

# **Программирование**

**для системы Lotsia PDM PLUS 5.70**  
**с использованием скриптов и API**

**Люция Софтва**

*Опубликовано  
Лоция Софтвэа  
127422, г. Москва, Тимирязевская ул., д.1, стр. 2.  
Телефон: (495) 748-04-74  
Факс: (495) 748-03-74  
E-mail: sales@lotsia.com  
Интернет: <http://www.lotsia.com>*

*Авторские права  
Авторскими правами обладает ООО «Лоция Софтвэа».*

*Никакая часть данного документа не может быть воспроизведена или передана в любой форме и любыми способами в каких-либо целях без предварительного письменного разрешения ООО «Лоция Софтвэа».*

*Лицензионное соглашение, поставляемое с программным обеспечением, определяет процедуру пользования продуктом.*

*© 1997-2018 ООО «Лоция Софтвэа». С сохранением всех прав.*

*Лоция Софтвэа, Lotsia PDM, Lotsia PDM PLUS, LS Flow являются зарегистрированными торговыми марками ООО «Лоция Софтвэа».*

*Все остальные упомянутые в документе торговые марки являются собственностью их законных владельцев.*

*Информация, содержащаяся в данном документе, может быть изменена без предварительного уведомления.*

*PN: 05-2500-10-RU  
10.05.2018*

## Содержание

<b>1</b>	<b>ОБЩИЕ СВЕДЕНИЯ .....</b>	<b>16</b>
1.1	СКРИПТЫ .....	16
1.2	API.....	16
<b>2</b>	<b>БАЗОВЫЕ ОБЪЕКТЫ И МЕТОДЫ.....</b>	<b>18</b>
2.1	ОБЪЕКТ APPLICATION .....	18
2.1.1	СВОЙСТВА ОБЪЕКТА APPLICATION .....	18
	ActiveDocument.....	18
	ConnectAttrib .....	18
	CurrentUserID .....	18
	Documents.....	18
	Interactive.....	18
	IsConnected.....	18
	Name .....	18
	Visible .....	19
	VersionBuild .....	19
	VersionMajor .....	19
	VersionMinor .....	19
	WsID .....	19
	FilialID .....	19
2.1.2	МЕТОДЫ ОБЪЕКТА APPLICATION .....	19
	Activate .....	19
	Connect .....	19
	GenerateID .....	19
	GetADOConnect .....	20
	GetNextSequence .....	20
	LogMessage .....	20
	GetUserName .....	20
	LogMessage .....	21
	CreateSelectionDialog.....	21
	CreateObjectService.....	21
	CreateModelService.....	21
	CreateReportService.....	21
	NewWindowProjectTree.....	22
	NewWindowProjectHistory .....	22
	NewWindowProjectView.....	22
	NewWindowReport.....	22
	NewWindowObjectCollection .....	23
	NewWindowMail .....	23
	ShowObjectProperties .....	23
	ActivateWindowMailBox .....	23
	OpenDocumentVersion .....	24
	CloseDocumentVersion .....	24
	ExportDocumentVersion.....	24
	CreateOpenDocumentDialog .....	25
	GetOperation .....	25
2.2	ОБЪЕКТ DOCUMENTS.....	25
2.2.1	СВОЙСТВА ОБЪЕКТА DOCUMENTS .....	25
	Count .....	25
2.2.2	МЕТОДЫ ОБЪЕКТА DOCUMENTS .....	25
	Item .....	25

<b>2.3 ОБЪЕКТ DOCUMENT</b> .....	<b>26</b>
2.3.1 СВОЙСТВА ОБЪЕКТА DOCUMENT .....	26
Name.....	26
ClassName .....	26
DataObjects .....	26
Saved .....	26
Title.....	26
2.3.2 МЕТОДЫ ОБЪЕКТА DOCUMENT .....	26
Activate.....	26
Close.....	26
Save .....	26
<b>2.4 ОБЪЕКТ LSDATA</b> .....	<b>27</b>
2.4.1 СВОЙСТВА ОБЪЕКТА LSDATA .....	27
ClassName .....	27
DataObject.....	27
ParentDocument .....	27
2.4.2 ОСНОВНЫЕ МЕТОДЫ ОБЪЕКТА LSDATA .....	27
DeletedCount.....	27
DeleteRow .....	27
Describe.....	27
Filter .....	27
FilteredCount.....	28
Find.....	28
FindGroupChange .....	28
GetItem .....	28
GetItemStrDT .....	29
GetItemDecimal .....	29
GetSelectedRow .....	29
GroupCalc.....	29
InsertRow .....	29
IsSelected.....	30
Modify .....	30
Reset .....	30
RetrieveRest .....	30
RowCount.....	30
RowsDiscard.....	30
SaveAs.....	31
SaveAsAscii .....	31
SetDetailHeight .....	32
SetFilter .....	32
SetItem.....	32
SetItemStrDT .....	32
SetSort .....	33
Sort .....	33
2.4.3 ДОПОЛНИТЕЛЬНЫЕ МЕТОДЫ ОБЪЕКТА LSDATA .....	33
OleExport .....	33
<b>2.5 ОБЪЕКТ LSDATASET</b> .....	<b>34</b>
2.5.1 СВОЙСТВА ОБЪЕКТА LSDATASET .....	34
Count .....	34
2.5.2 МЕТОДЫ ОБЪЕКТА LSDATASET .....	34
Item.....	34
<b>2.6 ОБЪЕКТ LSTRANSACTION</b> .....	<b>34</b>
2.6.1 МЕТОДЫ ОБЪЕКТА LSTRANSACTION.....	35
Execute .....	35
Select .....	35
<b>2.7 ОБЪЕКТ LSQUERY</b> .....	<b>36</b>

2.7.1	СВОЙСТВА ОБЪЕКТА LSQUERY .....	36
	SqlQuery .....	36
2.7.2	МЕТОДЫ ОБЪЕКТА LSQUERY .....	36
	IsCustomSet .....	36
	CustomWhere (формат 1) .....	36
	CustomWhere (формат 2) .....	36
	CustomParam (формат 1) .....	36
	CustomParam (формат 2) .....	37
<b>2.8</b>	<b>ОБЪЕКТ DCONNECTATTRIB.....</b>	<b>37</b>
2.8.1	СВОЙСТВА ОБЪЕКТА DCONNECTATTRIB .....	37
	DBType .....	37
	Driver .....	37
	Server .....	38
	DataBase .....	38
	DBParm .....	38
	User .....	38
2.8.2	МЕТОДЫ ОБЪЕКТА DCONNECTATTRIB .....	38
	Password .....	38
<b>3</b>	<b><u>РЕДАКТИРОВАНИЕ СКРИПТОВ.....</u></b>	<b>39</b>
<b>4</b>	<b><u>СКРИПТЫ В ГЕНЕРАТОРЕ ОТЧЕТОВ .....</u></b>	<b>41</b>
<b>5</b>	<b><u>СКРИПТЫ В ДЕЙСТВИЯХ НАД ОБЪЕКТАМИ .....</u></b>	<b>43</b>
<b>5.1</b>	<b>ОБЪЕКТ LSOPERVARs.....</b>	<b>43</b>
5.1.1	МЕТОДЫ ОБЪЕКТА LSOPERVARs .....	43
	GetVarValue .....	43
	SetVarValue .....	43
<b>6</b>	<b><u>РАБОТА С ДЕЙСТВИЯМИ НАД ОБЪЕКТАМИ .....</u></b>	<b>44</b>
<b>6.1</b>	<b>ОБЪЕКТ DPDMOPERATION .....</b>	<b>44</b>
6.1.1	СВОЙСТВА ОБЪЕКТА DPDMOPERATION .....	44
	ID.....	44
	Description .....	44
	ObjectService .....	44
	UiMode .....	44
	UiClose.....	44
6.1.2	МЕТОДЫ ОБЪЕКТА DPDMOPERATION.....	45
	Execute .....	45
	PrepareVars .....	45
	GetVariables.....	45
	GetVarValue .....	45
	SetVarValue .....	45
	Clear .....	46
<b>6.2</b>	<b>ОБЪЕКТ DPDMOPERATIONVARIABLE.....</b>	<b>46</b>
6.2.1	СВОЙСТВА ОБЪЕКТА DPDMOPERATIONVARIABLE .....	46
	Value .....	46
	Parent.....	46
	Name .....	46
	Description .....	46
	DataType .....	46

EditType .....	47
<b><u>7 СКРИПТЫ В ШАБЛОНАХ РАБОТ.....</u></b>	<b><u>48</u></b>
<b>7.1 ОБЪЕКТ LSJOB.....</b>	<b>48</b>
7.1.1 СВОЙСТВА ОБЪЕКТА LSJOB .....	48
ObjectService .....	48
7.1.2 МЕТОДЫ ОБЪЕКТА LSJOB .....	48
GetVarValue.....	48
SetVarValue .....	48
<b><u>8 СКРИПТЫ В ОКНЕ ПОИСКА ОБЪЕКТОВ.....</u></b>	<b><u>49</u></b>
<b><u>9 РАБОТА С ОБЪЕКТАМИ И ДОКУМЕНТАМИ.....</u></b>	<b><u>51</u></b>
<b>9.1 ОБЪЕКТ DPDMOBJECTSERVICE .....</b>	<b>51</b>
9.1.1 СВОЙСТВА ОБЪЕКТА DPDMOBJECTSERVICE .....	51
CheckRule .....	51
FastMode .....	51
PreventUpdate .....	51
9.1.2 МЕТОДЫ ОБЪЕКТА DPDMOBJECTSERVICE .....	51
Reset .....	51
Update.....	51
GetObject.....	52
CreateObject .....	52
GetLink.....	52
GetDocument.....	53
GetDocumentVersion .....	53
CreateObjectCollection .....	53
GetObjectCollection .....	53
<b>9.2 ОБЪЕКТ DPDMOBJECT .....</b>	<b>54</b>
9.2.1 СВОЙСТВА ОБЪЕКТА DPDMOBJECT .....	54
Name.....	54
Parent.....	54
ID.....	54
TypeID.....	54
Description .....	54
Protected .....	54
Rights.....	54
9.2.2 МЕТОДЫ ОБЪЕКТА DPDMOBJECT .....	54
GetNameForLink.....	54
GetAttrib.....	54
GetAttribs .....	55
AddAttrib.....	55
SetAttrib .....	55
DeleteAttrib .....	56
CreateOption.....	56
GetLinkedObjects .....	56
AddLink.....	57
DeleteLink .....	57
GetDocument.....	57
GetDocuments .....	57
ImportDocument.....	58
DeleteDocument .....	58

<b>9.3 DPDMATTRIB.....</b>	<b>59</b>
9.3.1 СВОЙСТВА ОБЪЕКТА DPDMATTRIB .....	59
Value .....	59
Parent.....	59
AttribID .....	59
<b>9.4 DPDMLINK .....</b>	<b>59</b>
9.4.1 СВОЙСТВА ОБЪЕКТА DPDMLINK .....	59
Parent.....	59
ID.....	59
ParentObject .....	59
ChildObject .....	59
LinkTypeID.....	59
LinkType .....	59
9.4.2 МЕТОДЫ ОБЪЕКТА DPDMLINK .....	60
GetAttrib .....	60
GetAttribs .....	60
AddAttrib .....	60
SetAttrib .....	60
DeleteAttrib.....	61
AddVariant .....	61
SetVariant.....	62
<b>9.5 DPDMDOCUMENT.....</b>	<b>62</b>
9.5.1 СВОЙСТВА ОБЪЕКТА DPDMDOCUMENT .....	62
Parent.....	62
ID.....	62
Description .....	62
TypeID .....	62
AppID.....	62
StorageID .....	62
Note .....	62
Rights .....	62
ActualVersion .....	62
9.5.2 МЕТОДЫ ОБЪЕКТА DPDMDOCUMENT.....	63
GetVersion .....	63
GetVersions.....	63
CreateVersion.....	63
DeleteVersion.....	63
<b>9.6 DPDMDOCUMENTVERSION.....</b>	<b>64</b>
9.6.1 СВОЙСТВА ОБЪЕКТА DPDMDOCUMENTVERSION.....	64
Parent.....	64
ID.....	64
Label.....	64
Note .....	64
IsActual .....	64
Rights .....	64
<b>9.7 ОБЪЕКТ DOPENDOCUMENTDIALOG .....</b>	<b>64</b>
9.7.1 СВОЙСТВА ОБЪЕКТА DOPENDOCUMENTDIALOG.....	64
MethodID .....	64
DocumentVersionID .....	64
9.7.2 МЕТОДЫ ОБЪЕКТА DOPENDOCUMENTDIALOG .....	64
OpenDocument.....	64
OpenDocumentVersion.....	65
<b>9.8 ОБЪЕКТ DPDMOBJECTCOLLECTION .....</b>	<b>65</b>
9.8.1 СВОЙСТВА ОБЪЕКТА DPDMOBJECTCOLLECTION .....	66
ID.....	66
Description .....	66

IsPersonal .....	66
Rights.....	66
9.8.2 МЕТОДЫ ОБЪЕКТА DPDMOBJECTCOLLECTION.....	66
GetObjectsID .....	66
GetObjects .....	66
AddObjects .....	66
DeleteObjects .....	67
Delete.....	67

## **10 РАБОТА С ПРАВАМИ ПОЛЬЗОВАТЕЛЯ ..... 68**

### **10.1 DOBJECTRIGHTS ..... 68**

10.1.1 СВОЙСТВА ОБЪЕКТА DOBJECTRIGHTS.....	68
Parent.....	68
10.1.2 МЕТОДЫ ОБЪЕКТА DOBJECTRIGHTS.....	68
Items .....	68
Add .....	68
Delete.....	69
GetEffective.....	69

### **10.2 DRIGHTENTRY ..... 69**

10.2.1 СВОЙСТВА ОБЪЕКТА DRIGHTENTRY.....	69
Parent.....	69
Grant .....	69
ID .....	70
UserID .....	70
RightID .....	70
Inherit .....	70
InheritLevel .....	70
ParentID.....	70
GrantorID .....	70
CD.....	70
WsID.....	70

### **10.3 ПЕРЕЧЕНЬ ПРАВ ПОЛЬЗОВАТЕЛЯ ..... 71**

## **11 РАБОТА С ОКНАМИ ВЫБОРА ..... 72**

### **11.1 ОБЪЕКТ DEMPLOYEESELECTION ..... 72**

11.1.1 СВОЙСТВА ОБЪЕКТА DEMPLOYEESELECTION .....	72
Title.....	72
11.1.2 МЕТОДЫ ОБЪЕКТА DEMPLOYEESELECTION.....	72
ID.....	72
Select .....	72

### **11.2 ОБЪЕКТ DOBJECTSELECTION..... 72**

11.2.1 СВОЙСТВА ОБЪЕКТА DOBJECTSELECTION.....	73
Title.....	73
TreeLinkId .....	73
11.2.2 МЕТОДЫ ОБЪЕКТА DOBJECTSELECTION .....	73
ID.....	73
Select .....	73
SelectTree.....	73

### **11.3 ОБЪЕКТ DOPERATIONSELECTION ..... 74**

11.3.1 СВОЙСТВА ОБЪЕКТА DOPERATIONSELECTION .....	74
Title.....	74
11.3.2 МЕТОДЫ ОБЪЕКТА DOPERATIONSELECTION .....	74



ID.....	74
Select.....	74
<b>11.4 ОБЪЕКТ DATTRIBSELECTION .....</b>	<b>75</b>
11.4.1 СВОЙСТВА ОБЪЕКТА DATTRIBSELECTION.....	75
Title .....	75
11.4.2 МЕТОДЫ ОБЪЕКТА DATTRIBSELECTION .....	75
ID.....	75
Select.....	75
<b>11.5 ОБЪЕКТ DOBJECTTYPESELECTION .....</b>	<b>75</b>
11.5.1 СВОЙСТВА ОБЪЕКТА DOBJECTTYPESELECTION.....	75
Title .....	75
11.5.2 МЕТОДЫ ОБЪЕКТА DOBJECTTYPESELECTION .....	76
ID.....	76
Select.....	76
<b>11.6 ОБЪЕКТ DLINKTYPESELECTION .....</b>	<b>76</b>
11.6.1 СВОЙСТВА ОБЪЕКТА DLINKTYPESELECTION .....	76
Title .....	76
11.6.2 МЕТОДЫ ОБЪЕКТА DLINKTYPESELECTION .....	76
ID.....	76
Select.....	77
<b>11.7 ОБЪЕКТ DDOCUMENTSELECTION .....</b>	<b>77</b>
11.7.1 СВОЙСТВА ОБЪЕКТА DDOCUMENTSELECTION.....	77
Title .....	77
11.7.2 МЕТОДЫ ОБЪЕКТА DDOCUMENTSELECTION .....	77
ID.....	77
SetObject.....	77
Select.....	78
<b>11.8 ОБЪЕКТ DDOCUMENTVERSIONSELECTION .....</b>	<b>78</b>
11.8.1 СВОЙСТВА ОБЪЕКТА DDOCUMENTVERSIONSELECTION .....	78
Title .....	78
11.8.2 МЕТОДЫ ОБЪЕКТА DDOCUMENTVERSIONSELECTION .....	78
ID.....	78
SetObject.....	78
Select.....	79
<b>11.9 ОБЪЕКТ DAPPLICATIONSELECTION .....</b>	<b>79</b>
11.9.1 СВОЙСТВА ОБЪЕКТА DAPPLICATIONSELECTION .....	79
Title .....	79
11.9.2 МЕТОДЫ ОБЪЕКТА DAPPLICATIONSELECTION .....	79
ID.....	79
Select.....	79
<b>11.10 ОБЪЕКТ DSTORAGESELECTION.....</b>	<b>80</b>
11.10.1 СВОЙСТВА ОБЪЕКТА DSTORAGESELECTION .....	80
Title .....	80
11.10.2 МЕТОДЫ ОБЪЕКТА DSTORAGESELECTION.....	80
ID.....	80
Select.....	80
<b>11.11 ОБЪЕКТ DDOCUMENTTYPESELECTION .....</b>	<b>81</b>
11.11.1 СВОЙСТВА ОБЪЕКТА DDOCUMENTTYPESELECTION .....	81
Title .....	81
11.11.2 МЕТОДЫ ОБЪЕКТА DDOCUMENTTYPESELECTION.....	81
ID.....	81
Select.....	81
<b>11.12 ОБЪЕКТ DREPORTSELECTION .....</b>	<b>81</b>
11.12.1 МЕТОДЫ ОБЪЕКТА DREPORTSELECTION .....	82
ID.....	82
Select.....	82

<b>11.13</b>	<b>ОБЪЕКТ DOBJECTCOLLECTIONSELECTION .....</b>	<b>82</b>
11.13.1	СВОЙСТВА ОБЪЕКТА DOBJECTCOLLECTIONSELECTION.....	82
	Title.....	82
11.13.2	МЕТОДЫ ОБЪЕКТА DOBJECTCOLLECTIONSELECTION .....	82
	ID.....	82
	Select .....	83

## **12 РАБОТА СО СТРУКТУРОЙ ДАННЫХ..... 84**

<b>12.1</b>	<b>ОБЪЕКТ DSTRMODELSERVICE .....</b>	<b>84</b>
12.1.1	МЕТОДЫ ОБЪЕКТА DSTRMODELSERVICE .....	84
	Reset .....	84
	Update.....	84
	GetLinkType.....	84
	GetAttribGroup.....	84
	GetAttrib.....	84
	CreateAttrib .....	85
	GetObjectType .....	85
	CreateObjectType .....	85
	GetDocumentType .....	86
	GetStorage .....	86
	GetApplication .....	86
<b>12.2</b>	<b>ОБЪЕКТ DSTRLINKTYPE .....</b>	<b>86</b>
12.2.1	СВОЙСТВА ОБЪЕКТА DSTRLINKTYPE .....	87
	Description .....	87
	Parent.....	87
	ID.....	87
	Type.....	87
<b>12.3</b>	<b>ОБЪЕКТ DSTRATTRIBGROUP.....</b>	<b>87</b>
12.3.1	СВОЙСТВА ОБЪЕКТА DSTRATTRIBGROUP .....	87
	Description .....	87
	Parent.....	87
	ID.....	87
<b>12.4</b>	<b>ОБЪЕКТ DSTRATTRIB .....</b>	<b>87</b>
12.4.1	СВОЙСТВА ОБЪЕКТА DSTRATTRIB.....	87
	Description .....	87
	Parent.....	87
	ID.....	87
	DataType .....	88
	MultiValue.....	88
	ForProject .....	88
	Group.....	88
	EditMask.....	88
	EditType .....	88
	History .....	88
<b>12.5</b>	<b>ОБЪЕКТ DSTROBJECTTYPE .....</b>	<b>88</b>
12.5.1	СВОЙСТВА ОБЪЕКТА DSTROBJECTTYPE.....	88
	Description .....	88
	Parent.....	88
	ID.....	88
	Class .....	89
	IconPath.....	89
	SortKey.....	89
	Mnemo.....	89
12.5.2	МЕТОДЫ ОБЪЕКТА DSTROBJECTTYPE .....	89

GetAttribs .....	89
AddAttrib .....	89
GetObjectTypes.....	89
AddObjectType .....	90
<b>12.6 ОБЪЕКТ DSTROBJECTTYPEATTRIB .....</b>	<b>90</b>
12.6.1 СВОЙСТВА ОБЪЕКТА DSTROBJECTTYPEATTRIB .....	90
Parent.....	90
Attrib .....	90
Required .....	90
Value .....	90
<b>12.7 ОБЪЕКТ DSTROBJECTTYPEOBJECTTYPE .....</b>	<b>90</b>
12.7.1 СВОЙСТВА ОБЪЕКТА DSTROBJECTTYPEOBJECTTYPE.....	90
Parent.....	90
ObjectType .....	90
LinkType .....	91
Required .....	91
<b>12.8 ОБЪЕКТ DSTRDOCUMENTTYPE.....</b>	<b>91</b>
12.8.1 СВОЙСТВА ОБЪЕКТА DSTRDOCUMENTTYPE .....	91
Parent.....	91
ID.....	91
Description .....	91
Note .....	91
MaxVer .....	91
<b>12.9 ОБЪЕКТ DSTRSTORAGE.....</b>	<b>91</b>
12.9.1 СВОЙСТВА ОБЪЕКТА DSTRSTORAGE .....	91
Parent.....	91
ID.....	91
Description .....	91
Type.....	91
Path.....	91
IsDefault.....	92
AccessType .....	92
VolumeLabel.....	92
IsDeleted .....	92
<b>12.10 ОБЪЕКТ DSTRAPPLICATION .....</b>	<b>92</b>
12.10.1 СВОЙСТВА ОБЪЕКТА DSTRAPPLICATION.....	92
Parent.....	92
ID.....	92
Description .....	92
AppID.....	92
Extension.....	92
ExtensionFilter .....	92
Icon.....	92
DefaultDocumentType .....	92
IsDeleted .....	92
12.10.2 МЕТОДЫ ОБЪЕКТА DSTRAPPLICATION .....	93
GetMethods .....	93
<b>12.11 ОБЪЕКТ DSTRAPPLICATIONMETHOD.....</b>	<b>93</b>
12.11.1 СВОЙСТВА ОБЪЕКТА DSTRAPPLICATIONMETHOD .....	93
Parent.....	93
ID.....	93
Description .....	93
Type.....	93
Parameters .....	93
Path.....	93
UseRegPath .....	93

RegValue .....	93
IsDefault .....	93
Integration.....	94
UseCheckOut.....	94
DeleteLocalFile .....	94
UseDDE.....	94
DDEOpenApp .....	94
DDEOpenTopic .....	94
DDEOpenRun.....	94
DDEOpenNotRun.....	94

## **13 РАБОТА С ОТЧЕТАМИ..... 95**

### **13.1 ОБЪЕКТ DREPORTOBJECTSERVICE ..... 95**

13.1.1 МЕТОДЫ ОБЪЕКТА DREPORTOBJECTSERVICE .....	95
GetReport .....	95
GetReports .....	95

### **13.2 ОБЪЕКТ DREPORTOBJECT ..... 95**

13.2.1 СВОЙСТВА ОБЪЕКТА DREPORTOBJECT .....	95
Parent.....	95
ID.....	95
ParentReport .....	95
Description .....	95
Type.....	95
LevelType.....	96
SouRoot .....	96
DataToReport .....	96
DataToReportFromColumn.....	96
13.2.2 МЕТОДЫ ОБЪЕКТА DREPORTOBJECT .....	96
Execute .....	96
GetColumns.....	97
GetArguments.....	97

### **13.3 ОБЪЕКТ DREPORTCOLUMN ..... 97**

13.3.1 СВОЙСТВА ОБЪЕКТА DREPORTCOLUMN.....	97
Name.....	97
Type.....	97
Description .....	97
ColType .....	97
Band.....	97

### **13.4 ОБЪЕКТ DREPORTARGUMENT..... 98**

13.4.1 СВОЙСТВА ОБЪЕКТА DREPORTARGUMENT .....	98
Name.....	98
Description .....	98
Type.....	98
Format.....	98
Initial.....	98

## **14 ПОДКЛЮЧАЕМЫЕ МОДУЛИ ..... 99**

### **14.1 ОБЪЕКТ DPLUGINMODULE ..... 99**

14.1.1 МЕТОДЫ ОБЪЕКТА DPLUGINMODULE.....	99
GetPluginInfo .....	99
Run .....	100
RunObject.....	100

EditValue.....	100
<b>14.2 ОБЪЕКТ DPLUGININFO .....</b>	<b>100</b>
14.2.1 МЕТОДЫ ОБЪЕКТА DPLUGININFO .....	100
SetInfo .....	100
AddFunctionRun .....	100
AddFunctionRunObject .....	101
AddFunctionEditValue.....	101
<b>14.3 ОБЪЕКТ DPLUGINRUNOBJECTPARM .....</b>	<b>101</b>
14.3.1 СВОЙСТВА ОБЪЕКТА DPLUGINRUNOBJECTPARM.....	101
LinkTypeID.....	101
Hwnd .....	101
Count .....	101
14.3.2 МЕТОДЫ ОБЪЕКТА DPLUGINRUNOBJECTPARM .....	102
ObjectID .....	102
TLinkID .....	102
<b>14.4 ОБЪЕКТ DPLUGINEDITVALUEPARM .....</b>	<b>102</b>
14.4.1 СВОЙСТВА ОБЪЕКТА DPLUGINEDITVALUEPARM.....	102
Value .....	102
AttribID .....	102
ObjectTypeID.....	102
ObjectID .....	102
TLinkID .....	103
Hwnd .....	103
14.4.2 МЕТОДЫ ОБЪЕКТА DPLUGINEDITVALUEPARM .....	103
SetEditType.....	103
AddValue .....	103
 <b>15 ПРИМЕРЫ СКРИПТОВ.....</b>	 <b>104</b>
 15.1 ОБРАЩЕНИЕ К БАЗЕ ДАННЫХ: ВЫПОЛНЕНИЕ SQL-ЗАПРОСА.....	 104
15.2 ВЫГРУЗКА ОТЧЕТОВ В ПРИЛОЖЕНИЯ MS WORD И MS EXCEL .....	106
15.3 ЭКСПОРТ ОТЧЕТА ПО ШАБЛОНУ .....	109
15.4 СЦЕПЛЕНИЕ СТРОК ЧЕРЕЗ УКАЗАННЫЙ РАЗДЕЛИТЕЛЬ .....	110
15.5 ПОЛУЧЕНИЕ ДАННЫХ ИЗ ДРУГОЙ СУБД В КАРТАХ РАБОТ .....	110
15.6 НАСЛЕДОВАНИЕ ЗНАЧЕНИЯ АТТРИБУТОВ В ДЕРЕВЕ ПРОЕКТА. ....	112
15.7 РАБОТА СО ЗНАЧЕНИЯМИ МНОЖЕСТВЕННОГО АТТРИБУТА.....	113
15.8 ОБРАЩЕНИЕ К API ИЗ ВНЕШНЕГО ПРИЛОЖЕНИЯ. ....	114

## **Отказ от ответственности**

Настоящее описание поставляется «КАК ЕСТЬ», без каких-либо других гарантий или условий, прямых или подразумеваемых.

