GROUP](#)
- class [xyplot::HPLOTREGION](#)
- class [xyplot::ALL_REGIONS_](#)
- class [xyplot::HLEVEL](#)
- class [xyplot::XYPlotError](#)
- class [xyplot::XYPlotRequestFailure](#)

Пространства имен

- [xyplot](#)

Макросы

- `#define XYPLOTFN __stdcall`
- `#define ALL_PLOTS ALL_PLOTS_()`
- `#define ALL_REGIONS ALL_REGIONS_()`

Определения типов

- `typedef long(XYPLOTFN * xyplot::LPFNINITIALIZE)(HWND hwndMain, long *lPlotID)`
- `typedef long(XYPLOTFN * xyplot::LPFNFINALIZE)(long lPlotID)`
- `typedef long(XYPLOTFN * xyplot::LPFNADDPROFILE)(long lPlotID, const char *name, COLORREF color, int width, int linetype, const char *pszTemplate, BOOL bVisible, BOOL bDataMarks, long XAxis, long YAxis, long *lProfileID)`
- `typedef long(XYPLOTFN * xyplot::LPFNDELETEPROFILE)(long lPlotID, long lProfileID)`
- `typedef long xyplot::lProfileID`
- `typedef long const double * xyplot::px`
- `typedef long const double const double * xyplot::py`
- `typedef long const double const double unsigned xyplot::size`
- `typedef long(XYPLOTFN * xyplot::LPFNAPPENDDATA)(long lPlotID, long lProfileID, const double *px, const double *py, unsigned size)`
- `typedef long(XYPLOTFN * xyplot::LPFNSETSTRING)(long lPlotID, long item, long subitem, const char *pszString)`
- `typedef long(XYPLOTFN * xyplot::LPFNGETSTRING)(long lPlotID, long item, long subitem, long *buflen, char *pszBuffer)`
- `typedef long(XYPLOTFN * xyplot::LPFNSETCOLOR)(long lPlotID, long item, long subitem, COLORREF color)`
- `typedef long(XYPLOTFN * xyplot::LPFNGETCOLOR)(long lPlotID, long item, long subitem, COLORREF *pColor)`
- `typedef long(XYPLOTFN * xyplot::LPFNSETLONG)(long lPlotID, long item, long subitem, long value)`
- `typedef long(XYPLOTFN * xyplot::LPFNGETLONG)(long lPlotID, long item, long subitem, long *value)`
- `typedef long(XYPLOTFN * xyplot::LPFNSETDOUBLE)(long lPlotID, long item, long subitem, double value)`
- `typedef long(XYPLOTFN * xyplot::LPFNGETDOUBLE)(long lPlotID, long item, long subitem, double *value)`
- `typedef long(XYPLOTFN * xyplot::LPFNSETFONT)(long lPlotID, long item, long subitem, const LOGFONT *plf)`
- `typedef long(XYPLOTFN * xyplot::LPFNISENABLED)(long lPlotID, long item, long subitem, BOOL *enabled)`
- `typedef long(XYPLOTFN * xyplot::LPFNENABLEITEM)(long lPlotID, long item, long subitem, BOOL enable)`
- `typedef long(XYPLOTFN * xyplot::LPFNSELECT)(long lPlotID, long lProfileID, long index)`
- `typedef long(XYPLOTFN * xyplot::LPFNGETSELECTION)(long lPlotID, long *pID)`
- `typedef long(XYPLOTFN * xyplot::LPFNSAVEPROPERTIES)(long lPlotID, const char *pszFileName)`
- `typedef long(XYPLOTFN * xyplot::LPFNLOADPROPERTIES)(long lPlotID, const char *pszFileName)`
- `typedef long(XYPLOTFN * xyplot::LPFNCOPYTOCLIPBOARD)(long lPlotID, unsigned width, unsigned height, unsigned dpi)`
- `typedef long(XYPLOTFN * xyplot::LPFNDRAWTOBITMAP)(long lPlotID, HBITMAP &hBitmap, bool bDrawWindowBG, bool bDrawClientBG)`
- `typedef long(XYPLOTFN * xyplot::LPFNCREATEGROUP)(long lPlotID, long *plGroupID)`
- `typedef long(XYPLOTFN * xyplot::LPFNDELETEGROUP)(long lPlotID, long lGroupID)`

- typedef long(XYPLOTFN * xyplot::LPFNBINDPROFILE)(long lPlotID, long lGroupID, long lProfileID)
- typedef long(XYPLOTFN * xyplot::LPFNGETPROFLIST)(long lPlotID, long *lSize, long *lProfileIDs)
- typedef long(XYPLOTFN * xyplot::LPFNREGRESTOREDEFAULTS)(long lPlotID)
- typedef long(XYPLOTFN * xyplot::LPFNREGLOADSETTINGS)(long lPlotID, const char *pszKeyName, const char *pszValue)
- typedef long(XYPLOTFN * xyplot::LPFNREGSAVESETTINGS)(long lPlotID, const char *pszKeyName, const char *pszValue)
- typedef long(XYPLOTFN * xyplot::LPFNRUNDIALOG)(long lPlotID)
- typedef long(XYPLOTFN * xyplot::LPFNADDREGION)(long lPlotID, long AxisX, long AxisY, double xFrom, double xTo, double yFrom, double yTo, const char *pszTitle, long *lRegionID)
- typedef long(XYPLOTFN * xyplot::LPFNDELETEREGION)(long lPlotID, long lRegionID)
- typedef long(XYPLOTFN * xyplot::LPFNSETREGION)(long lPlotID, long lRegionID, double x1, double x2, double y1, double y2)
- typedef long(XYPLOTFN * xyplot::LPFNADDLEVEL)(long lPlotID, long axis, double value, const char *pszTitle, COLORREF clr, int width, long *lLevelID)
- typedef long(XYPLOTFN * xyplot::LPFNDELETELEVEL)(long lPlotID, long lLevelID)
- typedef long(XYPLOTFN * xyplot::LPFNSETLEVEL)(long lPlotID, long lLevelID, double value)

Функции

- typedef xyplot::long (XYPLOTFN *LPFNSETDATA)(long lPlotID

12.4.1 Макросы

12.4.1.1 #define ALL_PLOTS ALL_PLOTS_()

12.4.1.2 #define ALL_REGIONS ALL_REGIONS_()

12.4.1.3 #define XYPLOTFN __stdcall