ООО «Лоция Софтвэз», не несёт никакой ответственности перед вами или любым другим юридическим или физическим лицом за любой возникший вследствие использования изложенной в данном описании информации косвенный, случайный, намеренный или другой ущерб, включая, помимо прочего, недополученную прибыль или доход, потерю или повреждение данных, а также любой другой денежный или экономический ущерб, даже если мы были заранее предупреждены о возможности такого ущерба или его можно было предвидеть. Мы также не несем ответственности по претензиям со стороны третьих лиц. Ограничения относятся к вам вне зависимости от того, является ли заявленное вами нарушение нарушением существенного положения или условия, или само по себе является существенным нарушением. Некоторые государства/страны не допускают исключений или ограничений ответственности за косвенный или случайный ущерб, и в силу этого вышеуказанные ограничения могут не применяться к вам.

ООО «Лоция Софтвэз» оставляет за собой право внесения изменения в настоящее описание в любое время.



# 1 Общие сведения

Для расширения возможностей в системе Lotsia PDM PLUS существует встроенный механизм выполнения программ, написанных с использованием скриптовых языков (VBScript, JScript). Для взаимодействия внешних систем с системой Lotsia PDM PLUS может быть использован API, основанный на COM-технологии.

Скрипты и API позволяют получить доступ к одинаковому набору базовых сервисов, через которые осуществляется взаимодействие с системой Lotsia PDM PLUS. Через эти сервисы можно получить информацию об объектах, их атрибутах, связях с другими объектами и документам архива. Так же они позволяют создавать новые и изменять существующие объекты и документы архива. В зависимости от контекста выполнения скрипты могут получить доступ к дополнительным сервисам. Подробное описание всех возможностей работы со скриптами и API приведено в этом руководстве.

Использование скриптов и API не требует дополнительных настроек и может быть начато сразу после установки системы Lotsia PDM PLUS.

## 1.1 Скрипты

Скрипты хранятся в отдельных модулях в базе данных. Модули могут быть двух типов «Скрипт» и «Модуль». Модули типа «Скрипт» всегда привязаны к определенному объекту системы, например, к отчету. Модули типа «Модуль» являются свободными. Они могут быть использованы в одном или нескольких других модулях. Для каждого модуля обязательно указывается язык, на котором написан скрипт и имя модуля, которое должно начинаться с символа латинского алфавита и может содержать латинские буквы и цифры. Модуль типа «Скрипт» всегда имеет имя «Global». Имена свободных модулей могут совпадать, но среди всех связанных модулей совпадение имени не допускается. Например, в базе данных могут содержаться сразу два модуля с именем «Mod1», но они не могут быть одновременно связаны с другим модулем. Модули с разными языками не могут быть использованы одновременно. В текущей версии программы поддерживаются языки VBScript и JScript.

Управление скриптами доступно администратору программы и пользователям, имеющим административную привилегию «Управление скриптами».

## 1.2 API

Для взаимодействия внешних приложений с системой Lotsia PDM PLUS можно воспользоваться API. API выполнен на базе COM-технологии. Это дает простую возможность работы из многих распространенных сред разработки, а также из многих языков, встроенных в другие приложения (например, Visual Basic for Applications в Microsoft Office).

Работа с API начинается с создания объекта Application, который позволяет получить доступ ко всем возможностям API. Для создания этого объекта необходимо использовать программный идентификатор 'LotsiaPDM.Application'. Если система Lotsia PDM PLUS запущена, то к ней будет произведено подключение для создания объекта Application. В противном случае будет запущена система Lotsia PDM PLUS в скрытом (т.е. с невидимым окном приложения) режиме. При таком запуске подключение к БД не производится, вызывающая система должна самостоятельно подключить систему к БД при помощи метода Connect объекта Application. В этом случае перед подключением к БД можно изменить настройки подключения к БД программно или дать такую возможность пользователю. Если параметры подключения к БД не меняются, то будут использованы текущие параметры. До выполнения подключения к БД никакие сервисы системы Lotsia PDM PLUS недоступны.



Одна запущенная копия системы Lotsia PDM PLUS может обслуживать сразу несколько подключений внешних систем. Однако в каждый момент времени может выполняться только одно обращение одного из подключений к свойству или методу объекта API. При одновременном обращении от других подключений, они будут ожидать окончания выполнения обращения первой системы. Если при вызове какого-либо метода объекта на экране будет показано окно, ожидающее действий пользователя, то в это время могут быть обработаны обращения от других систем.

Если при обращении к API на экран выводится диалоговое окно, то в некоторых случаях оно будет размещено на заднем плане, под окном активного приложения. Для того чтобы окно отображалось на переднем плане и могло получить фокус ввода, необходимо использовать системную функцию Windows `CoAllowSetForegroundWindow`. За подробностями следует обратиться к MSDN.

При необходимости запустить отдельную копию системы Lotsia PDM PLUS и выполнить подключение к БД с нужными параметрами, следует создать объект `ApplicationFactory`, используя программный идентификатор `'LotsiaPDM.ApplicationFactory'`. После создания объекта его свойство `Application` будет содержать объект `Application`. Для продолжения работы необходимо выполнить подключение к БД. Система Lotsia PDM PLUS, запущенная таким способом, может обрабатывать только одно подключение API.

По окончании использования программами API система Lotsia PDM PLUS будет автоматически закрыта, если она не находится в интерактивном (видимом) режиме.

Сервер автоматических этапов и сервер поиска не обрабатывают обращений к API.

## 2 Базовые объекты и методы

При выполнении любого скрипта всегда существует глобальный объект «Application». Этот объект содержит свойства и методы, позволяющие взаимодействовать с приложением. В зависимости от контекста выполнения скрипта, могут быть доступны различные дополнительные глобальные объекты, предоставляющие дополнительные возможности взаимодействия с приложением.

Каждый документ, загруженный в отдельном окне, представлен в скриптах как отдельный объект типа Document. Через этот объект скрипт может взаимодействовать как с содержимым документа, так и с окном, в котором этот документ загружен.

Объект типа Documents содержит список открытых документов в виде объектов типа Document.

Объект типа LsData представляет набор данных и описание их визуального представления. Объекты этого типа представляют различные элементы приложения, они могут быть как визуальными, так и невизуальными. Обычно этими элементами являются отчеты и формы. Содержимое документа может быть представлено одним или несколькими подобными элементами. В некоторых случаях в документе могут отсутствовать элементы данных.

При помощи объектов типа LsDataSet и LsDataRecord производится обмен сложными данными (структурами) между скриптом и приложением.

Ниже приводится описание базовых объектов.

### 2.1 Объект Application

Объект Application является объектом верхнего уровня. Он всегда доступен при выполнении скрипта. Объект предоставляет основные методы и свойства для взаимодействия с приложением.

#### 2.1.1 Свойства объекта Application

##### ActiveDocument

Объект. Активный документ. Если нет, то содержит пустое значение (Empty). Только для чтения.

##### ConnectAttrib

DConnectAttrib. Объект, содержащий текущие свойства подключения к БД.

##### CurrentUserID

Число. ID текущего пользователя. Только для чтения.

##### Documents

Объект. Коллекция объектов для каждого открытого документа. Только для чтения.

##### Interactive

Содержит истину, если пользователь может работать с приложением.

##### IsConnected

Логический. Содержит истину, если система подключена к БД.

##### Name

Строка. Имя приложения. Это свойство используется по умолчанию. Только для чтения.

## Visible

Содержит истину, если основное окно приложения видимо.

## VersionBuild

Число. Номер сборки. Только для чтения.

## VersionMajor

Число. Старший номер версии. Только для чтения.

## VersionMinor

Число. Младший номер версии. Только для чтения.

## WsID

Число. Номер рабочего места. Только для чтения.

## FilialID

Число. Номер филиала. Только для чтения.

## 2.1.2 Методы объекта Application

### Activate

Активизирует основное окно приложения.

#### Замечания

Если основное окно системы Lotsia PDM PLUS невидимо, то предварительно оно делается видимым. Если окно свернуто, то предварительно оно восстанавливается.

### Connect

Выполняет подключение к БД.

#### Возврат

Логическое. Истина, если система успешно подключена к БД. Ложь – пользователь отменил подключение к БД (нажал кнопку «Отмена» в диалоговом окне подключения к БД).

#### Замечания

Если система Lotsia PDM PLUS еще не была подключена к БД, то производится попытка подключения. В зависимости от установленных параметров подключения на экране может быть выдано диалоговое окно подключения к БД (см. описание объекта DConnectAttrib). В этом случае при возникновении ошибки подключения к БД пользователю будет выдано сообщение об ошибке, и он сможет ввести имя и пароль еще раз. Если система уже была ранее подключена к БД, то никаких действий не производится.

После неудачного подключения объект Application больше не может быть использован и должен быть пересоздан.

### GenerateID

Возвращает новое числовое ID, которое можно использовать для вставки данных в таблицу. Используется при прямом обновлении БД.

#### Аргументы

*TableName*

Строка. Имя таблицы. Допустимые значения:

object\_reference

attrib\_value

tree\_link

#### Возврат

Число. Новое значение ID.

## GetADOConnect

Создает новое подключение к текущей БД под текущим пользователем и возвращает объект типа «ADODB.Connection»

### Аргументы

#### *Params*

Строка. Не обязательный. Дополнительные параметры. Допустимые значения:

OraOLEDB=Yes      Использовать для подключения к БД Oracle поставщик Oracle OraOLEDB.

Требуется версия 5.50 сборка 140515.

### Возврат

Объект. Новое подключение к БД.

### Замечания

В предыдущих версиях этот метод можно было вызвать только через глобальную функцию LsGetADOConnect.

### Версия

См. описание метода.

## GetNextSequence

Возвращает следующее значение пользовательского счетчика.

### Аргументы

#### *Name*

Строка. Имя пользовательского счетчика. Максимальная длина 40 символов, регистр не учитывается.

#### *Description*

Строка. Описание для создаваемого счетчика. Используется, если происходит создание нового счетчика. Если не указано, то используется значение аргумента Name.

### Возврат

Decimal. Следующее значение пользовательского счетчика.

### Замечания

Если на момент вызова метода пользовательского счетчика с указанным именем не существует, то он будет создан. При создании счетчика ему устанавливается описание из аргумента Description. Для нового счетчика метод возвращает значение 1.

### Версия

5.70 сборка 180510.

## LogMessage

Записывает в журнал приложения (log-файл) строку.

### Аргументы

#### *Message*

Строка. Сообщение для записи в журнал.

## GetUserName

Возвращает имя указанного пользователя или группы.

### Аргументы

#### *UserID*

Число. ID пользователя или группы.

### Возврат

Строка. Имя пользователя или группы. Если пользователь или группа не найдены, то возвращается пустая строка.

## Замечания

В предыдущих версиях этот метод можно было вызвать только через глобальную функцию LsGetUserName.

## LogMessage

Записывает в журнал приложения (log-файл) строку.

### Аргументы

*Message*

Строка. Сообщение для записи в журнал.

## CreateSelectionDialog

Создает новый объект для работы пользователя с окном выбора указанного типа.

### Аргументы

*Type*

Строка. Тип окна выбора. Может содержать одно из приведенных значений (без учета регистра):

Тип	Создаваемый объект	Описание
Employee	DEmployeeSelection	Выбор пользователя или группы.
Object	DObjectSelection	Выбор объектов.
Operation	DOperationSelection	Выбор действий над объектом.
Attrib	DAttribSelection	Выбор атрибутов.
ObjectType	DObjectTypeSelection	Выбор типов объекта.
LinkType	DLinkTypeSelection	Выбор типов связи.
Document	DDocumentSelection	Выбор документов архива.
DocumentVersion	DDocumentVersionSelection	Выбор версий документов архива.
Application	DApplicationSelection	Выбор приложений.
Storage	DStorageSelection	Выбор хранилищ документов.
DocumentType	DDocumentTypeSelection	Выбор типов документов.
Report	DReportSelection	Выбор отчетов.
ObjectCollection	DObjectCollectionSelection	Выбор подборки объектов

### Возврат

Созданный объект. Тип объекта зависит от типа окна выбора.

## CreateObjectService

Создает новый сервис обработки объектов Lotsia PDM PLUS и документов архива.

### Возврат

DPDMObjectService. Созданный сервис обработки.

## CreateModelService

Создает новый сервис обработки структуры данных Lotsia PDM PLUS.

### Возврат

DSTRModelService. Созданный сервис обработки.

## CreateReportService

Создает новый сервис обработки отчетов.

### Аргументы

*Type*

Строка. Тип сервиса обработки отчетов. Может содержать одно из значений (без учета регистра):

Значение	Тип объекта	Описание
----------	-------------	----------

Object      DReportObjectService      Пользовательские отчеты по объектам.

### Возврат

Созданный сервис обработки отчетов. Тип объекта зависит от значения аргумента Type.

## NewWindowProjectTree

Открывает в приложении Lotsia PDM PLUS новое окно дерева проекта и загружает в него указанный объект.

### Аргументы

*ObjectID*

Decimal. ID корневого объекта. Если содержит 0, то пользователю будет предложено выбрать объект.

*LinkTypeID*

Long. ID типа связи. Если содержит 0, то пользователю будет предложено выбрать тип связи.

## NewWindowProjectHistory

Открывает в приложении Lotsia PDM PLUS новое окно истории проекта и загружает в него указанный объект.

### Аргументы

*ObjectID*

Decimal. ID корневого объекта. Если содержит 0, то пользователю будет предложено выбрать объект.

*LinkTypeID*

Long. ID типа связи. Если содержит 0, то пользователю будет предложено выбрать тип связи.

*History*

Дата. Содержит дату и время состояния открываемого проекта. Если содержит значение Null, то пользователю будет предложено выбрать значение.

## NewWindowProjectView

Открывает в приложении Lotsia PDM PLUS новое окно представление дерева проектов и загружает в него указанный объект.

### Аргументы

*ObjectID*

Decimal. ID корневого объекта. Если содержит 0, то пользователю будет предложено выбрать объект.

*ViewID*

Decimal. ID представления проекта. Если содержит 0, то пользователю будет предложено выбрать представление проекта.

## NewWindowReport

Открывает в приложении Lotsia PDM PLUS новое окно для выполнения пользовательского отчета.

### Аргументы

*Type*

Строка. Тип отчета. может содержать одно из значений (без учета регистра):

Значение	Описание
----------	----------

Object	Пользовательские отчеты по объектам
--------	-------------------------------------

*ReportID*

Decimal. ID отчета.

*ReportParm*

Строка. Не обязательный. Дополнительные параметры запуска отчета.

*AddSqlWhere*

Строка. Не обязательный. Дополнительные sql-условия отбора.

*ObjectID*

Decimal или массив Decimal. Не обязательный. ID объектов, передаваемых в отчет в качестве аргумента.

#### *Arg*

Не обязательный, любое количество. Аргументы отчета в порядке их описания в отчете.

## **NewWindowObjectCollection**

Открывает в приложении Lotsia PDM PLUS новое окно подборки объектов.

### **Аргументы**

#### *ID*

Decimal. Не обязательный. ID подборки объектов. Если аргумент не указан, то открывается окно с новой подборкой.

### **Версия**

5.60 сборка 150330.

## **NewWindowMail**

Открывает в приложении Lotsia PDM PLUS новое окно с указанным сообщением.

### **Аргументы**

#### *MailType*

Число. Тип сообщения. Может содержать одно из значений:

Значение	Описание
0	Входящее сообщение.
1	Исходящее сообщение.

#### *MailID*

Decimal. ID сообщения.

### **Замечания**

Если окно с указанным сообщением уже открыто, то оно будет активизировано.

### **Версия**

5.10 сборка 120618.

## **ShowObjectProperties**

Открывает стандартное окно свойств объекта.

### **Аргументы**

#### *ObjectID*

Decimal. ID объекта.

#### *LinkID*

Decimal. ID связи объекта с родительским объектом. Используется для доступа к атрибутам проекта. Если аргумент содержит 0, то свойства объекта отображаются без учета связи (без атрибутов проекта).

## **ActivateWindowMailBox**

Открывает в приложении Lotsia PDM PLUS окно «Задания и сообщения» и, если необходимо, выделяет в нем указанное сообщение.

### **Аргументы**

#### *FolderID*

Число. Не обязательный. ID почтовой папки, которую необходимо выделить. Если папку не надо выделять, то должен содежать 0. Значение параметра игнорируется, если аргумент MailID присутствует и содержит ненулевое значение.

#### *MailType*

Число. Не обязательный. Тип сообщения. Может содержать одно из значений:

Значение	Описание
----------	----------

- 0 Входящее сообщение.
- 1 Исходящее сообщение.

Значение параметра игнорируется, если аргумент MailID содержит нулевое значение.

#### **MailID**

Decimal. Не обязательный. ID сообщения. Содержит ID сообщения, которое необходимо сделать текущим в окне «Задания и сообщения». Если это не требуется, то должен содержать нулевой ID (0).

#### **UseVirtualFolder**

Boolean. Не обязательный. Если указан и содержит истину, то по возможности выделять виртуальную папку, в которой указанное в аргументе MailID сообщение может быть отображено.

#### **Refresh**

Boolean. Не обязательный. Если указан и содержит истину, то содержимое окна «Задания и сообщения» обновляется. Используется, если окно «Задание и сообщения уже открыто и не производится смена папки».

### **Замечания**

Если окно «Задания и сообщения» уже открыто, то оно будет активизировано.

### **Версия**

5.10 сборка 120618.

## **OpenDocumentVersion**

Открывает версию документ архива для доступа к ее содержимому.

### **Аргументы**

#### **DocumentVersionID**

Decimal. ID версии документа архива.

#### **ReadOnly**

Boolean. Не обязательный. Если указан и содержит истину, то версия документа архива открывается в режиме просмотра.

### **Возврат**

Строка. Полное имя файла для доступа к содержимому версии документа архива.

### **Замечания**

Версия документа архива помечается как используемая текущим пользователем. Если соответствующий документ находится в защищенном хранилище, то к файлу открывается доступ в соответствии со значением аргумента ReadOnly. Если у пользователя нет достаточных прав, то возникает ошибка. После окончания использования версии документа она должна быть закрыта вызовом метода CloseDocumentVersion. Количество открытых и закрытых версии документа архива должно совпадать.

## **CloseDocumentVersion**

Закрывает версию документ архива, открытую ранее вызовом метода OpenDocumentVersion.

### **Аргументы**

#### **DocumentVersionID**

Decimal. ID версии документа архива.

## **ExportDocumentVersion**

Экспортирует версию документа архива.

### **Аргументы**

#### **DocumentVersionID**

Decimal. ID версии документа архива.

#### **Path**

Строка. Не обязательный. Может содержать папку, в которую необходимо поместить файл с исходным именем с содержимым документа, или полное имя файла, в который будет записано содержимое документа. Если не указан или содержит пустую строку, то во временной папке (TEMP) создается подпапка с уникальным именем, в которую экспортируется файл. Если содержит строку,



оканчивающуюся символом '\', строка воспринимается как имя папки для экспорта. Иначе содержимое строки воспринимается как полное имя файла, в который будет записано содержимое документа. При экспорте содержимое существующего файла переписывается.

#### **Возврат**

Строка. Полное имя экспортируемого файла.

#### **Замечания**

Версия документа архива помечается как экспортированная. Экспортированный файл не удаляется.

#### **Версия**

5.60 сборка 150508.

## **CreateOpenDocumentDialog**

Создает новый объект для работы пользователя с окном открытия документа архива.

#### **Возврат**

DOpenDocumentDialog. Созданный сервис обработки.

## **GetOperation**

Возвращает объект типа DPDMOperation и загружает в него указанное действие над объектом.

#### **Аргументы**

*ID*

Long. ID действия над объектом.

*ObjectService*

DPDMObjectService. Не обязательный. Сервисный объект, который действие будет использовать для обработки данных. Если указан, то по окончании выполнения действия не будет автоматически вызываться метод Update, для внесения изменений в БД его необходимо вызвать в явном виде. Требуется версия 5.00 сборка 111003.

#### **Возврат**

DPDMOperation. Объект, содержащий указанное действие над объектом.

#### **Версия**

См. описание метода.

## **2.2 Объект Documents**

Объект Documents содержит коллекцию всех открытых документов в приложении.

### **2.2.1 Свойства объекта Documents**

#### **Count**

Число. Содержит количество элементов.

### **2.2.2 Методы объекта Documents**

#### **Item**

Возвращает указанный элемент коллекции.

#### **Аргументы**

*Ndx*

Число. Индекс элемента. Первый элемент имеет индекс 1.

#### **Возврат**

Объект. Объект с указанным индексом.

## 2.3 Объект Document

Объект Document представляет документ, загруженный в окно приложения. Этот объект позволяет производить операции с содержимым документа и с окном, в котором он загружен.

### 2.3.1 Свойства объекта Document

#### Name

Строка. Имя документа. Это свойство используется по умолчанию. Только для чтения.

#### ClassName

Строка. Имя класса окна. Только для чтения.

#### DataObjects

Объект. Содержит коллекцию всех объектов типа LsData, содержащихся в окне документа. Только для чтения.

#### Saved

Истина, если в окне документа нет несохраненных изменений. Только для чтения.

#### Title

Строка. Заголовок окна.

### 2.3.2 Методы объекта Document

#### Activate

Активизирует окно документа.

#### Close

Закрывает окно документа.

#### Замечания

Заккрытие окна документа происходит после окончания работы скрипта. Данные в окне не сохраняются. Пользоваться методом следует аккуратно, т.к. в некоторых случаях закрытие окна документа может привести к аварийному завершению работы приложения.

#### Save

Сохраняет все изменения в окне документа.

#### Возврат

Число. Результат сохранения данных. Если произошла ошибка, то возвращается отрицательное значение. Ниже приведены возможные значения:

Значение	Описание
1	Успешное сохранение
0	В окне не было измененных данных
-1	В окне есть неверно введенные данные
-3	Введенные данные не прошли проверки
-6	Ошибка записи данных в БД.
-7	Ошибка записи данных в БД.

#### Замечания

В случае возникновения ошибки, пользователю будет показано окно с сообщением об ошибке.

## 2.4 Объект LsData

Объект LsData представляет набор данных и описание их визуального представления. Объекты этого типа представляют различные элементы приложения, они могут быть как визуальными, так и невизуальными. Обычно этими элементами являются отчеты и формы.

Объект имеет два свойства `ClassName` и `DataObject`, которые могут быть использованы при поиске нужного объекта данных в окне.

Нумерация строк в объекте начинается с 1.

### 2.4.1 Свойства объекта LsData

#### ClassName

Строка. Имя класса объекта. Значение зависит от местоположения объекта. Имя класса всегда постоянно. Только для чтения.

#### DataObject

Строка. Имя описания данных объекта. В некоторых случаях может содержать пустую строку. Зависит от использования объекта в приложении. В некоторых случаях может изменяться в процессе выполнения. Только для чтения.

#### ParentDocument

Объект. Документ, в котором расположен текущий объект. Для некоторых объектов может отсутствовать, в этом случае свойство содержит пустое значение (`Empty`). Только для чтения.

### 2.4.2 Основные методы объекта LsData

#### DeletedCount

##### Возврат

Число. Количество удаленных строк в объекте.

#### DeleteRow

Удаляет указанную строку. Если строка была ранее считана из БД, то она переносится в буфер удаленных строк. При сохранении данных такие строки удаляются из БД.

##### Аргументы

*Row*

Число. Номер удаляемой строки.

#### Describe

Возвращает значение свойства элемента.

##### Аргументы

*Syntax*

Строка. Содержит требуемые свойства элемента.

##### Возврат

Строка. Значения запрошенных свойств элементов.

#### Filter

Выполняет фильтрацию данных в соответствии с установленным ранее критерием фильтрации. Отфильтрованные строки не удаляются, а переносятся в специальный буфер. При фильтрации эти строки всегда учитываются и переносятся в основной буфер, если они удовлетворяют критерию фильтра.

## FilteredCount

Возвращает количество отфильтрованных строк.

### Возврат

Число. Количество отфильтрованных строк.

## Find

Осуществляет поиск первой строки, удовлетворяющей условию поиска.

### Аргументы

#### *Find*

Строка. Условие поиска. В условии могут применяться любые выражения и функции допустимые для вычисляемых полей в редакторе форм.

#### *RowStart*

Номер строки, начиная с которой будет вестись поиск. Может быть больше, чем число строк.

#### *RowEnd*

Номер строки, до которой будет вестись поиск. Может быть больше, чем число строк.

### Возврат

Число. Номер первой найденной строки. Если строка не найдена, то возвращается 0. Если в выражение поиска содержится ошибки, то возвращается отрицательное значение.

### Замечания

Для обратного поиска необходимо установить значение RowEnd больше, чем RowStart. Поиск текстовых данных чувствителен к регистру. Примеры строк поиска:

```
Quantity > 0
```

```
Quantity * Price >= 2.50 and Upper(Left(Name, 4)) = "QNTR"
```

## FindGroupChange

Ищет границу группы

### Аргументы

#### *Row*

Номер строки, для которой осуществляется поиск.

#### *Group*

Номер группы, границу которой необходимо найти.

### Возврат

Число. Номер строки, в которой колонка группы принимает новое значение, т.е. начало новой группы.

### Замечания

Номер возвращаемой строки зависит от строки, для которой осуществляется поиск. Если начальная строка Row:

- является первой строкой группы, возвращается номер начальной строки;
- является строкой не в последней группе, возвращается номер первой строки в следующей группе;
- является не первой строкой в последней группе, возвращается 0.

## GetItem

Возвращает значение из указанного поля в указанной строке.

### Аргументы

#### *Row*

Число. Номер строки.

#### *Col*

Строка. Имя поля. Поле может быть вычисляемым.

### Возврат

Значение из указанного поля. Тип данных значения зависит от типа данных указанного поля.

## GetItemStrDT

Возвращает значение из поля Col типа datetime в строке Row в строковом формате.

### Аргументы

*Row*

Число. Номер строки.

*Col*

Строка. Имя поля. Поле может быть вычисляемым.

### Возврат

Строка. Значение из указанного поля.

### Замечания

Этот метод полностью аналогичен методу GetItem и должен вызываться только для полей типа datetime (Дата/Время). Он возвращает значение в фиксированном строковом формате «yyyy-mm-dd hh:mm:ss.ffffff». Метод используется для получения значений типа datetime без потери долей секунд, т.к. в скриптах значение времени хранится с точностью до 3-его знака после точки, а в приложении – до 6-го.

## GetItemDecimal

Возвращает значение из поля Col типа decimal в строке Row.

### Аргументы

*Row*

Число. Номер строки.

*Col*

Строка. Имя поля. Поле может быть вычисляемым.

### Возврат

Decimal. Значение из указанного поля.

### Замечания

Этот метод аналогичен методу GetItem и может вызываться только для полей типа decimal. Для таких полей функция GetItem возвращает значение, преобразованное к типу double. При этом преобразовании значение может получиться неточным вследствие округления. Функция GetItemDecimal возвращает непреобразованное значение, так что ее необходимо использовать, если важна точность значения поля (например, поле содержит ID).

## GetSelectedRow

Ищет номер следующей выделенной строки.

### Аргументы

*AfterRow*

Число. Номер строки, после которой осуществляется поиск.

### Возврат

Число. Номер выделенной строки. Если выделенная строка не найдена, то возвращается 0.

### Замечания

В зависимости от использования, у объекта может не быть выделенных строк.

## GroupCalc

Пересчитывает группы. Пересчет групп может потребоваться после изменения порядка строк или их содержимого. При пересчете групп производится расстановка границ групп и подсчет итогов групп.

## InsertRow

Добавляет новую строку данных в объект.

## Аргументы

### *BeforeRow*

Число. Номер строки, перед которой добавляется новая строка. Если содержит 0 или число, которое больше числа строк, то новая строка добавляется после последней строки.

## Возврат

Число. Номер новой строки.

## IsSelected

Определяет, выделена указанная строка или нет.

## Аргументы

### *Row*

Число. Номер строки.

## Возврат

Истина, строка выделена.

## Замечания

В зависимости от использования, у объекта может не быть выделенных строк. Например, если объект является отчетом.

## Modify

Изменяет свойства элементов. Позволяет изменять оформление объекта.

## Аргументы

### *Syntax*

Строка. Содержит команды для изменения свойств элементов.

## Возврат

Строка. Если изменения выполнены без ошибок, то возвращает пустую строку. Иначе возвращается текст ошибки.

## Reset

Очищает объект от всех строк данных.

## Замечания

Очищаются только данные, условия фильтрации и сортировки остаются неизменными.

## RetrieveRest

Выполняет чтение оставшихся данных, чтение которых было начато ранее.

## Замечания

Некоторые объекты загружают данные из БД не целиком, а только первую часть, необходимую для заполнения области экрана. Обычно, чтение остальной части данных происходит при прокручивании области экрана пользователем. Например, так работают некоторые окна выбора.

## RowCount

## Возврат

Число. Количество строк в объекте.

## RowsDiscard

Очищает (удаляет) диапазон строк в указанном буфере. В отличие от DeleteRow функция не переносит строки в буфер “удаленный”, а уничтожает их, в результате любые изменения (сделанные в этих строках) не попадут в БД при сохранении данных.

## Аргументы

### *StartRow*

Число. Номер первой удаляемой строки.

*EndRow*

Число. Номер последней удаляемой строки. Не может быть меньше *StartRow*.

*Buffer*

Строка. Имя буфера, из которого удаляются строки. Может принимать одно из перечисленных значений (регистр не учитывается):

Primary	Основной буфер данных (отображаемый пользователю).
Delete	Буфер удаленных строк.
Filter	Буфер отфильтрованных строк.

### **Возврат**

Число. Возвращает 1, если строки очищены. В случае ошибки возвращает -1.

## **SaveAs**

Сохраняет содержимое объекта в файл.

### **Аргументы**

*FileName*

Строка. Имя файла, в который будут записаны данные.

*Format*

Строка. Имя формата, в котором необходимо сохранить данные. Может принимать одно из перечисленных значений (регистр не учитывается):

CSV	Значения, разделенные запятой.
dBASE2	Формат dBASE-II.
dBASE3	Формат dBASE-III.
DIF	Data Interchange Format.
Excel	Формат Microsoft Excel.
Excel5	Формат Microsoft Excel версии 5.
Excel8	Формат Microsoft Excel версии 8 и выше.
HTMLTable	Формат HTML-таблицы с использованием тэгов TABLE, TR и TD.
Office	Аналогично «Сохранить в MS Office», в <i>FileName</i> должно быть указан полный путь к файлу с расширением “rtf”, “xls” или “xlsx”
PDF	Portable Document Format (PDF).
PSReport	Формат Powersoft Report (PSR).
SYLK	Формат Microsoft Multiplan.
Text	Текстовый формат. Колонки разделены символом табуляции, строки – символами новой строки.
WK1	Формат Lotus 1-2-3.
WKS	Формат Lotus 1-2-3.
WMF	Формат Windows Metafile.
XML	Extensible Markup Language (XML).
XSLFO	XSL (Extensible Stylesheet Language) Formatting Objects (XSL-FO).

*Heading*

Логический. Должен содержать истину, если необходимо сохранить заголовок колонок. Используется для форматов: CSV, XLS, WKS, TXT и SYLK.

### **Возврат**

Число. Возвращает 1, если файл успешно сохранен. В случае ошибки возвращает -1.

## **SaveAsAscii**

Сохраняет содержимое объекта в файл в текстовом виде.

### **Аргументы**

### *FileName*

Строка. Имя файла, в который будут записаны данные.

### *SeparatorCharacter*

Строка. Разделитель значений в строке. Может отсутствовать. По умолчанию используется символ табуляции.

### *QuoteCharacter*

Строка. Используется для обрамления значений. Может отсутствовать. По умолчанию используется символ двойной кавычки.

### *LineEnding*

Строка. Разделитель строк. Может отсутствовать. По умолчанию используется символы новой строки.

## **Возврат**

Число. Возвращает 1, если файл успешно сохранен. В случае ошибки возвращает -1.

## **SetDetailHeight**

Устанавливает для строк в указанном диапазоне высоту области данных.

### **Аргументы**

#### *StartRow*

Число. Номер первой строки диапазона.

#### *EndRow*

Число. Номер последней строки диапазона.

#### *Height*

Число. Высота области данных в единицах формы.

### **Замечания**

Если для разных строк требуется установить разную высоту, то функцию необходимо вызывать для каждой строки отдельно. Если установить нулевую высоту, то строки не будут видны на экране, при этом данные не удаляются.

## **SetFilter**

Устанавливает критерий фильтрации данных.

### **Аргументы**

#### *Filter*

Строка. Условие фильтрации. В условии могут применяться любые выражения и функции допустимые для вычисляемых полей в редакторе форм.

### **Замечания**

Выражение для текстовых полей чувствительно к регистру. Примеры строк условий фильтрации:

```
Not IsNull(AttrValue)
```

```
Lower(Group Name) = "отдел проверки" and SDate > Date('2006-12-31')
```

## **SetItem**

Устанавливает значение указанному полю в указанной строке.

### **Аргументы**

#### *Row*

Число. Номер строки.

#### *Col*

Строка. Имя поля. Поле не может быть вычисляемым.

#### *Data*

Устанавливаемое значение. Тип данных значения должен совпадать с типом данных колонки.

## **SetItemStrDT**



Устанавливает значение указанному полю с типом `datetime` в указанной строке. Значение передается в строковом виде.

#### Аргументы

##### *Row*

Число. Номер строки.

##### *Col*

Строка. Имя поля. Поле не может быть вычисляемым.

##### *Data*

Устанавливаемое значение. Тип данных значения должен совпадать с типом данных колонки.

#### Замечания

Этот метод полностью аналогичен методу `SetItem` и должен вызываться только для полей типа `datetime` (Дата/Время). Он принимает значение в фиксированном строковом формате «уууу-мм-дд hh:mm:ss.ffffff». Метод используется для установки значений типа `datetime` без потери долей секунд, т.к. в скриптах значение времени хранится с точностью до 3-его знака после точки, а в приложении – до 6-го.

### SetSort

Устанавливает критерий сортировки.

#### Аргументы

##### *Sort*

Строка. Выражение для сортировки в виде:

[*выражение* {A|D}[, *выражение* {A|D}[, ...]]]

где *выражение* – любое допустимое выражение;

A – признак сортировки по возрастанию (установлен по умолчанию, если не указан);

D – признак сортировки по убыванию.

В простейшем случае в качестве выражения указывается имя колонки.

Если строка пустая, то условия сортировки будут отсутствовать.

#### Замечания

Метод устанавливает только условия сортировки, но не сортирует данные.

### Sort

Сортирует данные в соответствии с установленным ранее критерием сортировки.

## 2.4.3 Дополнительные методы объекта `LsData`

### OleExport

Экспортирует отчет по шаблону (MS Word/MS Excel) или в файл формата XML.

#### Аргументы

##### *Template*

Строка. Имя файла шаблона, ID версии документа в строковом виде или значение “XML” для экспорта в XML-файл. При экспорте в MS Excel следует использовать следующий формат имени файла шаблона:

Имя\_файла\_шаблона->(Имя\_листа)

154300010->(Имя\_листа)

Шаблон должен быть предварительно настроен.

##### *OutFile*

Строка. Имя файла, в который будет сохранен создаваемый документ, или ID документа архива в строковом виде, в который нужно экспортировать отчет.

##### *Flags*

Строка. Состоит из двух комбинаций флагов, разделенных знаком точки с запятой. Первая комбинация флагов используется для управления экспортом, вторая – для указания параметров экспорта в файл

формата XML. Параметры выгрузки в XML должны начинаться, разделяться и заканчиваться точкой с запятой.

Первая комбинация может содержать следующие флаги:

- S    показать окно с документом после экспорта
- P    напечатать созданный документ (для XML не поддерживается)
- C    закрыть окно с отчетом после экспорта
- N    не отображать сообщение об успешном экспорте
- F    не сохранять формат полей при экспорте (для XML не поддерживается)

Вторая комбинация может содержать следующие параметры:

- Desc=<имя\_колонки>                      Колонка с описанием объекта
- LinkAttrs=<список\_имен\_колонок>    Список имен колонок, разделенных запятой, которые должны быть обработаны как атрибуты проекта

Пример: lsRpt.OleExport “XML”, “C:\Example.xml”, “C;Desc=col4;LinkAttrs=col6,col7;”

### **Замечания**

Шаблон должен быть предварительно настроен, то есть поля шаблона должны быть заранее привязаны к колонкам отчета.

В отчете, экспортируемом в XML файл, должны присутствовать системные колонки object\_type\_description, isobject\_id, parent\_id и должна быть строка для выделенного (корневого) объекта, которая идентифицируется, если для данной строки isobject\_id = parent\_id.

## **2.5 Объект LsDataSet**

Объект LsDataSet является коллекцией объектов типа LsDataRecord.

### **2.5.1 Свойства объекта LsDataSet**

#### **Count**

Число. Содержит количество элементов.

### **2.5.2 Методы объекта LsDataSet**

#### **Item**

Возвращает указанный элемент коллекции.

#### **Аргументы**

*Ndx*

Число. Индекс элемента. Первый элемент имеет индекс 1.

#### **Возврат**

Объект. Объект с указанным индексом.

#### **Замечания**

Объект поддерживает интерфейс IEnumVARIANT.

## **2.6 Объект LsTransaction**

Объект LsTransaction позволяет выполнять запросы к БД с использованием подключения, которым пользуется само приложение. При помощи этого объекта скрипт может произвести необходимые обновления в БД в рамках одной транзакции с

приложением. Если в процессе обновления возникает ошибка в скрипте или в приложении, то все изменения, внесенные в БД в текущей транзакции, будут отменены.

Использование этого объекта требует осторожности. Не рекомендуется выполнять запросы, прямо или косвенно управляющие транзакцией, т.к. в этом случае может быть нарушена целостность данных, вносимых в БД.

## 2.6.1 Методы объекта LsTransaction

### Execute

Выполняет SQL-запрос к БД, не возвращающий результат.

#### Аргументы

*Statement*

Строка. Запрос для выполнения сервером БД.

#### Замечания

Функция в первую очередь предназначена для вызова хранимых процедур и для выполнения запросов Insert, Update и Delete.

### Select

Выполняет SQL-запрос к БД, возвращающий результат.

#### Аргументы

*Statement*

Строка. Запрос для выполнения сервером БД.

*Columns*

Строка. Описание типа данных полей и их количества в возвращаемом результате. Типы данных перечисляются через запятую. Каждому полю соответствует один тип данных. Порядок перечисления типов данных и их количество должен соответствовать порядку следования полей в результате и количеству полей в результате запроса. Допустимые типы данных:

Char(n)      n – число символов

Date

DateTime

Decimal(n)    n – число разрядов после десятичной точки.

Long

Number

Real

Time

См. замечания. Аргумент может отсутствовать.

#### Возврат

Объект LsDataSet. Объект с результатами запроса.

#### Замечания

Выполнение запросов, возвращающих результат, следует использовать только в случаях, когда при помощи ADO невозможно получить такой же результат. Например, при обработке временных таблиц.

Если аргумент 'Columns' не указан или он содержит пустую строку, то при выполнении запроса список колонок и их имена в объекте 'LsDataSet' и тип данных определяются автоматически. Иначе имена колонок формируются в виде col1, col2,... Указывать типы данных колонок рекомендуется при вызове хранимых процедур, т.к. иначе они будут выполнены дважды.

## 2.7 Объект *LsQuery*

Объект *LsQuery* содержит SQL-условия поиска. Используется в скриптах, вызываемых в окнах выбора. Может содержать несколько SQL-условий поиска и дополнительные параметры для них. Обращение к определенному условию осуществляется по его имени. Набор условий, их имена, вид и параметры зависят от контекста использования.

### 2.7.1 Свойства объекта *LsQuery*

#### *SqlQuery*

Строка. Содержит базовый SQL-запрос.

### 2.7.2 Методы объекта *LsQuery*

#### *IsCustomSet*

Проверяет, установлены ли определенные условия

##### Аргументы

*Name*

Строка. Имя условия.

##### Возврат

Возвращает *True*, если указанный набор условий установлен.

#### *CustomWhere* (формат 1)

Возвращает SQL-условия запроса.

##### Аргументы

*Name*

Строка. Имя условия.

##### Возврат

Строка. Условия запроса. Если указанных условий не существует, то возвращает пустую строку.

#### *CustomWhere* (формат 2)

Устанавливает SQL-условия запроса.

##### Аргументы

*Name*

Строка. Имя условия.

*SqlWhere*

Строка. SQL-условия.

##### Замечания

Если условие с указанным именем не существует, будет добавлено новое условие с таким именем. Иначе будет изменено существующее условие.

#### *CustomParam* (формат 1)

Возвращает параметры, используемые для формирования SQL-условия запроса.

##### Аргументы

*Name*

Строка. Имя условия.

##### Возврат

Любое значение. Параметры условия. Возврат зависит от самих параметров. Если указанных условий не существует, то возвращает пустое значение.

## CustomParam (формат 2)

Устанавливает параметры SQL-условия запроса.

### Аргументы

*Name*

Строка. Имя условия.

*Param*

Любое значение. Параметры для указанного условия.

### Замечания

Если условие с указанным именем не существует, будет добавлено новое условие с таким именем. Иначе будет изменено существующее условие.

## 2.8 Объект DConnectAttrib

Объект DConnectAttrib содержит параметры подключения Lotsia PDM PLUS к БД.

До подключения к БД свойства объекта содержат параметры, установленные при настройке рабочего мета. В этот момент свойства объекта могут быть изменены. При записи любого значения в одно из свойств DBType, Driver, Server, DataBase, DBParm пользователь не сможет изменить параметры подключения к БД в окне настройки рабочего мета в запущенной копии Lotsia PDM PLUS. Изменения этих свойств не сохраняются в настройках рабочего места (в реестре). Запись любого значения в свойство User и последующий вызов метода Password подавляет появление окна ввода имени пользователя и пароля при подключении к БД.

После подключения к БД изменение любого свойства объекта не допускается.

### 2.8.1 Свойства объекта DConnectAttrib

#### DBType

Строка. Содержит тип БД. Может содержать одно из значений:

Значение	Тип
SQLANY	Sybase SQL Anywhere
MSSQL	Microsoft SQL server
ORACLE	Oracle

#### Driver

Строка. Содержит тип драйвера, используемого для подключения к БД.

При подключении к Sybase SQL Anywhere может содержать одно из значений:

Значение	Описание
ODBC	Использовать ODBC-источник
5	Sybase SQL Anywhere 5.0
6	Adaptive Server Anywhere 6.0
7	Adaptive Server Anywhere 7.0
8	Adaptive Server Anywhere 8.0
9	Adaptive Server Anywhere 9.0
10	SQL Anywhere 10
11	SQL Anywhere 11

При подключении к Microsoft SQL Server не используется.

При подключении к Oracle может содержать одно из значений:

<b>Значение</b>	<b>Описание</b>
O90	Oracle 9i
O10	Oracle 10g
ORA	Oracle 11g

## Server

Строка. Содержимое зависит от типа БД:

<b>Тип БД</b>	<b>Описание</b>
Sybase SQL Anywhere	При использовании ODBC содержит имя ODBC-источника, иначе содержит имя сервера.
Microsoft SQL Server	Содержит имя сервера.
Oracle	Содержит имя сервера (алиас).

## DataBase

Строка. Содержит имя БД на сервер. Используется при подключении к Sybase SQL Anywhere с использованием ODBC и Microsoft SQL Server.

## DBParm

Строка. Содержит дополнительные параметры.

## User

Строка. Содержит имя регистрации пользователя. При использовании авторизации ОС содержит пустую строку.

## 2.8.2 Методы объекта DConnectAttrib

### Password

Устанавливает пароль пользователя.

#### Аргументы

*Password*

Строка. Пароль.

#### Замечания

Метод может быть вызван только до подключения к БД.

### 3 Редактирование скриптов

Вызов редактора скриптов зависит от контекста выполнения скрипта. Общий вид Окна редактора представлен на рисунке (Рисунок 1).

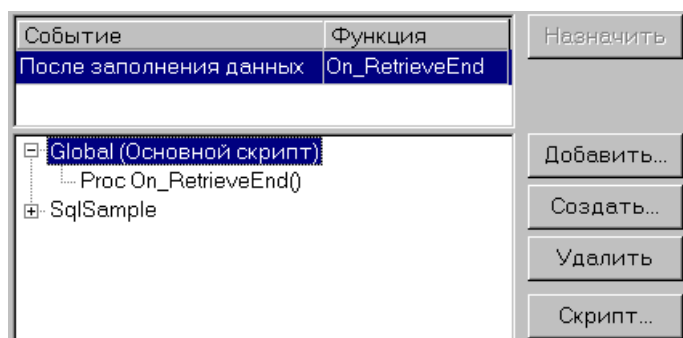
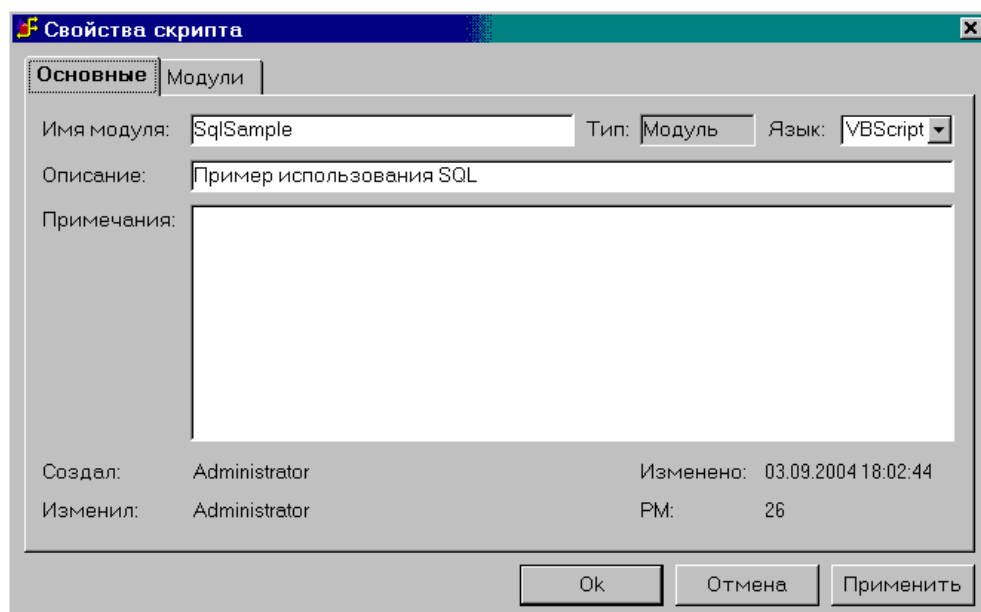


Рисунок 1 Общий вид Окна редактора скриптов

В нижней части находится список модулей, связанных с основным модулем. Если основной модуль еще не создан, то список будет пустым. В ветке каждого модуля содержится список его функций. Редактирование модуля производится в отдельном Окне, для вызова которого необходимо его выделить и нажать на кнопку «**Скрипт**». Эта же кнопка используется для создания основного модуля, если ранее он не был создан. Кнопками «**Добавить**» и «**Удалить**» можно изменять список модулей, связанных с основным модулем. При удалении модуля он не удаляется из БД, а просто исключается из списка связанных. При удалении основного модуля (Global) он удаляется физически из БД во время сохранения изменений.

В верхней части находится список доступных событий, по которым может быть вызван скрипт. Напротив каждого события в колонке «Функция» может быть указано имя функции, вызываемой при наступлении события. Допускается указывать функцию из любого связанного модуля. Если используется функция не из основного модуля, то перед ее именем должно быть указано имя модуля через точку: «Имя\_модуля.Имя\_функции». Имя функции может быть введено (или удалено) вручную или вставлено из нижней части нажатием на кнопку «**Назначить**».

В Окне редактирования модуля редактируется текст скрипта. Допускается существование скрипта с пустым текстом. По кнопке «**Свойства**» вызывается окно свойств модуля, имеющее вид, представленный на рисунке (Рисунок 2).



**Рисунок 2** Окно свойств модуля

На вкладке «Основные» устанавливаются основные параметры модуля. Поле «Имя модуля» является обязательным для заполнения. Для основного модуля это поле содержит значение «Global» и недоступно для редактирования. В поле «Язык» необходимо установить язык скрипта. Язык может быть изменен в любой момент, но при этом пользователем должен быть соответствующим образом изменен текст скрипта. На вкладке «Модули» редактируется список связанных модулей.

При нажатии кнопки «**Ok**» или «**Применить**» в Окне редактирования модуля производится проверка синтаксиса скрипта. Уровень проверки зависит от используемого языка и модуля, его обрабатывающего. После проверки производится запись изменений в БД, если Окно было открыто для скрипта типа «Модуль». Изменения основного модуля не сохраняются в БД, а передаются редактору скриптов, из которого было вызвано Окно редактирования модуля. Изменения в основном модуле записываются в БД одновременно с сохранением того объекта программы (например, отчета), для которого он создавался.



## 4 Скрипты в генераторе отчетов

Для использования скриптов предназначена вкладка «Скрипт» Окна свойств отчета (Рисунок 3).

В отчетах для скриптов доступны события «Двойной щелчок мыши», «После заполнения данных» и «Щелчок мыши». При выполнении отчета производится выборка данных из БД в соответствии с настройкой отчета. После чего будет выполнен скрипт, установленный на событие «После заполнения данных».

При вызове событий «Двойной щелчок мыши» и «Щелчок мыши», в назначенные им процедуры передаются два аргумента:

1. Номер строки (число), по которой был выполнен щелчок мышью.
2. Имя колонки (строка), по которой был выполнен щелчок мышью.

В скриптах для отчетов предоставляется глобальный объект LsRpt типа [LsData](#), в котором хранятся все данные отчета, доступ и модификация которых может быть произведены при помощи методов объекта.

Скрипты, использующиеся в генераторе отчетов, позволяют обрабатывать данные, извлеченные в форму отчета, отображать результаты обработки и обращаться к OLE-объектам, например, Word.Application, Excel.Application, ADODB.Connection и т.д. Использование OLE-объекта ADODB.Connection позволяет обращаться к базам данных других приложений.

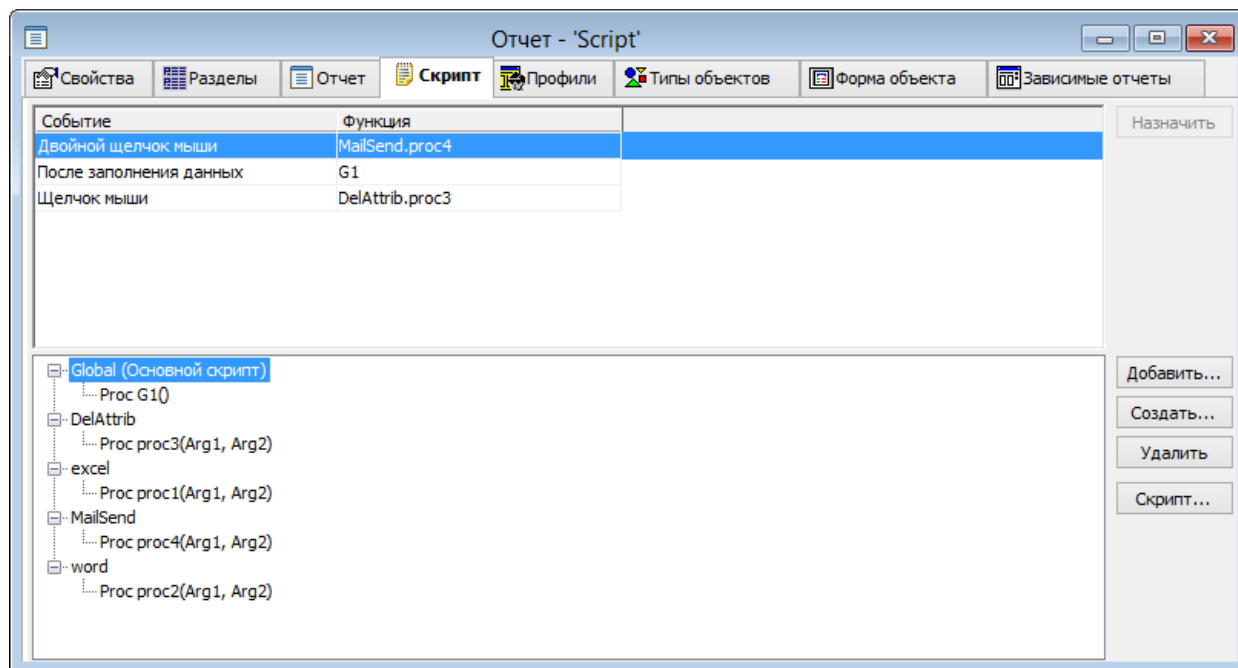


Рисунок 3 Вкладка «Скрипт» Окна свойств отчета

Если при написании скрипта ставится цель вывода данных в большее количество колонок, чем используется в отчете, то добавьте требуемое количество фиктивных колонок соответствующих типов данных (но не вычисляемых полей). Подмена данных в этих колонках производится с помощью функции SetItem. Так, например, нужно составить отчет, в котором, кроме характеристик объектов было бы указано, имеется ли у данного объекта документ архива или нет. Для решения задачи настраиваем отчет нужным образом, затем добавляем одну любую колонку строкового типа, например, описание потомка col5. Затем, при написании скрипта используем SQL-запрос и

учитываем, что полученные данные о наличии или отсутствии документа архива следует вывести в колонку col5, для чего и используем функцию SetItem.

В вызываемой процедуре proc1 модуля Mod1 информацию из глобальной переменной можно поместить в локальную переменную:

```
sub proc1 (aData, name, des)
```

Теперь, в процедуре proc1, при обращении к данным отчета можно указывать не глобальную переменную LsRpt, а локальную переменную aData. Обращение к данным колонки col2 будет происходить под именем name, а к данным колонки col3 под именем des.

## 5 Скрипты в действиях над объектами

Выполнение скриптов в действии над объектами производится на шагах типа «Скрипт». Для каждого действия над объектом, в котором применяются скрипты, создается отдельный глобальный модуль. Этот модуль загружается в память при выполнении действия и остается там до окончания выполнения действия. При вызове другого действия в памяти создается отдельный экземпляр глобального модуля. В каждом шаге типа «Скрипт» можно использовать только одно событие - «Выполнение шага». Во время выполнения скрипта доступны объекты *LsOperation* типа *DPDMSOperation* и *LsVars* типа *LsOperVars*, при помощи которых можно взаимодействовать с выполняемым действием.

### 5.1 Объект *LsOperVars*

Объект *LsOperVars* предназначен для доступа к переменным действия над объектами. В качестве значений переменных сложных типов (Объект, Атрибут и т.п.) используется числовой идентификатор, соответствующий указанному типу. Например, для объекта – это ID объекта.

#### 5.1.1 Методы объекта *LsOperVars*

##### **GetVarValue**

Возвращает значение указанной переменной.

##### **Аргументы**

*VarName*

Строка. Имя переменной.

##### **Возврат**

Значение указанной переменной. Тип данных значения зависит от типа данных указанной переменной.

##### **SetVarValue**

Устанавливает значение указанной переменной.

##### **Аргументы**

*VarName*

Строка. Имя переменной.

*Value*

Устанавливаемое значение. Тип данных значения должен совпадать с типом данных переменной.

## 6 Работа с действиями над объектами

### 6.1 Объект *DPDMSOperation*

Объект *DPDMSOperation* позволяет выполнить действие над объектами. Он содержит информацию по одному конкретному действию над объектами.

#### 6.1.1 Свойства объекта *DPDMSOperation*

##### **ID**

Long. Только для чтения. ID действия над объектами.

##### **Description**

Строка. Только для чтения. Описание действия над объектами.

##### **ObjectService**

*DPDMSObjectService*. Сервисный объект, используемый действием для обработки данных.

##### **Версия**

5.00 сборка 111003.

##### **UiMode**

Long. Режим отображения окна выполнения действия. Может содержать одно из значений:

Значение	Режим
83	Всегда показывать.
65	Показывать при необходимости. Окно выполнения действия будет выведено на экран, только если необходимо отобразить форму.
78	Никогда не показывать. Окно выполнения действия не будет отображаться. При необходимости вывести форму возникнет ошибка выполнения действия. Сообщения об ошибках выполнения действия так же не отображаются пользователю, а вызывают исключение (exception).

##### **Замечания**

Изменение значения свойства не вызывает моментального изменения видимости окна выполнения действия. Значение этого свойства используется при определении необходимости открыть окно в процессе выполнения действия: при запуске действия и при необходимости показать форму. Если окно выполнения действия было открыто, то оно остается открытым до окончания выполнения действия. При выполнении шага типа «Действие» значение этого свойства передается в вызываемое действие.

Изменение этого свойства игнорируется, если действие над объектами выполняется сервером автоматических этапов при включенном флажке «Запретить интерактивные действия» в настройках сервера.

##### **Версия**

5.00 сборка 111003.

##### **UiClose**

Long. Режим выдачи запроса на подтверждение пользователю при попытке остановить выполнение действия. Может содержать одно из значений:

Значение	Тип
89	Выдавать запрос на подтверждение отмены выполнения действия.
78	Не выдавать запрос на подтверждение отмены выполнения действия.

##### **Версия**

5.00 сборка 111003.

## 6.1.2 Методы объекта DPDMOperation

### Execute

Выполняет действие над объектами.

#### Аргументы

##### *ShowError*

Логический. Не обязательный. Если содержит истину, то в случае ошибки выполнения действия будет отображено окно, содержащее текст ошибки. Если содержит ложь, то окно с сообщением об ошибках будет выводиться только в том случае, если окно выполнения действия было открыто, иначе будет возвращена ошибка. По умолчанию содержит истину.

Требуется версия 5.00 сборки 111003.

#### Возврат

Boolean. Возвращает истину, если действие успешно выполнено. Возвращает ложь, если пользователь отказался от выполнения действия (нажал на кнопку «Отмена» или закрыл окно выполнения действия).

#### Замечания

Если ранее не был явно вызван метод PrepareVars, то он вызывается неявно перед началом выполнения действия над объектами.

#### Версия

См. описание метода.

### PrepareVars

Выполняет подготовку переменных для запуска действия над объектами.

#### Замечания

Значения переменных действия со стилем редактирования «Скрытый» и «Просмотр» сбрасываются путем присваивания пустого (Null) значения. Затем всем переменным действия, содержащим пустое значение, устанавливается значение по умолчанию. Если метод PrepareVars не был вызван явно, то метод Execute неявно его вызывает перед началом выполнения действия.

### GetVariables

Возвращает коллекцию переменных действия над объектами.

#### Возврат

DDataSet. Коллекция, содержащая объекты типа DPDMOperationVariable представляющие информацию о переменных действия над объектами. Если действие не содержит ни одной переменной, то коллекция не содержит элементов.

### GetVarValue

Возвращает значение указанной переменной.

#### Аргументы

##### *VarName*

Строка. Имя переменной.

#### Возврат

Значение указанной переменной. Тип данных значения зависит от типа данных указанной переменной.

### SetVarValue

Устанавливает значение указанной переменной.

#### Аргументы

##### *VarName*

Строка. Имя переменной.

##### *Value*

Устанавливаемое значение. Тип данных значения должен совпадать с типом данных переменной.

## Clear

Выполняет освобождение ресурсов, выделенных в процессе выполнения действия.

### Замечания

Метод может быть вызван только после завершения выполнения действия. Если в процессе выполнения действия выполнялся шаг типа «Скрипт», то после завершения выполнения действия в памяти остаются некоторые ресурсы, занятые при выполнении скрипта (например, глобальные переменные). Вызов метода освобождает все подобные ресурсы. После его вызова из памяти выгружаются все модули скриптов действия, тем самым разрушаются глобальные переменные. Вызов метода не запрещает повторное выполнение действия.

Этот метод рекомендуется вызывать после окончания использования объекта `DPDMSession` перед его освобождением. Это позволит полностью освободить память от больше неиспользуемых объектов. Так же наличие неосвобожденных ресурсов может приводить к тому, что приложение будет оставаться запущенным даже после окончания его использования.

### Версия

См. описание метода.

## 6.2 Объект `DPDMSessionVariable`

Объект `DPDMSessionVariable` содержит информацию о переменной действия над объектами.

### 6.2.1 Свойства объекта `DPDMSessionVariable`

#### Value

Значение переменной. Тип данных зависит от типа переменной.

#### Parent

`DPDMSession`. Только для чтения. Объект `DPDMSession`, которому относится текущая переменная действия над объектами.

#### Name

Строка. Только для чтения. Имя переменной.

#### Description

Строка. Только для чтения. Описание переменной.

#### DataType

Long. Только для чтения. Тип переменной. Может содержать одно из значений:

Значение	Тип
83	Строка
78	Число
84	Дата/Время
65	Атрибут
79	Объект
66	Тип объекта
76	Тип связи
67	Код связи
70	Документ
69	Версия документа

85	Пользователь
80	Отчет
71	Приложение
72	Тип документа
73	Шаблон работы
74	Событие календаря
75	Библиотека
81	Шаблон копирования
86	Подборка объектов

## EditType

Long. Только для чтения. Стил редактирования значения переменной. Может содержать одно из значений:

Значение	Тип
70	Выбор файла
68	Выпадающий список
71	Выбор папки
72	Скрытый
86	Просмотр
69	Свободный
87	Список

## 7 Скрипты в шаблонах работ

Выполнение скриптов в работе производится в действиях типа «Скрипт». Для каждого шаблона, в котором применяются скрипты, создается отдельный глобальный модуль. При старте новой работы для новой карты копируется только глобальный модуль скрипта из шаблона работы. При этом копируются и ссылки на остальные используемые модули, но сами модули не копируются. Глобальный модуль загружается в память при первом вызове действия типа «Скрипт» и остается в памяти до закрытия окна задачи. При выполнении задачи на сервере автоматических этапов скрипт остается в памяти до окончания обработки задачи. В каждом действии типа «Скрипт» можно использовать только одно событие - «Выполнение действия». Во время выполнения скрипта доступен объект LsJob типа LsJob, при помощи которого можно получать и изменять значения переменных задачи.

### 7.1 Объект LsJob

Объект LsJob предназначен для доступа к свойствам и переменным выполняемой задачи.

#### 7.1.1 Свойства объекта LsJob

##### ObjectService

DPDMSObjectService. Сервисный объект, обновляемый одновременно с сохранением или выполнением задачи. Для этого объекта запрещен вызов методов Reset и Update.

##### Версия

5.00 сборка 111122.

#### 7.1.2 Методы объекта LsJob

##### GetVarValue

Возвращает значение указанной переменной.

##### Аргументы

*VarName*

Строка. Имя переменной.

##### Возврат

Значение указанной переменной. Тип данных значения зависит от типа данных указанной переменной.

##### SetVarValue

Устанавливает значение указанной переменной.

##### Аргументы

*VarName*

Строка. Имя переменной.

*Data*

Устанавливаемое значение. Тип данных значения должен совпадать с типом данных переменной. Если переменная является массивом, а аргумент – одиночным значением, то это значение помещается в конец массива.



## 8 Скрипты в окне поиска объектов

Если для окна поиска объектов настроено выполнение скрипта (в окне «Общие параметры БД» на вкладке «Поиск объектов»), то условия поиска объектов могут быть изменены. В скрипте могут быть установлены следующие события: «Перед началом выполнения запроса» и «После окончания выполнения запроса». Глобальный модуль скрипта загружается перед выполнением запроса и остается в памяти до окончания выполнения запроса. При выполнении скрипта доступны глобальные объекты LsQuery и LsTransaction. В событии «Перед началом выполнения запроса» могут быть изменены некоторые SQL-условия поиска объектов или добавлены свои условия. Основной запрос не может быть изменен. Перед началом выполнения запроса все SQL-условия, содержащиеся в объекте LsQuery, объединяются между собой и с условиями базового запроса оператором И (AND).

При выполнении запроса в окне поиска объектов, если на вкладке «Атрибуты» были указаны какие-либо условия, в объект LsQuery добавляются условия с именем 'LsObjByAttrib'. В качестве параметра (CustomParam) используется объект типа LsDataSet. Этот объект содержит несколько объектов LsRecordSet, соответствующий отдельной строке стандартной формы поиска объекта по атрибутам. Объект LsRecordSet содержит поля:

bOpen	Число. Количество открывающих скобок.
lType	Строка. Тип левого значения: 'A' – атрибут 'O' – объект
lValue	Строка. Используется, если lType содержит 'O': 'D' – описание объекта 'T' – описание типа объекта 'M' – мнемона типа объекта 'C' – вид объекта
lAttribID	Число. Используется, если lType содержит 'A'. Содержит ID атрибута.
Oper	Строка. Логическая операция сравнения: like = > >= < <= <> not like exists not exists
rValue	Строка. Правое значение в строковом представлении.
bClose	Число. Количество закрывающих скобок.
Logic	Строка. Логическая операция: and or and not or not
DataType	Число. Тип данных: (83) S – строка (78) N – число

Формирование SQL-условия выполняется для базового запроса вида:

```
select *  
  from lsdbo.object_reference_view, lsdbo.object_type_view  
 where lsdbo.object_reference_view.type_id = lsdbo.object_type_view.id
```

В зависимости от настроек окна выбора, в запрос могут быть добавлены дополнительные таблицы и может быть изменен перечень извлекаемых полей. Если в объект LsQuery добавить два условия:

```
lsdbo.object_reference_view.description like 'Part%'
```

и

```
lsdbo.object_type_view.description = 'Деталь'
```

то итоговый запрос будет иметь вид:

```
select *  
  from lsdbo.object_reference_view, lsdbo.object_type_view  
 where (lsdbo.object_reference_view.type_id = lsdbo.object_type_view.id) and  
       ((lsdbo.object_reference_view.description like 'Part%') and  
        (lsdbo.object_type_view.description = 'Деталь'))
```

При выполнении событий «Перед началом выполнения запроса» и «После окончания выполнения запроса» скрипт не может вызывать любые функции, выводящие на экран любые окна.

## 9 Работа с объектами и документами

Для работы с объектами и документами создается сервисный объект. Этот объект содержит специальный буфер объектов и документов, в котором хранится информация о текущих объектах, их атрибутах, связях и документах. При обращении к объектам, значениям их атрибутов, связям объектов или документам, нужная информация загружается из БД и помещается в этот буфер. При повторном обращении к этой же информации по возможности используются значения из буфера, если они там есть, производится проверка ее наличия во внутреннем буфере объектов, и, если она там присутствует, то повторного обращения к БД не производится, а необходимые данные берутся из этого буфера. При изменении свойств объектов или документов, значений атрибутов и связей новые данные проходят предварительную проверку на допустимость значений, а также проверку бизнес-правил типа "Действие" (при изменении атрибутов), после чего изменения помещаются в буфер. Для сохранения всех изменений необходимо вызывать метод Update. После проверки бизнес-правил все изменения целиком заносятся в БД. Для очистки буфера используется метод Reset.

### 9.1 Объект *DPDMObjectService*

Объект *DPDMObjectService* является сервисным объектом для операций с объектами и документами.

#### 9.1.1 Свойства объекта *DPDMObjectService*

##### **CheckRule**

Логический. Содержит флаг использования проверки бизнес правил при изменении значений атрибутов. Отключение проверки возможно только при подключении к БД системы Lotsia PDM PLUS администратора. По умолчанию проверка включена.

##### **FastMode**

Логический. Содержит флаг включения быстрого режима. Если режим включен, то при создании объекта, изменении атрибутов или связей объектов выполнение некоторых проверок будет отложено до обновления БД методом Update. Этот режим увеличивает скорость обновления большого количества информации. По умолчанию режим отключен.

##### **PreventUpdate**

Логический. Содержит флаг разрешения обновления данных. Если содержит истину, то вызов метода Update вызовет ошибку. Вызов метода Reset будет проигнорирован. По умолчанию содержит ложь.

##### **Версия**

5.00 сборка 111003.

#### 9.1.2 Методы объекта *DPDMObjectService*

##### **Reset**

Очищает содержимое внутреннего буфера.

##### **Замечания**

Метод целиком очищает буфер от всей содержащейся в нем информации без записи ее в БД. Все объекты, которые были возвращены методами сервиса, потеряют работоспособность. Обращение к их свойствам или методам будет вызывать возникновение ошибки.

##### **Update**

Записывает в БД все изменения из внутреннего буфера с момента предыдущего успешного вызова метода Update или Reset, или с момента создания объекта.

### **Замечания**

Перед записью изменений производится проверка необходимых бизнес правил. Все изменения, содержащиеся во внутреннем буфере, могут быть записаны только целиком, частичное обновление невозможно. Если в процессе записи измененных данных произошла какая-либо ошибка, то ни одно изменение не попадает в БД. После успешного обновления БД, для всех измененных данных снимается признак изменения, но сами данные остаются во внутреннем буфере. Если в процессе обновления БД возникли ошибки, то признаки изменения для всех данных остается прежним.

Если во внутренний буфер не вносились какие-либо изменения, то никаких действий не производится.

## **GetObject**

Возвращает объект.

### **Аргументы**

*ObjectID*

Decimal. ID объекта.

### **Возврат**

DPDMLObject. Требуемый объект. Если объект не найден, то возвращается пустой объект.

### **Замечания**

Метод ищет информацию по указанному объекту во внутреннем буфере. Если информация не найдена, то она загружается из БД и помещается во внутренний буфер.

## **CreateObject**

Создает во внутреннем буфере новый объект указанного типа и обязательные атрибуты.

### **Аргументы**

*ObjectTypeID*

Decimal. ID типа объекта.

*ParentObjectID*

Decimal. Не обязательный. ID родительского объекта.

### **Возврат**

DPDMLObject. Созданный объект.

### **Замечания**

Метод создает во внутреннем буфере новый объект. Для созданного объекта создаются объекты вида "КД", "НКД" и "Проч.Док.", которые помечены в свойствах типа этого объекта как обязательные. Для каждого из новых объектов создаются обязательные атрибуты со значениями по умолчанию.

Если указан параметр ParentObjectID и его значение не содержит 0, то создаваемый объект помечается как созданный для помещения в дерево объекта ParentObjectID. После создания такого объекта необходимо создать связь между ним и родительским объектом при помощи метода AddLink родительского объекта. Связь должна быть создана до вызова метода Update. Для создаваемого объекта автоматически устанавливается тип защиты такой же, как и у родительского объекта, и распространяются все наследуемые права родительского объекта.

Для записи созданных объектов в БД необходимо вызвать метод Update.

## **GetLink**

Возвращает связь между объектами.

### **Аргументы**

*LinkID*

Decimal. ID связи.

### **Возврат**

DPDMLink. Требуемая связь. Если связь не найдена, то возвращается пустой объект.

### **Замечания**

Метод ищет информацию по указанной связи между объектами во внутреннем буфере. Если информация не найдена, то она загружается из БД и помещается во внутренний буфер.

## GetDocument

Возвращает документ архива.

### Аргументы

*DocumentID*

Decimal. ID документа архива.

### Возврат

DPDMDocument. Требуемый документ архива. Если документ не найден, то возвращается пустой объект.

### Замечания

Метод ищет информацию по требуемому документу архива во внутреннем буфере. Если информация не найдена, то она загружается из БД и помещается во внутренний буфер.

## GetDocumentVersion

Возвращает версию документа архива.

### Аргументы

*VersionID*

Decimal. ID версии документа архива.

### Возврат

DPDMDocumentVersion. Требуемая версия документа архива. Если версия не найдена, то возвращается пустой объект.

### Замечания

Метод ищет информацию по требуемой версии документа архива во внутреннем буфере. Если информация не найдена, то она загружается из БД и помещается во внутренний буфер.

## CreateObjectCollection

Создает во внутреннем буфере новую подборку объектов.

### Возврат

DPDMDObjectCollection. Созданная подборка объектов.

### Замечания

Для записи созданной подборки объектов в БД необходимо вызвать метод Update.

### Версия

5.60 сборка 150330.

## GetObjectCollection

Возвращает документ подборку объектов.

### Аргументы

*ObjectCollectionID*

Decimal. ID подборки объектов.

### Возврат

DPDMDObjectCollection. Требуемая подборка объектов. Если подборка объектов не найдена, то возвращается пустой объект.

### Замечания

Метод ищет информацию по требуемой подборке объектов во внутреннем буфере. Если информация не найдена, то она загружается из БД и помещается во внутренний буфер.

### Версия

5.60 сборка 150330.

## 9.2 Объект *DPDMLObject*

Объект *DPDMLObject* содержит информацию об объекте Lotsia PDM PLUS.

### 9.2.1 Свойства объекта *DPDMLObject*

#### **Name**

Строка. Только для чтения. Свойство по умолчанию. Название объекта в соответствии с настройкой Lotsia PDM PLUS «Формирование названия».

#### **Parent**

*DPDMLObjectService*. Только для чтения. Сервисный объект.

#### **ID**

Decimal. Только для чтения. ID объекта.

#### **TypeID**

Decimal. Только для чтения. ID типа объекта.

#### **Description**

Строка. Описание объекта.

#### **Protected**

Логический. Если истина, то объект защищенный.

#### **Rights**

*DObjectRights*. Если объект защищенный, то содержит права объекта.

### 9.2.2 Методы объекта *DPDMLObject*

#### **GetNameForLink**

Возвращает название объекта в соответствии с настройкой Lotsia PDM PLUS «Формирование названия» с учетом атрибутов проекта.

##### **Аргументы**

*TreeLink*

Может содержать:

**Тип данных**

Decimal

*DPDMLink*

**Значение**

ID связи объекта с родительским объектом в проекте.

Связь объекта с родительским объектом в проекте.

##### **Возврат**

Строка. Название объекта.

##### **Версия**

5.00 сборка 120207.

#### **GetAttrib**

Возвращает значение атрибута объекта.

##### **Аргументы**

*AttribID*

Decimal. ID атрибута объекта.

### **Возврат**

DPDMAtrib. Значение указанного атрибута объекта. Если значение атрибута не найдено, то возвращается пустой объект.

### **Замечания**

Метод ищет значение указанного атрибута объекта во внутреннем буфере. Если значение не найдено, то оно загружается из БД и помещается во внутренний буфер. Если у объекта существует несколько значений указанного атрибута, то возвращается первое попавшееся.

## **GetAttribs**

Возвращает коллекцию атрибутов объекта.

### **Аргументы**

*AttribID*

Decimal. Не обязательный. ID атрибута объекта или массив ID атрибутов.

### **Возврат**

DDataSet. Коллекция, содержащая объекты типа DPDMAtrib со значениями указанных атрибутов объекта. Если ни одного значения атрибутов не найдено, то коллекция не содержит элементов.

### **Замечания**

Метод ищет значения указанных атрибутов объекта во внутреннем буфере. Если значение какого-либо атрибута не найдено, то оно загружается из БД и помещается во внутренний буфер. Порядок следования значений в возвращаемой подборке не определен.

## **AddAttrib**

Добавляет новое значение атрибута объекта.

### **Аргументы**

*AttribID*

Decimal. ID атрибута объекта.

### **Возврат**

DPDMAtrib. Значение созданного атрибута объекта.

### **Замечания**

Метод добавляет новое значение атрибута объекта. Если атрибут не является множественным, то при попытке добавления второго значения возникнет ошибка. Если во внутреннем буфере содержится удаленное ранее значение указанного атрибута, то оно будет использовано вместо добавления нового значения. При последующем обновлении БД такое значение атрибута будет считаться измененным.

Если для указанного атрибута установлено значение по умолчанию, то оно будет содержаться в возвращаемом объекте DPDMAtrib. Иначе свойство Value этого объекта будет содержать Null.

Для записи измененного значения в БД необходимо вызвать метод Update соответствующего объекта типа DPDMObjectService.

## **SetAttrib**

Устанавливает объекту значение указанного атрибута. При необходимости, значение атрибута будет добавлено, изменено или удалено.

### **Аргументы**

*AttribID*

Decimal. ID атрибута объекта.

### **Аргументы**

*Value*

Значение атрибута. Тип данных зависит от атрибута.

### **Замечания**

Метод производит поиск значения указанного атрибута во внутреннем буфере, и, если оно не найдено, то оно загружается в буфер из БД. Если во внутреннем буфере содержится удаленное ранее значение

указанного атрибута, то оно будет использовано для обновления значения. При последующем обновлении БД такое значение атрибута будет считаться измененным.

Если Value содержит не пустое значение (не Null), то оно будет установлено указанному атрибуту во внутреннем буфере. Если такого атрибута у объекта ранее не было, то атрибут будет предварительно добавлен.

Если Value содержит пустое значение (Null) и у объекта нет указанного атрибута, то изменения не производятся. Если такой атрибут у объекта существует, то его значение помечается во внутреннем буфере как удаленное.

Если указанный атрибут может быть множественным и у объекта уже есть более одного значения этого атрибута, то операция по изменению или удалению будет произведена над одним из значений.

Для записи измененного значения в БД необходимо вызвать метод Update соответствующего объекта типа DPDMObjectService.

## DeleteAttrib

Удаляет значение атрибута объекта.

### Аргументы

*AttribID*

Может содержать:

Decimal. ID атрибута объекта. Если у объекта существует несколько значений указанного атрибута, то будет удалено первое попавшееся.

DPDMAttrib. Удаляемое значение.

### Замечания

Значение атрибута объекта помечается во внутреннем буфере как удаленное.

Для записи измененного значения в БД необходимо вызвать метод Update соответствующего объекта типа DPDMObjectService.

## CreateOption

Создает исполнение объекта.

### Возврат

DPDMObject. созданное исполнение.

### Замечания

Исполнение объекта создается во внутреннем буфере. Метод автоматически изменяет атрибут «Обозначение» созданного объекта и связывает созданный объект с базовым объектом.

Для записи измененного значения в БД необходимо вызвать метод Update соответствующего объекта типа DPDMObjectService.

## GetLinkedObjects

Возвращает коллекцию связей с объектами.

### Аргументы

*LinkTypeID*

Long. ID типа связи.

*Flags*

Long. Содержит флажки, определяющие режим считывания связей. Может содержать одно из значений:

- 1 Для подчиненных типов связей возвращаются связи текущего объекта с дочерними объектами. Для направленных типов связей возвращаются связи текущего объекта с входящими объектами.
- 2 Для подчиненных типов связей возвращаются связи текущего объекта с родительскими объектами. Для направленных типов связей возвращаются связи текущего объекта с исходящими объектами.
- 3 Используется для возврата равнозначных связей. Объединяет в себе флаги 1 и 2.

### Возврат



DDataset. Коллекция, содержащая объекты типа DPDMLink, представляющие требуемые связи с объектами. Если ни одна связь не найдена, то коллекция не содержит элементов.

#### **Замечания**

Метод ищет информацию по требуемым связям с объектами во внутреннем буфере. Если информация не найдена, то она загружается из БД и помещается во внутренний буфер.

### **AddLink**

Создает связь с объектом.

#### **Аргументы**

*ObjectID*

Decimal. ID объекта, с которым создается связь.

*LinkTypeID*

Long. ID типа создаваемой связи.

#### **Возврат**

DPDMLink. Созданная связь.

#### **Замечания**

При создании подчиненной связи (используется для построения дерева проектов) объект ObjectID будет считаться дочерним объектом. Для новой связи автоматически создаются обязательные атрибуты проекта.

При создании направленной связи, объект ObjectID будет считаться входящим объектом.

Для записи добавленной связи в БД необходимо вызвать метод Update соответствующего объекта типа DPDMObjectService.

### **DeleteLink**

Удаляет связь между объектами.

#### **Аргументы**

*Link*

DPDMLink. Удаляемая связь.

#### **Замечания**

Для записи удаленной связи в БД необходимо вызвать метод Update соответствующего объекта типа DPDMObjectService.

### **GetDocument**

Возвращает документ архива для текущего объекта.

#### **Аргументы**

*DocumentID*

Decimal. ID документа архива.

#### **Возврат**

DPDMDocument. Требуемый документ архива. Если документ не найден, то возвращается пустой объект.

#### **Замечания**

Метод ищет информацию по требуемому документу архива во внутреннем буфере. Если информация не найдена, то она загружается из БД и помещается во внутренний буфер.

### **GetDocuments**

Возвращает коллекцию документов архива для текущего объекта.

#### **Аргументы**

*DocumentsID*

Decimal. Не обязательный. ID документа архива или массив ID документов архива.

#### **Возврат**

DDataset. Коллекция, содержащая объекты типа DPDMDocument представляющие требуемые документы архива. Если ни одного документа не найдено, то коллекция не содержит элементов.

### Замечания

Метод ищет информацию по указанным документам архива во внутреннем буфере. Если информация по какому-либо документу не найдена, то она загружается из БД и помещается во внутренний буфер. Порядок следования документов архива в возвращаемой подборке не определен.

## ImportDocument

Импортирует новый документ архива для объекта.

### Аргументы

#### Source

Может содержать:

Тип данных	Значение
Строка	Имя импортируемого файла.
DPDMDocumentVersion	Версия документа, на основе которой будет создан новый документ. Требуется версия 5.00 сборки 111003.

#### StorageID

Long. Не обязательный. ID хранилища документов, в которое будет помещен создаваемый документ. Если аргумент не указан или содержит 0, то хранилище будет определено по настройкам для текущего пользователя.

#### Mode

Long. Не обязательный. Режим импорта. Может содержать одно из значений:

- 76 Исходный файл уже находится в указанном хранилище и будет в нем зарегистрирован.
- 67 Исходный файл будет скопирован в указанное хранилище.
- 77 Исходный файл будет перенесен в указанное хранилище.

Если аргумент не указан, то используется режим копирования (67). Если аргумент Source содержит версию документа, то значение параметра игнорируется.

#### ApplicationID

Long. Не обязательный. ID приложения. Если аргумент не указан или содержит 0, то приложение будет определено автоматически на основе расширения исходного файла или взято из версии документа, в зависимости от содержимого параметра Source.

#### DocumentTypeID

Long. Не обязательный. ID типа документа. Если аргумент не указан или содержит 0, то тип будет определен автоматически по приложению или взят из версии документа, в зависимости от содержимого параметра Source.

### Возврат

DPDMDocumentVersion. Версия созданного документа.

### Замечания

Для импорта документа в хранилище и записи созданного документа в БД необходимо вызвать метод Update соответствующего объекта типа DPDMObjectService.

### Версия

См. описание метода.

## DeleteDocument

Удаляет документ архива со всеми его версиями.

### Аргументы

#### DocumentID

DPDMDocument. Удаляемый документ.

### Замечания

Для удаления документа архива из БД необходимо вызвать метод Update соответствующего объекта типа DPDMObjectService.

## 9.3 *DPDMAattrib*

Объект *DPDMAattrib* содержит значение атрибута объекта или проекта.

### 9.3.1 Свойства объекта *DPDMAattrib*

#### Value

Значение атрибута. Тип данных зависит от типа атрибута.

#### Parent

Только для чтения. В зависимости от типа атрибута содержит объект типа *DPDMLinkObject* (для атрибута объекта) или *DPDMLinkObject* (для атрибута проекта) к которому относится значение атрибута.

#### AttribID

Decimal. Только для чтения. ID атрибута.

#### Замечания

Для записи измененного значения в БД необходимо вызвать метод *Update* соответствующего объекта типа *DPDMLinkObjectService*.

## 9.4 *DPDMLink*

Объект *DPDMLink* содержит связь между двумя объектами.

### 9.4.1 Свойства объекта *DPDMLink*

#### Parent

*DPDMLinkObjectService*. Только для чтения. Сервисный объект.

#### ID

Decimal. Только для чтения. ID связи.

#### ParentObject

*DPDMLinkObject*. Только для чтения. Родительский объект.

#### ChildObject

*DPDMLinkObject*. Только для чтения. Дочерний объект.

#### LinkTypeID

Long. Только для чтения. ID типа связи.

#### LinkType

Int. Только для чтения. Тип связи. Может содержать одно из значений:

84	Подчиненная.
69	Равнозначная
68	Направленная
86	Старые варианты
80	Исполнения

#### Замечания

Для записи измененного значения в БД необходимо вызвать метод *Update* соответствующего объекта типа *DPDMLinkObjectService*.

## 9.4.2 Методы объекта DPDMLink

### GetAttrib

Возвращает значение атрибута проекта.

#### Аргументы

*AttribID*

Decimal. ID атрибута проекта.

#### Возврат

DPDMAtrib. Значение указанного атрибута проекта. Если значение атрибута не найдено, то возвращается пустой объект.

#### Замечания

Метод может быть вызван только для связи типа «подчиненная».

Метод ищет значение указанного атрибута проекта во внутреннем буфере. Если значение не найдено, то оно загружается из БД и помещается во внутренний буфер. Если у связи существует несколько значений указанного атрибута, то возвращается первое попавшееся.

### GetAttribs

Возвращает коллекцию атрибутов проекта.

#### Аргументы

*AttribID*

Decimal. Не обязательный. ID атрибута проекта или массив ID атрибутов.

#### Возврат

DDataset. Коллекция, содержащая объекты типа DPDMAtrib со значениями указанных атрибутов проекта. Если ни одного значения атрибутов не найдено, то коллекция не содержит элементов.

#### Замечания

Метод может быть вызван только для связи типа «подчиненная».

Метод ищет значения указанных атрибутов проекта во внутреннем буфере. Если значение какого-либо атрибута не найдено, то оно загружается из БД и помещается во внутренний буфер. Порядок следования значений в возвращаемой подборке не определен.

### AddAttrib

Добавляет новое значение атрибута проекта.

#### Аргументы

*AttribID*

Decimal. ID атрибута проекта.

#### Возврат

DPDMAtrib. Значение созданного атрибута проекта.

#### Замечания

Метод может быть вызван только для связи типа «подчиненная».

Метод добавляет новое значение атрибута проекта. Если атрибут не является множественным, то при попытке добавления второго значения возникнет ошибка. Если во внутреннем буфере содержится удаленное ранее значение указанного атрибута, то оно будет использовано вместо добавления нового значения. При последующем обновлении БД такое значение атрибута будет считаться измененным.

Если для указанного атрибута установлено значение по умолчанию, то оно будет содержаться в возвращаемом объекте DPDMAtrib. Иначе свойство Value этого объекта будет содержать Null.

Для записи измененного значения в БД необходимо вызвать метод Update соответствующего объекта типа DPDMObjectService.

### SetAttrib

Устанавливает значение указанного атрибута проекта. При необходимости, значение атрибута будет добавлено, изменено или удалено.

#### **Аргументы**

*AttribID*

Decimal. ID атрибута проекта.

#### **Аргументы**

*Value*

Значение атрибута. Тип данных зависит от атрибута.

#### **Замечания**

Метод может быть вызван только для связи типа «подчиненная».

Метод производит поиск значения указанного атрибута во внутреннем буфере, и, если оно не найдено, то оно загружается в буфер из БД. Если во внутреннем буфере содержится удаленное ранее значение указанного атрибута, то оно будет использовано для обновления значения. При последующем обновлении БД такое значение атрибута будет считаться измененным.

Если Value содержит непустое значение (не Null), то оно будет установлено указанному атрибуту во внутреннем буфере. Если такого атрибута проекта ранее не было, то атрибут будет предварительно добавлен.

Если Value содержит пустое значение (Null) и у связи нет указанного атрибута, то изменения не производятся. Если такой атрибут проекта существует, то его значение помечается во внутреннем буфере как удаленное.

Если указанный атрибут может быть множественным и у связи уже есть более одного значения этого атрибута, то операция по изменению или удалению будет произведена над одним из значений.

Для записи измененного значения в БД необходимо вызвать метод Update соответствующего объекта типа DPDMObjectService.

### **DeleteAttrib**

Удаляет значение атрибута проекта.

#### **Аргументы**

*AttribID*

Может содержать:

Decimal. ID атрибута проекта. Если у связи существует несколько значений указанного атрибута, то будет удалено первое попавшееся.

DPDMAttrib. Удаляемое значение.

#### **Замечания**

Метод может быть вызван только для связи типа «подчиненная».

Значение атрибута проекта помечается во внутреннем буфере как удаленное.

Для записи измененного значения в БД необходимо вызвать метод Update соответствующего объекта типа DPDMObjectService.

### **AddVariant**

Добавляет объект к проекту как вариант.

#### **Аргументы**

*ObjectID*

Decimal. ID добавляемого объекта.

#### **Возврат**

DPDMLink. Созданная связь.

#### **Замечания**

Метод разрывает текущую связь и добавляет в ее разрыв объект типа «Вариант». Если ранее для этой связи существовали варианты, то будет использован старый объект типа «Вариант», иначе он будет создан.

Для записи добавленной связи в БД необходимо вызвать метод Update соответствующего объекта типа DPDMObjectService.

## SetVariant

Устанавливает связь как основной вариант.

### Аргументы

*LinkID*

Decimal. ID связи объекта типа «Вариант» с основным вариантом.

### Возврат

DPDMLink. Созданная связь.

### Замечания

Метод разрывает связь LinkID и создает связь родительского объекта с объектом, принимаемым как основной вариант.

Для записи добавленной связи в БД необходимо вызвать метод Update соответствующего объекта типа DPDMObjectService.

## 9.5 DPDMDocument

Объект DPDMDocument содержит документ архива.

### 9.5.1 Свойства объекта DPDMDocument

#### Parent

DPDMObject. Только для чтения. Объект, к которому относится документ архива.

#### ID

Decimal. Только для чтения. ID документа архива.

#### Description

Строка. Описание документа.

#### TypeID

Long. ID типа документа архива.

#### AppID

Long. Только для чтения. ID приложения.

#### StorageID

Long. Только для чтения. ID хранилища.

#### Note

Строка. Примечания документа.

#### Rights

DObjectRights. Только для чтения. Если документ архива защищенный, то содержит права на документ.

#### ActualVersion

DPDMDocumentVersion. Только для чтения. Содержит актуальную версию документа.

#### Версия

5.60 сборка 151027.

## 9.5.2 Методы объекта DPDMDocument

### GetVersion

Возвращает версию текущего документа архива.

#### Аргументы

*VersionID*

Decimal. ID версии документа архива.

#### Возврат

DPDMDocumentVersion. Требуемая версия документа архива. Если версия не найдена, то возвращается пустой объект.

#### Замечания

Метод ищет информацию по требуемой версии документа архива во внутреннем буфере. Если информация не найдена, то она загружается из БД и помещается во внутренний буфер.

### GetVersions

Возвращает коллекцию версий текущего документа архива.

#### Возврат

DDataSet. Коллекция, содержащая объекты типа DPDMDocumentVersion представляющие версии документов архива.

#### Замечания

Метод ищет все версии текущего документа во внутреннем буфере. Если какие-либо данные не найдены, то они загружаются из БД и помещаются во внутренний буфер. Порядок следования значений в возвращаемой подборке не определен.

### CreateVersion

Создает новую версию текущего документа архива.

#### Аргументы

*Source*

Может содержать:

Тип	Значение
Строка	Имя импортируемого файла.
DPDMDocumentVersion	Версия документа архива, на основании которой создается новая версия.

#### Mode

Long. Не обязательный. Режим импорта. Используется, если аргумент Source содержит имя файла.

Может содержать одно из значений:

- 76 Исходный файл уже находится в указанном хранилище и будет в нем зарегистрирован.
- 67 Исходный файл будет скопирован в указанное хранилище.
- 77 Исходный файл будет перенесен в указанное хранилище.

Если аргумент не указан, то используется режим копирования (67).

#### Возврат

DPDMDocumentVersion. Версия созданного документа.

#### Замечания

Для импорта документа в хранилище и записи созданной версии документа в БД необходимо вызвать метод Update соответствующего объекта типа DPDMObjectService.

### DeleteVersion

Удаляет версию текущего документа архива.

#### Аргументы

## *Version*

DPDMDocumentVersion. Удаляемая версия документа архива.

### **Замечания**

Метод не может удалить единственную версию документа архива.

Для удаления версии документа из хранилища и из БД необходимо вызвать метод Update соответствующего объекта типа DPDMObjectService.

## **9.6 *DPDMDocumentVersion***

Объект DPDMDocumentVersion содержит версию документа архива.

### **9.6.1 Свойства объекта DPDMDocumentVersion**

#### **Parent**

DPDMDocument. Только для чтения. Документ архива, к которому относится версия документа.

#### **ID**

Decimal. Только для чтения. ID версии документа архива.

#### **Label**

Строка. Только для чтения. Номер версии документа.

#### **Note**

Строка. Примечания версии документа.

#### **IsActual**

Логический. Содержит истину, если версия документа является актуальной.

#### **Версия**

5.60 сборка 151027.

#### **Rights**

DObjectRights. Только для чтения. Если документ архива защищенный, то содержит права на версию документа.

## **9.7 *DOpenDocumentDialog***

При помощи этого объекта осуществляется взаимодействие с окном открытия документа архива в приложении.

### **9.7.1 Свойства объекта DOpenDocumentDialog**

#### **MethodID**

Long. Только для чтения. Содержит ID последнего использованного метода запуска приложения для открытия документа архива.

#### **DocumentVersionID**

Decimal. Только для чтения. Содержит ID последней выбранной версии документа архива.

### **9.7.2 Методы объекта DOpenDocumentDialog**

#### **OpenDocument**



Открывает документ архива в приложении. Предварительно пользователю предлагается выбрать версию документа. Если необходимо, то пользователю предлагается выбрать метод открытия документа архива в приложении.

#### Аргументы

##### *DocumentID*

Decimal. ID открываемого документа архива.

##### *OpenMode*

Long. Не обязательный. Содержит режим открытия документа архива. Значение этого аргумента игнорируется, если аргумент MethodID содержит ненулевое значение. Может содержать одно из значений:

Значение	Описание
0	Любой режим открытия.
79	Открытие документа.
86	Просмотр документа.
80	Печать документа.

##### *MethodID*

Long. Не обязательный. Содержит ID метода запуска приложения, который необходимо использовать для открытия документа. Если указан и содержит не нулевое значение, то аргумент OpenMode игнорируется.

#### Возврат

Boolean. Содержит истину, если документ был успешно открыт. Содержит ложь, если пользователь отказался от открытия документа архива.

### OpenDocumentVersion

Открывает версию документа архива в приложении. Если необходимо, то пользователю предлагается выбрать метод открытия документа архива в приложении.

#### Аргументы

##### *DocumentVersionID*

Decimal. ID открываемой версии документа архива.

##### *OpenMode*

Long. Не обязательный. Содержит режим открытия версии документа архива. Значение этого аргумента игнорируется, если аргумент MethodID содержит ненулевое значение. Может содержать одно из значений:

Значение	Описание
0	Любой режим открытия.
79	Открытие документа.
86	Просмотр документа.
80	Печать документа.

##### *MethodID*

Long. Не обязательный. Содержит ID метода запуска приложения, который необходимо использовать для открытия версии документа. Если указан и содержит не нулевое значение, то аргумент OpenMode игнорируется.

#### Возврат

Boolean. Содержит истину, если документ был успешно открыт. Содержит ложь, если пользователь отказался от открытия версии документа архива.

## 9.8 Объект *DPDMObjectCollection*

Объект DPDMObjectCollection содержит информацию о подборке объектов Lotsia PDM PLUS.

## 9.8.1 Свойства объекта DPDMObjectCollection

### ID

Decimal. Только для чтения. ID подборки объектов. Содержит null для новой подборки объектов.

### Description

Строка. Описание подборки объектов.

### IsPersonal

Логический. Если истина, то подборка объектов является персональной. Если подборка объектов была сохранена в БД как общая, то изменение этого свойства не допускается.

### Rights

DObjectRights. Если подборка объектов общая, то содержит права подборки объектов.

## 9.8.2 Методы объекта DPDMObjectCollection

### GetObjectsID

Возвращает массив ID объектов, содержащихся в подборке объектов.

#### Возврат

Массив Decimal. Массив с ID объектов. Если ни одного объекта не найдено, то возвращается пустой массив.

#### Замечания

Порядок следования значений в возвращаемой подборке не определен.

### GetObjects

Возвращает коллекцию объектов, содержащихся в подборке объектов.

#### Возврат

DDataSet. Коллекция, содержащая объекты типа DPDMObject с информацией об объектах. Если ни одного объекта не найдено, то коллекция не содержит элементов.

#### Замечания

Метод ищет информацию по объектам во внутреннем буфере. Если информация не найдена, то она загружается из БД и помещается во внутренний буфер. Порядок следования значений в возвращаемой подборке не определен.

### AddObjects

Добавляет объекты в подборку объектов.

#### Аргументы

##### *Objects*

Decimal, массив Decimal или DPDMObject. Добавляемые объекты.

##### *LinkTypeID*

Long. Не обязательный. ID типа связи. Если аргумент указан, то в подборку объектов дополнительно будут добавлены все дочерние объекты из дерева проектов добавляемых объектов по указанному типу связи.

#### Замечания

Метод добавляет объекты в подборку объектов. Если добавляемые объекты уже существуют в подборке объектов, то они повторно не добавляются.

Для записи изменений в подборке объектов в БД необходимо вызвать метод Update соответствующего объекта типа DPDMObjectService.

## DeleteObjects

Удаляет объекты из подборки объектов.

### Аргументы

*Objects*

Decimal, массив Decimal или DPDMObject. Удаляемые объекты.

### Замечания

Метод удаляет объекты из подборки объектов. Если указанных объектов не существует в подборке объектов, то они игнорируются.

Для записи изменений в подборке объектов в БД необходимо вызвать метод Update соответствующего объекта типа DPDMObjectService.

## Delete

Удаляет подборку объектов.

### Замечания

Метод помечает подборку как удаленную. После вызова этого метода обращения к свойствам и методам подборки будут вызывать ошибку.

Для удаления подборки объектов из БД необходимо вызвать метод Update соответствующего объекта типа DPDMObjectService.

## 10 Работа с правами пользователя

### 10.1 *DObjectRights*

Объект *DObjectRights* содержит права на определенный объект системы (объект, документ и т.д.).

#### 10.1.1 Свойства объекта *DObjectRights*

##### Parent

Только для чтения. Объект, для которого установлены текущие права.

#### 10.1.2 Методы объекта *DObjectRights*

##### Items

Возвращает коллекцию записей из списка прав.

##### Аргументы

###### *RightID*

Long. Не обязательный. Содержит ID права. Если указан и содержит значение, отличное от 0, то метод вернет только те записи, которые относятся к этому праву.

###### *UserID*

Long. Не обязательный. ID пользователя или группы. Если указан, то метод вернет только те записи, которые установлены для этого пользователя или этой группы.

##### Возврат

*DDataset*. Коллекция, содержащая объекты типа *DRightEntry* хранящие записи из списка прав. Если ни одной записи не найдено, то коллекция не содержит элементов.

##### Add

Добавляет новую запись в список прав.

##### Аргументы

###### *RightID*

Long. ID добавляемого права. Допустимые значения аргумента зависят от объекта, для которого назначаются права.

###### *UserID*

Long. ID пользователя или группы, для которых назначаются права.

###### *Inherit*

Логический. Не обязательный. Если содержит истину, то добавляемые права добавляются как наследуемые. По умолчанию содержит ложь.

###### *Grant*

Строка. Не обязательный. Содержит устанавливаемые разрешения.

Строка всегда содержит четыре символа. Каждый символ отвечает за свое разрешение. Символы в строке расположены в фиксированном порядке: доступ, создание, изменение, удаление. Каждый символ может содержать одно из значений (буквы латинского алфавита):

Значение	Описание
<пробел>	Разрешение не указано в этой записи.
D	Запрещено.
E	Разрешено.
G	Разрешено с правом передачи.

##### Возврат

*DRightEntry*. Созданная запись.

### Замечания

Если не указаны разрешения, то запись создается с пустыми разрешениями.

Запись изменений в БД производится одновременно с сохранением изменений объекта, для которого установлены права.

## Delete

Удаляет запись из списка прав.

### Аргументы

*Entry*

DRightEntry. Удаляемая запись.

### Замечания

Запись изменений в БД производится одновременно с сохранением изменений объекта, для которого установлены права.

## GetEffective

Возвращает действующие права для указанного пользователя или группы.

### Аргументы

*RightID*

Long. ID проверяемых прав. Допустимые значения аргумента зависят от объекта, для которого назначаются права.

*UserID*

Long. Не обязательный. ID пользователя или группы, для которых проверяются права. Если не указан, то метод выполняется для текущего пользователя.

### Возврат

Строка. Действующие разрешения.

Строка всегда содержит четыре символа. Каждый символ отвечает за свое разрешение. Символы в строке расположены в фиксированном порядке: доступ, создание, изменение, удаление. Каждый символ может содержать одно из значений (буквы латинского алфавита):

Значение	Описание
D	Запрещено.
E	Разрешено.
G	Разрешено с правом передачи.

### Замечания

Метод возвращает суммарные разрешения для пользователя или группы на текущий момент напрямую из БД. При этом не учитываются внесенные, но не сохраненные изменения в права, а также не учитывается признак защищенности для объекта.

### Версия

5.00 сборка 120220.

## 10.2 DRightEntry

Объект DRightEntry содержит запись списка прав на определенный объект системы (объект, документ и т.д.).

### 10.2.1 Свойства объекта DRightEntry

#### Parent

DObjectRights. Только для чтения. Список прав, к которому относится запись.

#### Grant

Строка. Содержит устанавливаемые разрешения. Пользователь должен обладать достаточными правами для возможности изменения этого свойства.

Строка всегда содержит четыре символа. Каждый символ отвечает за свое разрешение. Символы в строке расположены в фиксированном порядке: доступ, создание, изменение, удаление. Каждый символ может содержать одно из значений (буквы латинского алфавита):

Значение	Описание
<пробел>	Разрешение не указано в этой записи.
D	Запрещено.
E	Разрешено.
G	Разрешено с правом передачи.

## ID

Long. Только для чтения. ID записи.

## UserID

Long. Только для чтения. ID пользователя или группы, для которых установлены права.

## RightID

Long. Только для чтения. ID права.

## Inherit

Long. Только для чтения. Тип наследования. Может содержать значения:

Значение	Описание
78	Непосредственные права.
89	Если InheritLevel содержит 0, то это наследуемые права. Иначе это унаследованные права от родителя.
82	Это унаследованные права от родителя, с которым была разорвана связь пользователем, у которого не было достаточных административных прав для удаления этой записи.

## InheritLevel

Long. Только для чтения. Уровень наследования.

## ParentID

Long. Только для чтения. ID записи, от которой унаследована текущая запись.

## GrantorID

Long. Только для чтения. ID пользователя, изменившего запись.

## CD

Date. Только для чтения. Дата и время последней модификации записи.

## WsID

Long. Только для чтения. Номер рабочего места, с которого было произведена последние изменения записи.

## 10.3 Перечень прав пользователя

Для каждого типа объекта системы (объект, документ, и т.д.) существует свой уникальный набор прав пользователя. Допустимые права приведены в таблице ниже. Символом ‘\*’ обозначены разрешения, которые могут быть использованы.

Тип объекта RightID	Д	С	И	У	Описание
<b>Объект</b>					
2004000	*		*		Доступ к объекту
2004001	*	*	*	*	Доступ к атрибутам объекта
2004003	*	*	*	*	Доступ к проекту
2004004		*			Создание исполнения
2004005		*	*	*	Правка ЭЦП
<b>Документ:</b>					
2000000	*	*	*	*	Правка версий документа
<b>Версия документа</b>					
2001000	*		*	*	Правка версии документа
<b>Подборка объектов</b>					
2009000	*		*	*	Правка подборки объектов

## 11 Работа с окнами выбора

### 11.1 Объект *DEmployeeSelection*

При помощи этого объекта осуществляется взаимодействие с окном выбора пользователя или группы пользователей.

#### 11.1.1 Свойства объекта *DEmployeeSelection*

##### Title

Строка. Текст заголовка окна выбора. Если содержит пустую строку, то текст заголовка формируется автоматически.

#### 11.1.2 Методы объекта *DEmployeeSelection*

##### ID

Возвращает последние выбранные значения ID пользователя или группы.

##### Аргументы

###### *Index*

Long. Не обязательный. Индекс возвращаемого значения. Индекс первого значения – 0. См. описание возврата.

##### Возврат

Зависит от аргумента *Index*:

Тип	Описание
Long	Если аргумент <i>Index</i> указан. Возвращается значение с индексом <i>Index</i> .
Массив Long	Если аргумент <i>Index</i> не указан. Возвращается массив всех выбранных значений.

##### Select

Выводит окно выбора пользователя или группы.

##### Аргументы

###### *MultiSelect*

Boolean. Не обязательный. Если аргумент указан и содержит истину, то пользователь в окне выбора может выбрать сразу несколько строк. Иначе допускается выбор только одной строки.

##### Возврат

Строка. Код нажатой клавиши в окне выбора. Может принимать одно из значений:

Значение	Описание
OK	Пользователь выбрал одно или несколько значений и нажал на кнопку ОК.
CANCEL	Пользователь отказался от выбора, закрыв окно или нажав на кнопку «Отмена».

##### Замечания

При успешном выборе строк пользователем в окне выбора в объекте сохраняются соответствующие ID выбранных данных вместо предыдущих. Если было выбрано сразу несколько значений, то порядок их следования в массиве не определен.

### 11.2 Объект *DObjectSelection*

При помощи этого объекта осуществляется взаимодействие с окном выбора объектов.



## 11.2.1 Свойства объекта DObjectSelection

### Title

Строка. Текст заголовка окна выбора. Если содержит пустую строку, то текст заголовка формируется автоматически.

### TreeLinkId

Decimal. Только для чтения. Содержит ID связи выбранного объекта в дереве при вызове функции SelectTree.

## 11.2.2 Методы объекта DObjectSelection

### ID

Возвращает последние выбранные значения ID объектов.

#### Аргументы

##### *Index*

Long. Не обязательный. Индекс возвращаемого значения. Индекс первого значения – 0. См. описание возврата.

#### Возврат

Зависит от аргумента Index:

Тип	Описание
Decimal	Если аргумент Index указан. Возвращается значение с индексом Index.
Массив Decimal	Если аргумент Index не указан. Возвращается массив всех выбранных значений.

### Select

Выводит окно выбора объектов.

#### Аргументы

##### *MultiSelect*

Boolean. Не обязательный. Если аргумент указан и содержит истину, то пользователь в окне выбора может выбрать сразу несколько строк. Иначе допускается выбор только одной строки.

#### Возврат

Строка. Код нажатой клавиши в окне выбора. Может принимать одно из значений:

Значение	Описание
OK	Пользователь выбрал одно или несколько значений и нажал на кнопку ОК.
CANCEL	Пользователь отказался от выбора, закрыв окно или нажав на кнопку «Отмена».

#### Замечания

При успешном выборе строк пользователем в окне выбора в объекте сохраняются соответствующие ID выбранных данных вместо предыдущих. Если было выбрано сразу несколько значений, то порядок их следования в массиве не определен.

### SelectTree

Выводит окно выбора объектов из указанного дерева проекта.

#### Аргументы

##### *RootObjectID*

Decimal. ID объекта, являющегося корнем дерева проекта.

##### *LinkTypeID*

Long. ID типа связи, используемого для построения дерева проектов.

#### Возврат

Строка. Код нажатой клавиши в окне выбора. Может принимать одно из значений:

Значение	Описание
OK	Пользователь выбрал одно или несколько значений и нажал на кнопку ОК.
CANCEL	Пользователь отказался от выбора, закрыв окно или нажав на кнопку «Отмена».

### Замечания

В этом окне можно выбрать только один объект. При успешном выборе пользователем в объекте `DObjectSelection` сохраняется ID выбранного объекта и ID связи этого объекта с родительским объектом. Если был выбран корневой объект, то ID связи будет содержать 0. Доступ к ID связи можно получить через свойство `TreeLinkID`.

## 11.3 Объект `DOperationSelection`

При помощи этого объекта осуществляется взаимодействие с окном выбора действий над объектами.

### 11.3.1 Свойства объекта `DOperationSelection`

#### Title

Строка. Текст заголовка окна выбора. Если содержит пустую строку, то текст заголовка формируется автоматически.

### 11.3.2 Методы объекта `DOperationSelection`

#### ID

Возвращает последние выбранные значения ID действий над объектами.

#### Аргументы

##### *Index*

Long. Не обязательный. Индекс возвращаемого значения. Индекс первого значения – 0. См. описание возврата.

#### Возврат

Зависит от аргумента `Index`:

Тип	Описание
Long	Если аргумент <code>Index</code> указан. Возвращается значение с индексом <code>Index</code> .
Массив Long	Если аргумент <code>Index</code> не указан. Возвращается массив всех выбранных значений.

#### Select

Выводит окно выбора действий над объектами.

#### Аргументы

##### *MultiSelect*

Boolean. Не обязательный. Если аргумент указан и содержит истину, то пользователь в окне выбора может выбрать сразу несколько строк. Иначе допускается выбор только одной строки.

#### Возврат

Строка. Код нажатой клавиши в окне выбора. Может принимать одно из значений:

Значение	Описание
OK	Пользователь выбрал одно или несколько значений и нажал на кнопку ОК.
CANCEL	Пользователь отказался от выбора, закрыв окно или нажав на кнопку «Отмена».

### Замечания

При успешном выборе строк пользователем в окне выбора в объекте сохраняются соответствующие ID выбранных данных вместо предыдущих. Если было выбрано сразу несколько значений, то порядок их следования в массиве не определен.

## 11.4 Объект DAttribSelection

При помощи этого объекта осуществляется взаимодействие с окном выбора атрибутов.

### 11.4.1 Свойства объекта DAttribSelection

#### Title

Строка. Текст заголовка окна выбора. Если содержит пустую строку, то текст заголовка формируется автоматически.

### 11.4.2 Методы объекта DAttribSelection

#### ID

Возвращает последние выбранные значения ID атрибутов.

#### Аргументы

##### Index

Long. Не обязательный. Индекс возвращаемого значения. Индекс первого значения – 0. См. описание возврата.

#### Возврат

Зависит от аргумента Index:

Тип	Описание
Decimal	Если аргумент Index указан. Возвращается значение с индексом Index.
Массив Decimal	Если аргумент Index не указан. Возвращается массив всех выбранных значений.

#### Select

Выводит окно выбора атрибутов.

#### Аргументы

##### MultiSelect

Boolean. Не обязательный. Если аргумент указан и содержит истину, то пользователь в окне выбора может выбрать сразу несколько строк. Иначе допускается выбор только одной строки.

#### Возврат

Строка. Код нажатой клавиши в окне выбора. Может принимать одно из значений:

Значение	Описание
OK	Пользователь выбрал одно или несколько значений и нажал на кнопку ОК.
CANCEL	Пользователь отказался от выбора, закрыв окно или нажав на кнопку «Отмена».

#### Замечания

При успешном выборе строк пользователем в окне выбора в объекте сохраняются соответствующие ID выбранных данных вместо предыдущих. Если было выбрано сразу несколько значений, то порядок их следования в массиве не определен.

## 11.5 Объект DObjectTypeSelection

При помощи этого объекта осуществляется взаимодействие с окном выбора типов объекта.

### 11.5.1 Свойства объекта DObjectTypeSelection

#### Title

Строка. Текст заголовка окна выбора. Если содержит пустую строку, то текст заголовка формируется автоматически.

## 11.5.2 Методы объекта DObjectTypeSelection

### ID

Возвращает последние выбранные значения ID типов объекта.

#### Аргументы

##### *Index*

Long. Не обязательный. Индекс возвращаемого значения. Индекс первого значения – 0. См. описание возврата.

#### Возврат

Зависит от аргумента Index:

Тип	Описание
Decimal	Если аргумент Index указан. Возвращается значение с индексом Index.
Массив Decimal	Если аргумент Index не указан. Возвращается массив всех выбранных значений.

### Select

Выводит окно выбора типов объекта.

#### Аргументы

##### *MultiSelect*

Boolean. Не обязательный. Если аргумент указан и содержит истину, то пользователь в окне выбора может выбрать сразу несколько строк. Иначе допускается выбор только одной строки.

#### Возврат

Строка. Код нажатой клавиши в окне выбора. Может принимать одно из значений:

Значение	Описание
OK	Пользователь выбрал одно или несколько значений и нажал на кнопку ОК.
CANCEL	Пользователь отказался от выбора, закрыв окно или нажав на кнопку «Отмена».

#### Замечания

При успешном выборе строк пользователем в окне выбора в объекте сохраняются соответствующие ID выбранных данных вместо предыдущих. Если было выбрано сразу несколько значений, то порядок их следования в массиве не определен.

## 11.6 Объект DLinkTypeSelection

При помощи этого объекта осуществляется взаимодействие с окном выбора типов связи.

### 11.6.1 Свойства объекта DLinkTypeSelection

#### Title

Строка. Текст заголовка окна выбора. Если содержит пустую строку, то текст заголовка формируется автоматически.

### 11.6.2 Методы объекта DLinkTypeSelection

### ID

Возвращает последние выбранные значения ID типов связи.

#### Аргументы

##### *Index*

Long. Не обязательный. Индекс возвращаемого значения. Индекс первого значения – 0. См. описание возврата.

### Возврат

Зависит от аргумента Index:

Тип	Описание
Long	Если аргумент Index указан. Возвращается значение с индексом Index.
Массив Long	Если аргумент Index не указан. Возвращается массив всех выбранных значений.

## Select

Выводит окно выбора типов связи.

### Аргументы

#### *MultiSelect*

Boolean. Не обязательный. Если аргумент указан и содержит истину, то пользователь в окне выбора может выбрать сразу несколько строк. Иначе допускается выбор только одной строки.

### Возврат

Строка. Код нажатой клавиши в окне выбора. Может принимать одно из значений:

Значение	Описание
OK	Пользователь выбрал одно или несколько значений и нажал на кнопку ОК.
CANCEL	Пользователь отказался от выбора, закрыв окно или нажав на кнопку «Отмена».

### Замечания

При успешном выборе строк пользователем в окне выбора в объекте сохраняются соответствующие ID выбранных данных вместо предыдущих. Если было выбрано сразу несколько значений, то порядок их следования в массиве не определен.

## 11.7 Объект *DDocumentSelection*

При помощи этого объекта осуществляется взаимодействие с окном выбора документов архива.

### 11.7.1 Свойства объекта *DDocumentSelection*

#### Title

Строка. Текст заголовка окна выбора. Если содержит пустую строку, то текст заголовка формируется автоматически.

### 11.7.2 Методы объекта *DDocumentSelection*

#### ID

Возвращает последние выбранные значения ID документов архива.

### Аргументы

#### *Index*

Long. Не обязательный. Индекс возвращаемого значения. Индекс первого значения – 0. См. описание возврата.

### Возврат

Зависит от аргумента Index:

Тип	Описание
Decimal	Если аргумент Index указан. Возвращается значение с индексом Index.
Массив Decimal	Если аргумент Index не указан. Возвращается массив всех выбранных значений.

## SetObject

Устанавливает ID объекта для ограничения выбора документов архива.

## Аргументы

### *ObjectID*

Decimal. ID объекта. См. описание метода Select.

## Select

Выводит окно выбора документов архива.

## Аргументы

### *MultiSelect*

Boolean. Не обязательный. Если аргумент указан и содержит истину, то пользователь в окне выбора может выбрать сразу несколько строк. Иначе допускается выбор только одной строки.

## Возврат

Строка. Код нажатой клавиши в окне выбора. Может принимать одно из значений:

Значение	Описание
OK	Пользователь выбрал одно или несколько значений и нажал на кнопку OK.
CANCEL	Пользователь отказался от выбора, закрыв окно или нажав на кнопку «Отмена».

## Замечания

Если при помощи метода SetObject был указан ненулевой ID объекта, то выбор будет осуществляться среди документов архива этого объекта. Иначе предварительно будет выведено окно выбора объекта. При успешном выборе строк пользователем в окне выбора в объекте сохраняются соответствующие ID выбранных данных вместо предыдущих. Если было выбрано сразу несколько значений, то порядок их следования в массиве не определен.

## 11.8 Объект DDocumentVersionSelection

При помощи этого объекта осуществляется взаимодействие с окном выбора версий документов архива.

### 11.8.1 Свойства объекта DDocumentVersionSelection

#### Title

Строка. Текст заголовка окна выбора. Если содержит пустую строку, то текст заголовка формируется автоматически.

### 11.8.2 Методы объекта DDocumentVersionSelection

#### ID

Возвращает последние выбранные значения ID версий документов архива.

## Аргументы

### *Index*

Long. Не обязательный. Индекс возвращаемого значения. Индекс первого значения – 0. См. описание возврата.

## Возврат

Зависит от аргумента Index:

Тип	Описание
Decimal	Если аргумент Index указан. Возвращается значение с индексом Index.
Массив Decimal	Если аргумент Index не указан. Возвращается массив всех выбранных значений.

## SetObject

Устанавливает ID объекта для ограничения выбора версий документов архива.

## Аргументы

*ObjectID*

Decimal. ID объекта. См. описание метода Select.

## Select

Выводит окно выбора версий документов архива.

### Аргументы

*MultiSelect*

Boolean. Не обязательный. Если аргумент указан и содержит истину, то пользователь в окне выбора может выбрать сразу несколько строк. Иначе допускается выбор только одной строки.

### Возврат

Строка. Код нажатой клавиши в окне выбора. Может принимать одно из значений:

Значение	Описание
OK	Пользователь выбрал одно или несколько значений и нажал на кнопку ОК.
CANCEL	Пользователь отказался от выбора, закрыв окно или нажав на кнопку «Отмена».

### Замечания

Если при помощи метода SetObject был указан ненулевой ID объекта, то выбор будет осуществляться среди версий документов архива этого объекта. Иначе предварительно будет выведено окно выбора объекта.

При успешном выборе строк пользователем в окне выбора в объекте сохраняются соответствующие ID выбранных данных вместо предыдущих. Если было выбрано сразу несколько значений, то порядок их следования в массиве не определен.

## 11.9 Объект DApplicationSelection

При помощи этого объекта осуществляется взаимодействие с окном выбора приложений.

### 11.9.1 Свойства объекта DApplicationSelection

#### Title

Строка. Текст заголовка окна выбора. Если содержит пустую строку, то текст заголовка формируется автоматически.

### 11.9.2 Методы объекта DApplicationSelection

#### ID

Возвращает последние выбранные значения ID приложений.

#### Аргументы

*Index*

Long. Не обязательный. Индекс возвращаемого значения. Индекс первого значения – 0. См. описание возврата.

#### Возврат

Зависит от аргумента Index:

Тип	Описание
Long	Если аргумент Index указан. Возвращается значение с индексом Index.
Массив Long	Если аргумент Index не указан. Возвращается массив всех выбранных значений.

## Select

Выводит окно выбора приложений.

### Аргументы

### *MultiSelect*

Boolean. Не обязательный. Если аргумент указан и содержит истину, то пользователь в окне выбора может выбрать сразу несколько строк. Иначе допускается выбор только одной строки.

### **Возврат**

Строка. Код нажатой клавиши в окне выбора. Может принимать одно из значений:

<b>Значение</b>	<b>Описание</b>
OK	Пользователь выбрал одно или несколько значений и нажал на кнопку ОК.
CANCEL	Пользователь отказался от выбора, закрыв окно или нажав на кнопку «Отмена».

### **Замечания**

При успешном выборе строк пользователем в окне выбора в объекте сохраняются соответствующие ID выбранных данных вместо предыдущих. Если было выбрано сразу несколько значений, то порядок их следования в массиве не определен.

## **11.10 Объект DStorageSelection**

При помощи этого объекта осуществляется взаимодействие с окном выбора хранилищ документов.

### **11.10.1 Свойства объекта DStorageSelection**

#### **Title**

Строка. Текст заголовка окна выбора. Если содержит пустую строку, то текст заголовка формируется автоматически.

### **11.10.2 Методы объекта DStorageSelection**

#### **ID**

Возвращает последние выбранные значения ID хранилищ документов.

#### **Аргументы**

##### *Index*

Long. Не обязательный. Индекс возвращаемого значения. Индекс первого значения – 0. См. описание возврата.

#### **Возврат**

Зависит от аргумента Index:

<b>Тип</b>	<b>Описание</b>
Long	Если аргумент Index указан. Возвращается значение с индексом Index.
Массив Long	Если аргумент Index не указан. Возвращается массив всех выбранных значений.

#### **Select**

Выводит окно выбора хранилищ документов.

#### **Аргументы**

##### *MultiSelect*

Boolean. Не обязательный. Если аргумент указан и содержит истину, то пользователь в окне выбора может выбрать сразу несколько строк. Иначе допускается выбор только одной строки.

#### **Возврат**

Строка. Код нажатой клавиши в окне выбора. Может принимать одно из значений:

<b>Значение</b>	<b>Описание</b>
OK	Пользователь выбрал одно или несколько значений и нажал на кнопку ОК.
CANCEL	Пользователь отказался от выбора, закрыв окно или нажав на кнопку «Отмена».

### **Замечания**



При успешном выборе строк пользователем в окне выбора в объекте сохраняются соответствующие ID выбранных данных вместо предыдущих. Если было выбрано сразу несколько значений, то порядок их следования в массиве не определен.

## 11.11 *Объект DDocumentTypeSelection*

При помощи этого объекта осуществляется взаимодействие с окном выбора типов документа.

### 11.11.1 Свойства объекта DDocumentTypeSelection

#### Title

Строка. Текст заголовка окна выбора. Если содержит пустую строку, то текст заголовка формируется автоматически.

### 11.11.2 Методы объекта DDocumentTypeSelection

#### ID

Возвращает последние выбранные значения ID типов документа.

#### Аргументы

##### *Index*

Long. Не обязательный. Индекс возвращаемого значения. Индекс первого значения – 0. См. описание возврата.

#### Возврат

Зависит от аргумента Index:

Тип	Описание
Long	Если аргумент Index указан. Возвращается значение с индексом Index.
Массив Long	Если аргумент Index не указан. Возвращается массив всех выбранных значений.

#### Select

Выводит окно выбора типов документа.

#### Аргументы

##### *MultiSelect*

Boolean. Не обязательный. Если аргумент указан и содержит истину, то пользователь в окне выбора может выбрать сразу несколько строк. Иначе допускается выбор только одной строки.

#### Возврат

Строка. Код нажатой клавиши в окне выбора. Может принимать одно из значений:

Значение	Описание
OK	Пользователь выбрал одно или несколько значений и нажал на кнопку ОК.
CANCEL	Пользователь отказался от выбора, закрыв окно или нажав на кнопку «Отмена».

#### Замечания

При успешном выборе строк пользователем в окне выбора в объекте сохраняются соответствующие ID выбранных данных вместо предыдущих. Если было выбрано сразу несколько значений, то порядок их следования в массиве не определен.

## 11.12 *Объект DReportSelection*

При помощи этого объекта осуществляется взаимодействие с окном выбора отчета.

## 11.12.1 Методы объекта DReportSelection

### ID

Возвращает последние выбранные значения ID отчетов.

#### Аргументы

##### *Index*

Long. Не обязательный. Индекс возвращаемого значения. Индекс первого значения – 0. См. описание возврата.

#### Возврат

Зависит от аргумента Index:

Тип	Описание
Decimal	Если аргумент Index указан. Возвращается значение с индексом Index.
Массив Decimal	Если аргумент Index не указан. Возвращается массив всех выбранных значений.

### Select

Выводит окно выбора отчета.

#### Возврат

Строка. Код нажатой клавиши в окне выбора. Может принимать одно из значений:

Значение	Описание
OK	Пользователь выбрал отчет и нажал на кнопку ОК.
CANCEL	Пользователь отказался от выбора, закрыв окно или нажав на кнопку «Отмена».

#### Замечания

При успешном выборе отчета пользователем в окне выбора в объекте сохраняются ID выбранного отчета вместо предыдущего.

## 11.13 Объект DObjectCollectionSelection

При помощи этого объекта осуществляется взаимодействие с окном выбора подборки объектов.

### 11.13.1 Свойства объекта DObjectCollectionSelection

#### Title

Строка. Текст заголовка окна выбора. Если содержит пустую строку, то текст заголовка формируется автоматически.

### 11.13.2 Методы объекта DObjectCollectionSelection

#### ID

Возвращает последние выбранные значения ID подборки объектов.

#### Аргументы

##### *Index*

Long. Не обязательный. Индекс возвращаемого значения. Индекс первого значения – 0. См. описание возврата.

#### Возврат

Зависит от аргумента Index:

Тип	Описание
Decimal	Если аргумент Index указан. Возвращается значение с индексом Index.
Массив Decimal	Если аргумент Index не указан. Возвращается массив всех выбранных значений.

## Select

Выводит окно выбора подборки объектов.

### Аргументы

#### *MultiSelect*

Boolean. Не обязательный. Если аргумент указан и содержит истину, то пользователь в окне выбора может выбрать сразу несколько строк. Иначе допускается выбор только одной строки.

### Возврат

Строка. Код нажатой клавиши в окне выбора. Может принимать одно из значений:

Значение	Описание
OK	Пользователь выбрал одно или несколько значений и нажал на кнопку ОК.
CANCEL	Пользователь отказался от выбора, закрыв окно или нажав на кнопку «Отмена».

### Замечания

При успешном выборе строк пользователем в окне выбора в объекте сохраняются соответствующие ID выбранных данных вместо предыдущих. Если было выбрано сразу несколько значений, то порядок их следования в массиве не определен.

## 12 Работа со структурой данных

### 12.1 Объект *DSTRModelService*

Объект *DSTRModelService* является сервисным объектом для операций со структурой данных.

#### 12.1.1 Методы объекта *DSTRModelService*

##### **Reset**

Очищает содержимое внутреннего буфера.

##### **Замечания**

Метод целиком очищает буфер от всей содержащейся в нем информации без записи ее в БД. Все объекты, которые были возвращены методами сервиса, теряют работоспособность. Обращение к их свойствам или методам будет вызывать возникновение ошибки.

##### **Update**

Записывает в БД все изменения из внутреннего буфера с момента предыдущего успешного вызова метода *Update* или *Reset*, или с момента создания объекта.

##### **Замечания**

Все изменения, содержащиеся во внутреннем буфере, могут быть записаны только целиком, частичное обновление невозможно. Если в процессе записи измененных данных произошла какая-либо ошибка, то ни одно изменение не попадает в БД. После успешного обновления БД, для всех измененных данных снимается признак изменения, но сами данные остаются во внутреннем буфере. Если в процессе обновления БД возникли ошибки, то признаки изменения для всех данных остаются прежними.

Если во внутренний буфер не вносились какие-либо изменения, то никаких действий не производится.

##### **GetLinkType**

Возвращает тип связи объектов.

##### **Аргументы**

*LinkTypeID*

Long. ID типа связи объектов.

##### **Возврат**

*DSTRLinkType*. Требуемый тип связи объектов. Если тип связи не найден, то возвращается пустой объект.

##### **Замечания**

Метод ищет информацию по указанному типу связи во внутреннем буфере. Если информация не найдена, то она загружается из БД и помещается во внутренний буфер.

##### **GetAttribGroup**

Возвращает группу атрибутов.

##### **Аргументы**

*AttribGroupID*

Decimal. ID группы атрибутов.

##### **Возврат**

*DSTRAttribGroup*. Требуемая группа атрибутов. Если группа не найдена, то возвращается пустой объект.

##### **Замечания**

Метод ищет информацию по указанной группе атрибутов во внутреннем буфере. Если информация не найдена, то она загружается из БД и помещается во внутренний буфер.

##### **GetAttrib**

Возвращает атрибут.

#### Аргументы

*AttribID*

Decimal. ID атрибута.

#### Возврат

DSTRAttrib. Требуемый атрибут. Если атрибут не найден, то возвращается пустой объект.

#### Замечания

Метод ищет информацию по указанному атрибуту во внутреннем буфере. Если информация не найдена, то она загружается из БД и помещается во внутренний буфер.

## CreateAttrib

Создает во внутреннем буфере новый атрибут.

#### Аргументы

*Data Type*

Long. Тип данных атрибута. Может принимать одно из значений:

Значение	Тип
83	Строка
78	Число
84	Дата/Время

*ForProject*

Boolean. Не обязательный. Если указан и содержит истину, то атрибут будет привязываться к проекту (атрибут проекта).

*MultiValue*

Boolean. Не обязательный. Если указан и содержит истину, то атрибут может иметь несколько значений (множественный атрибут).

#### Возврат

DSTRAttrib. Созданный атрибут.

#### Замечания

Метод создает во внутреннем буфере новый атрибут.

Для записи созданных объектов в БД необходимо вызвать метод Update.

## GetObjectType

Возвращает тип объекта.

#### Аргументы

*ObjectTypeID*

Decimal. ID типа объекта.

#### Возврат

DSTRObjType. Требуемый тип объекта. Если тип объекта не найден, то возвращается пустой объект.

#### Замечания

Метод ищет информацию по указанному типу объекта во внутреннем буфере. Если информация не найдена, то она загружается из БД и помещается во внутренний буфер.

## CreateObjectType

Создает во внутреннем буфере новый тип объекта.

#### Аргументы

*Class*

Long. Не обязательный. Вид объекта. Может принимать одно из значений:

Значение	Тип
----------	-----

80	Изделие (используется по умолчанию)
68	Документ
78	НКД
82	Проч. Док.

### Возврат

DSTRObjectType. Созданный тип объекта.

### Замечания

Метод создает во внутреннем буфере новый тип объекта.

Для записи созданных объектов в БД необходимо вызвать метод Update.

## GetDocumentType

Возвращает тип документа.

### Аргументы

*DocumentTypeID*

Long. ID типа документа.

### Возврат

DSTRDocumentType. Требуемый тип документа. Если тип документа не найден, то возвращается пустой объект.

### Замечания

Метод ищет информацию по указанному типу документа во внутреннем буфере. Если информация не найдена, то она загружается из БД и помещается во внутренний буфер.

## GetStorage

Возвращает хранилище документов.

### Аргументы

*StorageID*

Long. ID хранилища документов.

### Возврат

DSTRStorage. Требуемое хранилище документов. Если хранилище документов не найдено, то возвращается пустой объект.

### Замечания

Метод ищет информацию по указанному хранилищу документов во внутреннем буфере. Если информация не найдена, то она загружается из БД и помещается во внутренний буфер.

## GetApplication

Возвращает приложение.

### Аргументы

*ApplicationID*

Long. ID приложения.

### Возврат

DSTRApplication. Требуемое приложение. Если приложение документов не найдено, то возвращается пустой объект.

### Замечания

Метод ищет информацию по указанному приложению во внутреннем буфере. Если информация не найдена, то она загружается из БД и помещается во внутренний буфер.

## 12.2 Объект DSTRLinkType

Объект DSTRLinkType содержит информацию о типе связи объектов.

## 12.2.1 Свойства объекта **DSTRLinkType**

### Description

Строка. Описание типа связи.

### Parent

DSTRModelService. Только для чтения. Сервисный объект.

### ID

Long. Только для чтения. ID типа связи.

### Type

Long. Только для чтения. Тип связи. Может принимать одно из значений:

Значение	Описание
84	Подчиненная
69	Равнозначная
86	Старые варианты
80	Исполнения
68	Направленная
75	Календарь

## 12.3 Объект **DSTRAttribGroup**

Объект DSTRAttribGroup содержит информацию о группе атрибутов.

### 12.3.1 Свойства объекта **DSTRAttribGroup**

#### Description

Строка. Описание группы атрибутов.

#### Parent

DSTRModelService. Только для чтения. Сервисный объект.

#### ID

Decimal. Только для чтения. ID группы атрибутов.

## 12.4 Объект **DSTRAttrib**

Объект DSTRAttrib содержит информацию о атрибуте.

### 12.4.1 Свойства объекта **DSTRAttrib**

#### Description

Строка. Описание атрибута.

#### Parent

DSTRModelService. Только для чтения. Сервисный объект.

#### ID

Decimal. Только для чтения. ID атрибута.

## DataType

Long. Только для чтения. Тип данных атрибута. Может принимать одно из значений:

Значение	Тип
83	Строка
78	Число
84	Дата и время

## MultiValue

Boolean. Только для чтения. Содержит истину, если атрибут может иметь несколько значений (множественный атрибут).

## ForProject

Boolean. Только для чтения. Содержит истину, если атрибут может привязываться к проекту (атрибут проекта).

## Group

DSTRAttribGroup. Группа атрибутов.

### Замечания

В свойство можно записывать непосредственно ID группы атрибутов.

## EditMask

Строка. Формат атрибута.

## EditType

Long. Способ редактирования атрибута. Может принимать одно из значений:

Значение	Тип
69	Свободный
87	Список
88	Классификатор
80	Внешнее

## History

Boolean. Содержит истину, если сохраняется история значений атрибута.

## 12.5 Объект *DSTRObjectType*

Объект DSTRObjectType содержит информацию о типе объекта.

### 12.5.1 Свойства объекта DSTRObjectType

#### Description

Строка. Описание типа объекта.

#### Parent

DSTRModelService. Только для чтения. Сервисный объект.

#### ID

Decimal. Только для чтения. ID типа объекта.



## Class

Long. Только для чтения. Вид объекта. Может принимать одно из значений:

Значение	Тип
80	Изделие (используется по умолчанию)
68	Документ
78	НКД
82	Проч. Док.

## IconPath

Строка. Имя пиктограммы.

## SortKey

Long. Ключ сортировки.

## Mnemo

Строка. Мнемоническое обозначение типа объекта.

## 12.5.2 Методы объекта DSTRObjecType

### GetAttribs

Возвращает информацию о связанных с типом объекта атрибутах.

#### Возврат

DDataSet. Коллекция, содержащая объекты типа DSTRObjecTypeAttrib содержащие информацию о привязке атрибута к типу объекта. Если к типу объекта не привязано ни одного атрибута, то коллекция не содержит элементов.

#### Замечания

Метод ищет информацию по привязанным атрибутам во внутреннем буфере. Если какая-либо информация не найдена, то она загружается из БД и помещается во внутренний буфер.

### AddAttrib

Привязывает к типу объекта атрибут.

#### Аргументы

*Attribute*

DSTRAtrib или Decimal. Добавляемый атрибут или его ID.

#### Возврат

DSTRObjecTypeAttrib. Объект, содержащий информацию о привязке атрибута к типу объекта.

#### Замечания

Метод добавляет информацию о привязке атрибута к типу объекта во внутреннем буфере.

Для записи изменений в БД необходимо вызвать метод Update соответствующего объекта типа DSTRModelService.

### GetObjectTypes

Возвращает информацию о дочерних типах объектов, связанных с текущим типом объекта.

#### Возврат

DDataSet. Коллекция, содержащая объекты типа DSTRObjecTypeObjecType содержащие информацию о привязке типа объекта к текущему типу объекта. Если к типу объекта не привязано ни одного дочернего типа объекта, то коллекция не содержит элементов.

#### Замечания

Метод ищет информацию по привязанным типам объекта во внутреннем буфере. Если какая-либо информация не найдена, то она загружается из БД и помещается во внутренний буфер.

## AddObjectType

Привязывает к типу объекта дочерний тип объекта.

### Аргументы

#### *ObjectType*

DSTRObjecType или Decimal. Добавляемый тип объекта или его ID.

#### *LinkType*

DSTRLinkType или Decimal. Не обязательный. Используемый тип связи или его ID. Если не указан, то используется тип связи с ID = 1.

### Возврат

DSTRObjecTypeObjecType. Объект, содержащий информацию о привязке к типу объекта дочернего типа объекта.

### Замечания

Метод добавляет информацию о привязке к типу объекта дочернего типа объекта во внутреннем буфере.

Для записи изменений в БД необходимо вызвать метод Update соответствующего объекта типа DSTRModelService.

## 12.6 Объект DSTRObjecTypeAttrib

Объект DSTRObjecTypeAttrib содержит информацию о привязке атрибута к типу объекта.

### 12.6.1 Свойства объекта DSTRObjecTypeAttrib

#### Parent

DSTRObjecType. Только для чтения. Тип объекта, к которому привязан атрибут.

#### Attrib

DSTRAttrib. Только для чтения. Атрибут.

#### Required

Boolean. содержит истину, если атрибут является обязательным для объектов данного типа.

#### Value

Тип данных зависит от атрибута. Содержит значение по-умолчанию. Если значение не установлено, то содержит Null.

## 12.7 Объект DSTRObjecTypeObjecType

Объект DSTRObjecTypeObjecType содержит информацию о привязке к типу объекта дочернего типа объекта.

### 12.7.1 Свойства объекта DSTRObjecTypeObjecType

#### Parent

DSTRObjecType. Только для чтения. Тип объекта, к которому привязан дочерний тип объекта.

#### ObjecType

DSTRObjecType. Только для чтения. Дочерний тип объекта.

## LinkType

DSTRLinkType. Только для чтения. Тип связи.

## Required

Boolean. содержит истину, если объект дочернего тип является обязательным для объектов данного типа.

## 12.8 Объект *DSTRDocumentType*

Объект DSTRDocumentType содержит информацию о типе документа.

### 12.8.1 Свойства объекта DSTRDocumentType

#### Parent

DSTRModelService. Только для чтения. Сервисный объект.

#### ID

Long. Только для чтения. ID типа документа.

#### Description

Строка. Только для чтения. Описание типа документа.

#### Note

Строка. Только для чтения. Примечания для типа документа.

#### MaxVer

Long. Только для чтения. Максимальный номер версии для документов данного типа.

## 12.9 Объект *DSTRStorage*

Объект DSTRStorage содержит информацию о хранилище документов архива.

### 12.9.1 Свойства объекта DSTRStorage

#### Parent

DSTRModelService. Только для чтения. Сервисный объект.

#### ID

Long. Только для чтения. ID хранилища.

#### Description

Строка. Только для чтения. Описание хранилища.

#### Type

Long. Только для чтения. Тип хранилища. Может содержать одно из значений:

Значение	Описание
83	Постоянный

#### Path

Строка. Только для чтения. Путь размещения хранилища.

## IsDefault

Boolean. Только для чтения. Содержит истину, если хранилище используется по умолчанию.

## AccessType

Long. Только для чтения. Тип доступа к хранилищу. Может содержать одно из значений:

Значение	Описание
87	Полный
78	Защищенный (NetWare)
84	Защищенный (Windows NT)
68	Hummingbird DM (полный)
72	Hummingbird DM (защищенный)

## VolumeLabel

Строка. Только для чтения. Метка тома.

## IsDeleted

Boolean. Только для чтения. Содержит истину, если хранилище аннулировано.

## 12.10 Объект *DSTRApplication*

Объект DSTRApplication содержит информацию о приложении.

### 12.10.1 Свойства объекта DSTRApplication

#### Parent

DSTRModelService. Только для чтения. Сервисный объект.

#### ID

Long. Только для чтения. ID приложения.

#### Description

Строка. Только для чтения. Описание приложения.

#### AppID

Строка. Только для чтения. Строковый идентификатор приложения.

#### Extension

Строка. Только для чтения. Расширение файлов по умолчанию.

#### ExtensionFilter

Строка. Только для чтения. Фильтр расширений файлов.

#### Icon

Строка. Только для чтения. Имя пиктограммы.

#### DefaultDocumentType

DSTRDocumentType. Только для чтения. Тип документа, используемый по умолчанию.

#### IsDeleted

Boolean. Только для чтения. Содержит истину, если приложение аннулировано.

## 12.10.2 Методы объекта *DSTRApplication*

### GetMethods

Возвращает информацию о методах запуска приложения.

#### Возврат

DDataset. Коллекция, содержащая объекты типа *DSTRApplicationMethod* содержащие информацию о методах запуска приложения. Если для приложения не определено ни одного метода запуска, то коллекция не содержит элементов.

#### Замечания

Метод ищет информацию по методам запуска приложения во внутреннем буфере. Если какая-либо информация не найдена, то она загружается из БД и помещается во внутренний буфер.

## 12.11 Объект *DSTRApplicationMethod*

Объект *DSTRApplicationMethod* содержит информацию о приложении.

### 12.11.1 Свойства объекта *DSTRApplicationMethod*

#### Parent

*DSTRApplication*. Только для чтения. Приложение, к которому относится метод запуска.

#### ID

Long. Только для чтения. ID приложения.

#### Description

Строка. Только для чтения. Описание приложения.

#### Type

Long. Только для чтения. Содержит тип метода. Может содержать одно из перечисленных значений:

Значение	Описание
79	Открытие документа.
86	Просмотр документа.
80	Печать документа.
78	Создание нового документа.

#### Parameters

Строка. Только для чтения. Параметры приложения.

#### Path

Строка. Только для чтения. Имя исполняемого файла для запуска приложения.

#### UseRegPath

Boolean. Только для чтения. Если содержит истину, то путь запуска приложения берется из реестра.

#### RegValue

Строка. Только для чтения. Имя значения реестра.

#### IsDefault

Boolean. Только для чтения. Содержит истину, если метод используется по умолчанию для запуска приложения.

## Integration

Long. Только для чтения. Тип интеграции. Может содержать одно из перечисленных значений:

Значение	Описание
0	Без интеграции.
79	ODMA.
83	Стандартная.

## UseCheckOut

Boolean. Только для чтения. Использовать захват документа.

## DeleteLocalFile

Boolean. Только для чтения. Удалять локальные копии после закрытия захваченного документа.

## UseDDE

Boolean. Только для чтения. Использовать DDE.

## DDEOpenApp

Строка. Только для чтения. Параметр DDE Application.

## DDEOpenTopic

Строка. Только для чтения. Параметр DDE Topic.

## DDEOpenRun

Строка. Только для чтения. Команда DDE, если приложение запущено.

## DDEOpenNotRun

Строка. Только для чтения. Команда DDE, если приложение не запущено.

## 13 Работа с отчетами

### 13.1 Объект *DReportObjectService*

Объект *DReportObjectService* является сервисным объектом для операций с пользовательскими отчетами.

#### 13.1.1 Методы объекта *DReportObjectService*

##### **GetReport**

Возвращает пользовательский отчет.

##### **Аргументы**

*ReportID*

Decimal. ID отчета.

##### **Возврат**

*DReportObject*. Требуемый отчет. Если отчет не найден, то возвращается пустой объект.

##### **GetReports**

Возвращает список дочерних пользовательских отчетов из дерева отчетов для указанного родительского отчета.

##### **Аргументы**

*ParentReportID*

Decimal. ID родительского отчета. Для получения отчетов первого уровня должен содержать 0.

##### **Возврат**

*DDataset*. Коллекция, содержащая объекты типа *DReportObject* представляющие информацию о пользовательских отчетах. Если ни один отчет не найден, то коллекция не содержит элементов.

### 13.2 Объект *DReportObject*

Объект *DReportObject* содержит информацию о пользовательском отчете.

#### 13.2.1 Свойства объекта *DReportObject*

##### **Parent**

*DReportObjectService*. Только для чтения. Соответствующий сервисный объект *DReportObjectService*.

##### **ID**

Decimal. Только для чтения. ID отчета.

##### **ParentReport**

*DReportObject*. Только для чтения. Родительский отчет в дереве отчетов. Если отчет находится на верхнем уровне, то содержит пустой объект.

##### **Description**

Строка. Только для чтения. Описание отчета.

##### **Type**

Long. Только для чтения. Тип отчета. Может принимать одно из перечисленных значений:

Значение	Описание
1	Настраиваемый отчет.

2	Встроенный отчет.
3	Предопределенный отчет: спецификация.
259	Предопределенный отчет: групповая спецификация (тип А).
4	Папка.
0	Прочие отчеты.

## LevelType

Long. Только для чтения. Глубина поиска объектов в дереве проектов. Используется только для настраиваемых отчетов, для остальных типов значение не определено. Может принимать одно из перечисленных значений:

Значение	Описание
65	Объекты из всех уровней проекта.
49	Объекты из первого уровня проекта.
68	Не используется.
79	Текущие объекты. Текущими считаются объекты, ID которых переданы через аргумент «Объекты».
66	Все объекты из БД.

## SouRoot

Long. Только для чтения. Тип отбора объектов. Используется только для настраиваемых отчетов, для остальных типов значение не определено. Может принимать одно из перечисленных значений:

Значение	Описание
0	Все объекты.
1	Отбор по условиям отчета.

## DataToReport

DReportObject. Только для чтения. Связанный отчет. Если связанный отчет не определен, то содержит пустой объект.

## DataToReportFromColumn

Строка. Только для чтения. Наименование колонки отчета, значение которой передается связанному отчету.

## 13.2.2 Методы объекта DReportObject

### Execute

Выполняет отчет.

#### Аргументы

*ObjectID*

Decimal или массив Decimal. Не обязательный. ID объектов, передаваемых в отчет в качестве аргумента.

*Arg*

Не обязательный, любое количество. Аргументы отчета в порядке их описания в отчете.

#### Возврат

DDataStore. Данные отчета.

#### Замечания

Для выполнения отчета должны быть переданы все аргументы, которые определены в отчете. Если какие-либо аргументы не указаны или тип данных переданных значений не соответствует аргументу, то возникает ошибка выполнения.



## GetColumns

Возвращает информацию о колонках отчета.

### Возврат

DDataSet. Коллекция, содержащая объекты типа DReportColumn содержащие информацию о колонках отчета.

## GetArguments

Возвращает информацию об аргументах отчета.

### Возврат

DDataSet. Коллекция, содержащая объекты типа DSTObjectTypeObjectTypes содержащие информацию об аргументах отчета.

## 13.3 Объект DReportColumn

Объект DReportColumn содержит информацию о колонке отчета.

### 13.3.1 Свойства объекта DReportColumn

#### Name

Строка. Только для чтения. Имя колонки.

#### Type

Строка. Только для чтения. Тип колонки. Может содержать одно из значений:

Значение	Описание
column	Колонка с данными.
compute	Вычисляемое поле.

#### Description

Строка. Только для чтения. Описание колонки.

#### ColType

Строка. Только для чтения. Тип данных колонки. Может содержать одно из значений:

Значение	Описание
char(n)	Строка. n – число символов. Для колонки с данными.
char	Строка. Для вычисляемых полей.
date	Дата.
datetime	Дата и время.
long	Целое число.
decimal(n)	Число с фиксированной точкой. n – число десятичных разрядов после точки.
number	Число с плавающей точкой.
time	Время.

#### Band

Строка. Только для чтения. Расположение колонки. Может содержать одно из значений:

Значение	Описание
detail	Область данных.
footer	Область подвала отчета.
summary	Область итогов.

header	Заголовок отчета.
header.n	Заголовок группы с номером n.
trailer.n	Область итогов группы с номером n.

## 13.4 Объект *DReportArgument*

Объект *DReportArgument* содержит информацию об аргументе отчета.

### 13.4.1 Свойства объекта *DReportArgument*

#### Name

Строка. Только для чтения. Имя аргумента.

#### Description

Строка. Только для чтения. Описание аргумента

#### Type

Строка. Только для чтения. Тип данных аргумента. Может содержать одно из значений:

Значение	Описание
string	Строка.
date	Дата.
datetime	Дата и время.
number	Число.
time	Время.

#### Format

Строка. Только для чтения. Формат представления аргумента.

#### Initial

Строка. Только для чтения. Значение по умолчанию.

## 14 Подключаемые модули

Для расширения возможностей системы Lotsia PDM PLUS могут использоваться подключаемые модули. Для выполнения некоторых действий система Lotsia PDM PLUS вызывает определенные функции из этих модулей. Подключаемые модули могут содержать функции для обработки выделенных в Lotsia PDM PLUS объектов, для ввода и редактирование значений атрибутов и для выполнения отдельных программ.

Подключаемый модуль представляет собой COM-объект. Все модули должны быть зарегистрированы в ветке 'HKLM\SOFTWARE\LotsiaSoft\PartY\Plugins' реестра. Каждый ключ этой ветки описывает отдельный модуль. Имя ключа является программным идентификатором (ProgID) или класса (CLSID), позволяющий создать объект с интерфейсом DPluginModule. В значении по умолчанию может содержаться имя модуля. При запуске Lotsia PDM PLUS пытается загрузить каждый модуль и получить его описание при помощи вызова метода 'GetPluginInfo'. Информация успешно получена, то модуль остается загруженными до окончания работы системы Lotsia PDM PLUS. В случае возникновения ошибки модуль не будет использоваться. Информация об ошибке будет записана в файл журнала.

Функции подключаемых модулей могут быть одного из 3-х типов: RUN, RUNOBJECT и EDITVALUE.

Функции типа RUN и RUNOBJECT, не требующие наличие выделенного объекта, используются для выполнения каких-либо действий. Они могут быть вызваны из пункта «Выполнить внешний модуль» главного меню. При выполнении настройки главного меню в него могут быть добавлены отдельные пункты с указанием выполняемой функции.

Функции типа RUNOBJECT используются для выполнения каких-либо действий с объектами, выделенными в окне. Они могут быть вызваны из пункта «Все внешние расширения» меню «Объект» из окон обработки объектов (Дерево проекта, Подборка и т.п.). При выполнении настройки меню для типов объектов в него могут быть добавлены отдельные пункты с указанием выполняемой функции.

Функции типа EDITVALUE используются для ввода значения атрибута с типом редактирования «Внешний». Для каждого такого атрибута может быть указана своя внешняя функция редактирования значения. Одна функция может быть использована для редактирования значений нескольких атрибутов.

Функции подключаемых модулей могут обращаться к объекту Application, для выполнения нужных действий.

### 14.1 Объект DPluginModule

Объект DPluginModule является базовым объектом подключаемого модуля, для взаимодействия с ним системы Lotsia PDM PLUS. Этот объект создается подключаемым модулем при его инициализации.

#### 14.1.1 Методы объекта DPluginModule

##### GetPluginInfo

Вызывается при загрузке подключаемого модуля для его инициализации. Этот метод должен заполнить объект PluginInfo информацией о функциях подключаемого модуля.

##### Аргументы

*PluginInfo*

DPluginInfo. Объект параметров подключаемого модуля.

##### Замечания

Этот метод вызывается до окончания полной загрузки системы Lotsia PDM PLUS, часть сервисов в это момент могут быть недоступны.

## Run

Вызывается для выполнения функции типа RUN.

### Аргументы

*ID*

Строка. Идентификатор функции.

*Hwnd*

HWND. Handle основного окна приложения.

## RunObject

Вызывается для обработки выделенных объектов.

### Аргументы

*ID*

Строка. Идентификатор функции.

*Parm*

DPluginRunObjectParm. Содержит список выделенных объектов и дополнительные параметры.

## EditValue

Вызывается для ввода или редактирования значения атрибута.

### Аргументы

*ID*

Строка. Идентификатор функции.

*Parm*

DPluginEditValueParm. Параметры вызова.

### Возврат

Boolean. Истина если редактирование закончено успешно. Ложь если пользователь отказался от редактирования.

## 14.2 Объект DPluginInfo

Объект DPluginInfo содержит параметры подключаемого модуля.

### 14.2.1 Методы объекта DPluginInfo

#### SetInfo

Устанавливает описание о подключаемом модуле.

##### Аргументы

*Name*

Строка. Описание модуля.

*Version*

Строка. Версия модуля.

*Copyright*

Строка. Авторские права на модуль.

#### AddFunctionRun

Добавляет описание функции типа RUN.

##### Аргументы

*ID*

Строка. ID функции. Это значение передается в качестве аргумента ID методу Run при вызове функции.

*Description*

Строка. Описание функции.

## AddFunctionRunObject

Добавляет описание функции типа RUNOBJECT.

### Аргументы

*ID*

Строка. ID функции. Это значение передается в качестве аргумента ID методу RunObject при вызове функции.

*Description*

Строка. Описание функции.

*ObjectRequired*

Boolean. Содержит истину, если для выполнения функции необходимо наличие хотя бы одного выделенного объекта.

## AddFunctionEditValue

Добавляет описание функции типа EDITVALUE.

### Аргументы

*ID*

Строка. ID функции. Это значение передается в качестве аргумента ID методу EditValue при вызове функции.

*Description*

Строка. Описание функции.

*Data Type*

Long. Тип данных значения, которое может редактировать функция. Может принимать одно из значений:

Значение	Тип
83	Строка
78	Число
84	Дата и время

## 14.3 Объект DPluginRunObjectParm

Объект DPluginRunObjectParm содержит параметры при вызове функции подключаемого модуля типа RUNOBJECT.

### 14.3.1 Свойства объекта DpluginRunObjectParm

#### LinkTypeID

Long. Только для чтения. ID текущего типа связи, если в момент вызова функции он был известен, иначе поле содержит 0. ID типа связи передается при вызове функции из окна "Дерево проектов", "История проектов" или "Представление проектов". При вызове функции из других окон текущий тип связи неизвестен.

#### Hwnd

Long. Только для чтения. Handle активного окна приложения.

#### Count

Long. Только для чтения. Количество обрабатываемых объектов.

## 14.3.2 Методы объекта DPluginRunObjectParm

### ObjectID

Возвращает ID выделенных объектов, переданных для обработки функции.

#### Аргументы

##### *Index*

Long. Не обязательный. Индекс возвращаемого значения. Индекс первого значения – 0. Общее число элементов содержится в свойстве Count. См. описание возврата.

#### Возврат

Зависит от аргумента Index:

Тип	Описание
Decimal	Если аргумент Index указан. Возвращается значение с индексом Index.
Массив Decimal	Если аргумент Index не указан. Возвращается массив всех значений.

### TLinkID

Возвращает ID связей выделенных объектов с родительскими объектами, переданных для обработки функции.

#### Аргументы

##### *Index*

Long. Не обязательный. Индекс возвращаемого значения. Индекс первого значения – 0. Общее число элементов содержится в свойстве Count. См. описание возврата.

#### Возврат

Зависит от аргумента Index:

Тип	Описание
Decimal	Если аргумент Index указан. Возвращается значение с индексом Index.
Массив Decimal	Если аргумент Index не указан. Возвращается массив всех значений.

#### Замечания

Если при вызове функции ID связи не был известен, то будут возвращаться нулевые значения.

## 14.4 Объект DPluginEditValueParm

Объект DPluginEditValueParm содержит параметры при вызове функции подключаемого модуля типа EDITVALUE.

### 14.4.1 Свойства объекта DPluginEditValueParm

#### Value

Тип данных зависит от атрибута. Изменяемое значение атрибута.

#### AttribID

Decimal. Только для чтения. ID атрибута, значение которого редактируется.

#### ObjectTypeID

Decimal. Только для чтения. ID типа объекта, для которого изменяется значение атрибута. Если объект не определен, то содержит 0.

#### ObjectID

Decimal. Только для чтения. ID объекта, для которого изменяется значение атрибута. Если объект не определен, то содержит 0.

## TLinkID

Decimal. Только для чтения. ID связи объекта, для которого изменяется значение атрибута, с родительским объектом. Если объект не определен или связь неизвестна, то содержит 0.

## Hwnd

Long. Только для чтения. Handle активного окна приложения.

## 14.4.2 Методы объекта DPluginEditValueParm

### SetEditType

Устанавливает требуемый тип редактирования значения атрибута. Используется, если значение необходимо ввести методами системы Lotsia PDM PLUS.

#### Аргументы

*EditType*

Long. Требуемый способ редактирования значения. Может принимать одно из значений:

Значение	Тип
69	Свободный
87	Список
70	Выбор файла

### AddValue

Добавляет значения для других атрибутов, которые должны быть изменены для объекта одновременно с основным значением. Если объект не определен, то дополнительные значения игнорируются.

#### Аргументы

*AttribID*

Decimal. ID атрибута.

*Value*

Тип данных зависит от атрибута. Значение атрибута.

## 15 Примеры скриптов

Ниже, в данном разделе приводятся примеры скриптов, которые могут быть полезны в некоторых ситуациях.

### 15.1 Обращение к базе данных: выполнение SQL-запроса

Необходимо получить информацию о наличии в объекте документа архива и отобразить полученную информацию в колонке отчета.

Отчет:

Object	Наличие документа архива
col2	col4

N	Описание	Колонка	Уровень	Атрибут	Тип
1	Описание объекта (Потомок)	col2	Потомок		
2	Объект (код)	isobject_id	Системный		
3	Описание объекта (Потомок)	col4	Потомок		

Колонка col4 – фиктивная. В нее будет выводиться информация о наличии в объекте документа архива.

Вкладка «Скрипт»:

Событие	Функция	Назначить
После заполнения данных	MainScript	

Global (Основной скрипт)

Proc MainScript()

Con\_ExecSQL

Proc Con(Arg1, Arg2, Arg3, Arg4)

Proc ExecSQL(Arg1, Arg2, Arg3, Arg4, Arg5)

Добавить...

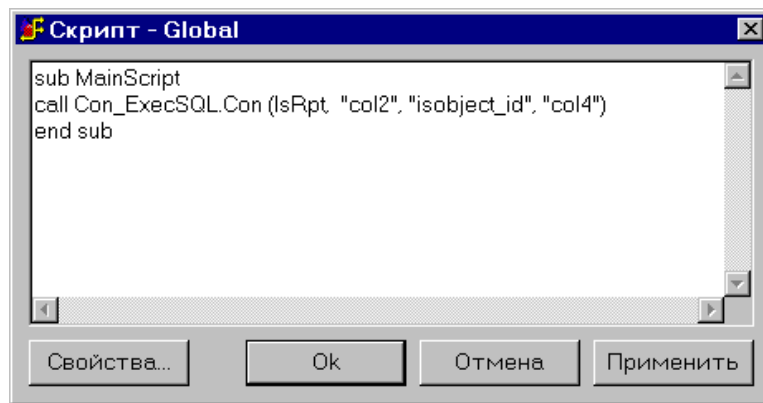
Создать...

Удалить

Скрипт...

Главный скрипт:

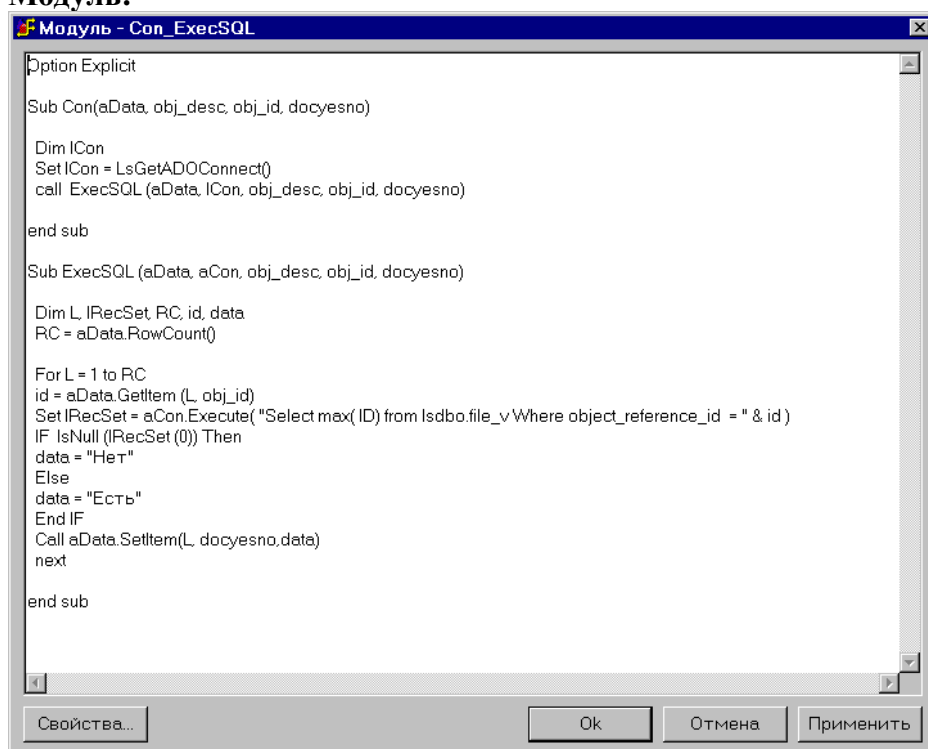




Текст главного скрипта:

```
sub MainScript  
    call Con_ExecSQL.Con (lsRpt, "col2", "isobject_id", "col4")  
end sub
```

## Модуль:



## Текст модуля:

```
Option Explicit

Sub Con(aData, obj_desc, obj_id, docyesno)
    Dim lCon
    Set lCon = Application.GetADOConnect()
    call ExecSQL (aData, lCon, obj_desc, obj_id, docyesno)
end sub

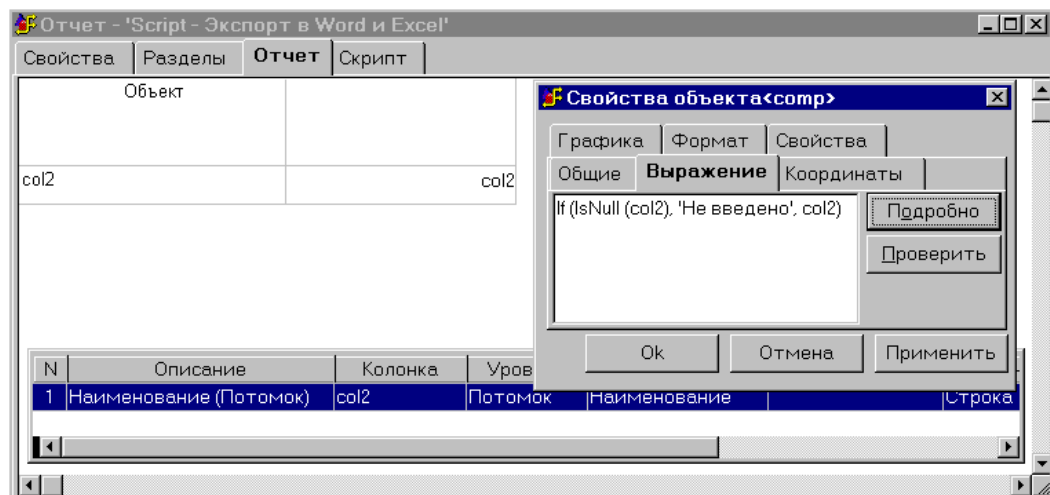
Sub ExecSQL (aData, aCon, obj_desc, obj_id, docyesno)
    Dim L, lRecSet, RC, id, data
    RC = aData.RowCount()

    For L = 1 to RC
        id = aData.GetItem (L, obj_id)
        Set lRecSet = aCon.Execute( "Select max( ID) from lsdbo.file_v Where object_reference_id = " & id )
        IF IsNull(lRecSet(0)) Then
            data = "Нет"
        Else
            data = "Есть"
        End IF
        Call aData.SetItem(L, docyesno,data)
    next
end sub
```

## 15.2 Выгрузка отчетов в приложения MS Word и MS Excel

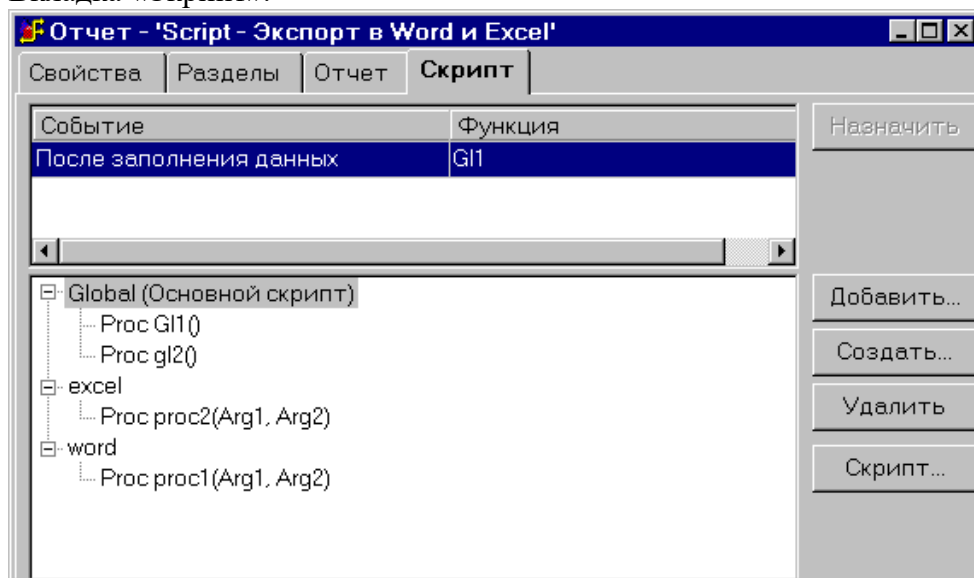
Необходимо выгрузить информацию из формы отчета в приложение MS Word или MS Excel.

Отчет:

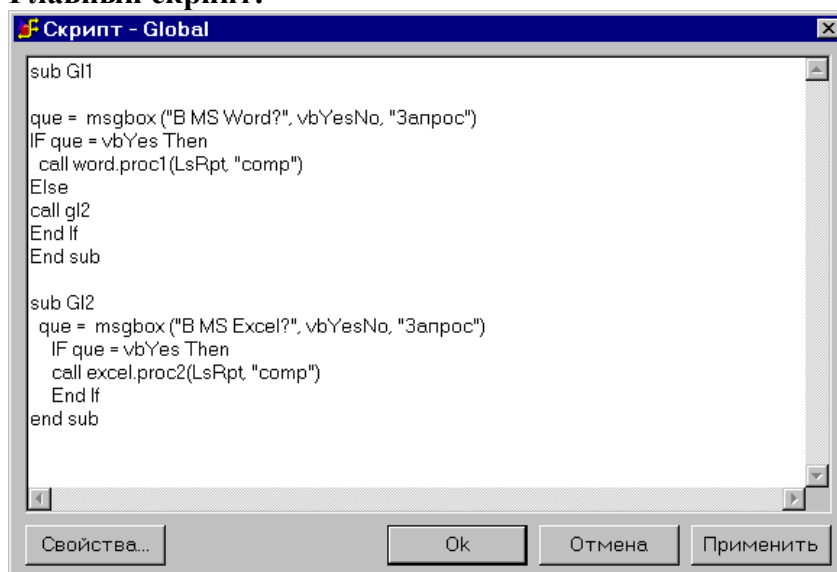


Здесь всего одна колонка. Вторая колонка – вычисляемое поле. Его содержание приведено. Оно нужно для того, чтобы в колонке не было Null-значений.

Вкладка «Скрипт»:



Главный скрипт:

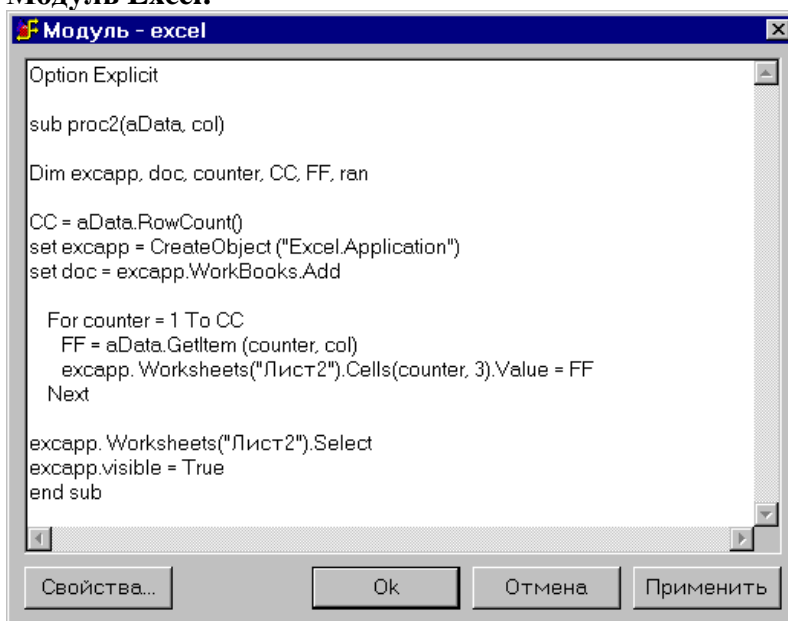


### Текст главного скрипта:

```
sub G12
    que = msgbox ("B MS Excel?", vbYesNo, "Запрос")
    IF que = vbYes Then
        call excel.proc2(LsRpt, "comp")
    Else
        call g11
    End If
end sub

sub G11
    que = msgbox ("B MS Word?", vbYesNo, "Запрос")
    IF que = vbYes Then
        call word.proc1(LsRpt, "comp")
    End If
end sub
```

### Модуль Excel:



### Текст модуля Excel:

```
Option Explicit

sub proc2(aData, col)
    Dim excapp, doc, counter, CC, FF, ran

    CC = aData.RowCount()
    set excapp = CreateObject ("Excel.Application")
    set doc = excapp.WorkBooks.Add

    For counter = 1 To CC
        FF = aData.GetItem (counter, col)
        excapp.Worksheets("Лист2").Cells(counter, 3).Value = FF
    Next

    excapp.Worksheets("Лист2").Select
    excapp.visible = True
end sub
```

### Модуль Word:



Текст модуля Word:

```
Option Explicit

sub proc1(aData, col)
    Dim wrdapp, CC, TT, J, doc, ran, tabl, aCell

    set wrdapp = CreateObject("Word.Application")
    set doc = wrdapp.Documents.Add

    CC = aData.RowCount()

    set ran = wrdapp.Selection.Range
    set tabl = wrdapp.ActiveDocument.Tables.Add(ran, CC, 2)

    J = 1
    For Each aCell In tabl.Columns(1).Cells
        aCell.Select
        TT = aData.GetItem(J, col)
        wrdapp.Selection.TypeText TT
        J = J+1
        IF J > CC then
            exit for
        end if
    Next
    wrdapp.visible = True

    call gl2
end sub
```

## 15.3 Экспорт отчета по шаблону

Текст модуля:

```
sub OE1
    iRc = LsRpt.OleExport ("\\Server\Mail\Template.doc", "C:\_Report.doc", "SP")
```

```
' или для экспорта в версию документа архива iRc = LsRpt.OleExport ( "\\Server\Mail\Template.doc" ,
"100004309300098", "SP" )

if iRc = 1 then
    MsgBox "Экспорт отчета выполнен успешно", 0, "Скрипт"
else
    MsgBox "При экспорте отчета произошли ошибки", 16, "Скрипт"
end if
end sub
```

## 15.4 Сцепление строк через указанный разделитель

Задача: необходимо произвести в области группировки сцепление строк из колонок отчета (в нашем примере - col7 и col8) через указанный разделитель.

Текст главного скрипта:

```
sub Main
    call ConString.Con(LsRpt, "col7", "col8")
end sub
```

Текст модуля ConString:

```
Option Explicit

sub Con(aData, mst, mst_list)
    Dim RC, L, M, M_list, l2, l1

    RC = aData.RowCount()
    M_list = ""
    l1 = 1
    For L = 1 to RC
        if L>1 and L = aData.FindGroupChange(L, 1) then
            for l2=l1 to l1-1
                Call aData.SetItem(l2, mst_list, M_list)
            next
            M_list = ""
            l1 = 1
        end if

        M = aData.GetItem (L, mst)

        if M_list<>"" then
            M_list = M_list + ", "
        end if
        M_list = M_list + M
    next

    if M_list<>"" then
        for l2=l1 to RC
            Call aData.SetItem(l2, mst_list, M_list)
        next
    end if
end sub
```

## 15.5 Получение данных из другой СУБД в картах работ

Для получения данных из другой СУБД необходимо в шаблоне работы для каждого поля результата запроса создать переменную-массив с соответствующим типом данных. Затем нужно создать действие типа «Скрипт», пример которого приведен ниже.

В примере производится считывание списка всех счетов из БД Координатора за указанную дату. Дата берется из переменной шаблона 'aDate'. Результат помещается в переменные-массивы карты 'aID', 'aNumber', 'aState' и 'aSum'. Хотя поле 'ID' таблицы счетов имеет тип данных datetime, значения этого поля в запросе преобразовываются в строковый вид и сохраняются в строковой переменной карты 'aID', т.к. в скриптах точность типа данных Date недостаточная.

Предполагается, что это действие может быть вызвано несколько раз, например, по кнопке из формы задачи. Для этого, с целью оптимизации в глобальной переменной скрипта, сохраняется подготовленный ADO-объект, содержащий команду и подключенный к БД.

```
Option Explicit

Dim gSelAcc ' Запрос к БД

sub AccountRetrieve
    Dim lRow, lRecSet, aEmpty(), lCon, lCmd

    if not IsObject(gSelAcc) then
        ' Подключиться к БД Координатора, если еще не подключены
        Set lCon = CreateObject("ADODB.Connection")
        Call lCon.Open("Provider=MSDASQL;Dsn=Trade_TestSQL10;uid=mainadmin;pwd=main")

        ' Сформировать запрос
        Set lCmd = CreateObject("ADODB.Command")
        lCmd.CommandText = "select Cast(id as varchar), number2, state, AccountSum " + _
            "from lsdbo.Account where datedoc = ?"
        lCmd.CommandType = 1 'adCmdText
        lCmd.Prepared = True
        lCmd.Parameters.Append lCmd.CreateParameter("aDate", 7) 'adDate
        lCmd.ActiveConnection = lCon

        Set gSelAcc = lCmd
    end if

    ' Выполнить запрос к БД по указанной дате
    gSelAcc("aDate") = LsJob.GetVarValue("aDate")
    Set lRecSet = gSelAcc.Execute

    ' Очистить переменные
    LsJob.SetVarValue "aID", aEmpty
    LsJob.SetVarValue "aNumber", aEmpty
    LsJob.SetVarValue "aState", aEmpty
    LsJob.SetVarValue "aSum", aEmpty

    ' Считать данные по всем счетам и занести их в массивы построчно
    lRow = 0
    Do While Not lRecSet.EOF
        LsJob.SetVarValue "aID", lRecSet(0).Value
        LsJob.SetVarValue "aNumber", lRecSet(1).Value
        LsJob.SetVarValue "aState", lRecSet(2).Value
        LsJob.SetVarValue "aSum", lRecSet(3).Value

        lRow = lRow + 1
        lRecSet.MoveNext
    Loop
end sub
```

Если необходимо выполнить запрос из текущей БД, то для подключения к БД необходимо использовать метод GetADOConnect объекта Application:

```
. . .

if not IsObject(gSelAcc) then
    ' Подключиться к БД, если еще не подключены
    Set lCon = Application.GetADOConnect()

    ' Сформировать запрос

. . .
```

Если запрос возвращает большое количество строк, то установка значений переменных задачи по одному значению в цикле может существенно замедлить выполнение скрипта. Для уменьшения накладных расходов выгоднее изменять значение переменной-массива за один вызов. Ниже приведен модифицированный фрагмент примера, который собирает результаты запроса в локальные массивы, а затем обновляет ими переменные задачи. Поскольку VBScript не поддерживает динамическое расширение

массивов, то массивы расширяются вручную перед чтением очередных 100 строк результата.

```
. . .

' Выполнить запрос к БД
gSelAcc("aDate") = LsJob.GetVarValue("aDate")
Set lRecSet = gSelAcc.Execute

' Считать целиком результат запроса
lRow = 0
Do While Not lRecSet.EOF
    if lRow mod 100 = 0 then
        ' Увеличить размер массивов для хранения результата
        ReDim Preserve aID((lRow / 100 + 1) * 100)
        ReDim Preserve aNumber((lRow / 100 + 1) * 100)
        ReDim Preserve aState((lRow / 100 + 1) * 100)
        ReDim Preserve aSum((lRow / 100 + 1) * 100)
    end if

    aID(lRow) = lRecSet(0).Value
    aNumber(lRow) = lRecSet(1).Value
    aState(lRow) = lRecSet(2).Value
    aSum(lRow) = lRecSet(3).Value

    lRow = lRow + 1
    lRecSet.MoveNext
Loop

if lRow > 0 then
    ' Скорректировать размер массивов в соответствии с числом считанных строк
    ReDim Preserve aID(lRow - 1)
    ReDim Preserve aNumber(lRow - 1)
    ReDim Preserve aState(lRow - 1)
    ReDim Preserve aSum(lRow - 1)
end if

' Записать массивы в переменные работы
LsJob.SetVarValue "aID", aID
LsJob.SetVarValue "aNumber", aNumber
LsJob.SetVarValue "aState", aState
LsJob.SetVarValue "aSum", aSum
end sub
```

## 15.6 Наследование значения атрибутов в дереве проекта.

В приведенном ниже примере производится обработка всех объектов дерева проекта. При этом при обходе дерева обрабатываются дочерние объекты только родительских объектов определенного типа. Всем найденным объектам подходящего типа устанавливается одно и то же значение определенного атрибута. Если указано значение второго атрибута, то оно добавляется у этих же объектов. Обработка объектов в дереве производится рекурсивным вызовом процедуры `InheritAttrib`.

В примере используются числовые идентификатор (ID) из примера настройки БД «Замок (пример из документации)». В качестве корневого объекта используется объект «Замок». Дочерние объекты ищутся у родительских объектов типа «Башня», «Здание» и «Стена». Для объектов типа «Эскиз» и «Проект» устанавливается значение атрибута «Дата окончания разработки документации» и добавляется значение атрибута «Файл».

Пример должен запускаться из действия над объектами. Для запуска необходимо указывать процедуру `Run`.

```
sub Run
    ' Выполнить изменение атрибутов
    RunAttrib LsOperation.ObjectService
end sub

function RunAttrib(aObjSvc)
    Dim oRoot, lDate

    ' Установить корневой объект
```



```

set oRoot = aObjSvc.GetObject(793987520900000) ' Замок
if not IsObject(oRoot) then
    MsgBox "Объект не найден"
    RunAttrib = false
    exit function
end if

lDate = Date

' Запустить обработку дерева
InheritAttrib oRoot, lDate, "del_me.txt"

RunAttrib = true
end function

sub InheritAttrib(aObj, aDate, aFile)
    Dim oLink, oLinks, oChild, lTypeID

    ' обработать все дочерние объекты
    set oLinks = aObj.GetLinkedObjects(1, 1)
    for each oLink in oLinks
        set oChild = oLink.ChildObject
        lTypeID = CCur(oChild.TypeID)

        if lTypeID = 793987517900000 or lTypeID = 793987517600000 then ' Эскиз, Проект
            ' Установить атрибут "Дата окончания разработки документации" (ID: 9) объекту
            oChild.SetAttrib 9, aDate

            if aFile <> "" then
                ' Добавить атрибут "файл" (ID: -6) объекту, если необходимо
                oChild.SetAttrib -6, aFile
            end if
        end if

        ' Обработать все дочерние объекты (рекурсивно)
        if lTypeID = 680515049901014 or lTypeID = 680515050001014 or
            lTypeID = 680515049801014 then ' Башня, Здание, Стена
            ' Обработать потомки только объектов типа "Башня", "Здание", "Стена"
            InheritAttrib oChild, aDate, aFile
        end if
    next
end sub

```

В данном примере все изменения сохраняются по окончании выполнения действия над объектами. В некоторых случаях требуется произвести самостоятельные изменения, независимо от хода выполнения действия. В этом случае необходимо создать новый сервис обработки объектов, произвести необходимые изменения и сохранить их. Ниже приведена измененная для такого случая процедура Run

```

sub Run
    Dim oObjSvc

    ' Создать сервис обработки объектов
    set oObjSvc = Application.CreateObjectService

    ' Выполнить изменение атрибутов
    if RunAttrib(oObjSvc) then
        ' Сохранить изменения
        oObjSvc.Update
    end if
end sub

```

## 15.7 Работа со значениями множественного атрибута

Ниже представлены две процедуры, обрабатывающие значения множественного атрибута.

Процедура Load получает все значения множественного атрибута и сцепляет их в одну строку, размещая каждое значение в отдельной строке.

Процедура Del выполняет поиск заданного значения среди всех значений множественного атрибута. Если значение найдено, то оно удаляется.

Пример написан для использования в действиях над объектами.

```
Option Explicit
```

```

sub Load
    Dim sVals, oObj, oAttribs, oItem

    ' Получить объект
    set oObj = LsOperation.ObjectService.GetObject(LsOperation.GetVarValue("a_Object"))

    ' Получить значения атрибутов
    set oAttribs = oObj.GetAttribs(LsOperation.GetVarValue("Attrib"))

    ' Сформировать список значений
    sVals = ""
    for each oItem in oAttribs
        sVals = sVals + vbCrLf + CStr(oItem)
    next

    ' Передать список значений в переменную действия
    LsOperation.SetVarValue "Vals", Mid(sVals, 3)
end sub

sub Del
    Dim sDel, oObj, oAttribs, oItem

    ' Получить объект
    set oObj = LsOperation.ObjectService.GetObject(LsOperation.GetVarValue("a_Object"))

    ' Получить значения атрибутов
    set oAttribs = oObj.GetAttribs(LsOperation.GetVarValue("Attrib"))

    ' Получить удаляемое значение
    sDel = LsOperation.GetVarValue("Del")

    ' Найти удаляемое значение
    for each oItem in oAttribs
        if oItem.Value = sDel then
            ' Значение найдено, удалить его
            oObj.DeleteAttrib oItem
        exit for
    end if
    next
end sub

```

## 15.8 Обращение к API из внешнего приложения.

В приведенном ниже примере показано, как выполнить из внешнего приложения обработку дерева проектов, приведенную в предыдущем примере. Для этого необходимо подключиться к API, создать сервис обработки объектов, выполнить обработку и сохранить изменения. В приведенном ниже коде представлена только измененная процедура Run, а остальной код остается точно таким же, как в предыдущем примере.

```

sub Run
    Dim oPDM, oObjSvc

    ' Подключиться к API
    set oPDM = CreateObject("LotsiaPDM.Application")

    ' Проверить подключение к БД
    if oPDM.Connect() then
        ' Создать сервис обработки объектов
        set oObjSvc = oPDM.CreateObjectService

        ' Выполнить изменение атрибутов
        if RunAttrib(oObjSvc) then
            ' Сохранить изменения
            oObjSvc.Update
        end if
    end if
end sub

```