POP-Java

1.0

Generated by Doxygen 1.8.3.1

Fri Mar 28 2014 14:51:33

Contents

1 Namespace Index **1**

1.1 Packages 1

2 Hierarchical Index **3**

2.1 Class Hierarchy 3

3 Class Index **7**

3.1 Class List 7

4 Namespace Documentation **11**

4.1 Package popjava.interfacebase 11

4.1.1 Detailed Description 11

5 Class Documentation **13**

5.1 popjava.baseobject.AccessPoint Class Reference 13

5.1.1 Detailed Description 14

5.1.2 Constructor & Destructor Documentation 14

5.1.2.1 AccessPoint 14

5.1.3 Member Function Documentation 14

5.1.3.1 create 14

5.1.3.2 getHost 14

5.1.3.3 getPort 15

5.1.3.4 getProtocol 15

5.1.3.5 isEmpty 15

5.1.3.6 setHost 15

5.1.3.7 setPort 15

5.1.3.8 setProtocol 16

5.2 popjava.codemanager.AppService Interface Reference 16

5.3 popjava.base.BindStatus Class Reference 16

5.3.1 Detailed Description 17

5.3.2 Member Function Documentation 17

5.3.2.1 getCode 17

5.3.2.2 getInfo 17

5.3.2.3 getPlatform 17
5.3.2.4 setCode 17
5.3.2.5 setInfo 18
5.3.2.6 setPlatform 18
5.4 popjava.broker.Broker Class Reference 18
5.4.1 Detailed Description 20
5.4.2 Member Function Documentation 20
5.4.2.1 clearResourceAfterInvoke 20
5.4.2.2 findEndcoding 20
5.4.2.3 getAccessPoint 20
5.4.2.4 getLogPrefix 20
5.4.2.5 getPOPObjectClass 20
5.4.2.6 getState 21
5.4.2.7 initialize 21
5.4.2.8 invoke 21
5.4.2.9 isDaemon 22
5.4.2.10 main 22
5.4.2.11 popCall 22
5.4.2.12 sendException 22
5.4.2.13 sendResponse 23
5.4.2.14 serveRequest 23
5.4.2.15 setState 23
5.4.2.16 treatRequests 23
5.5 popjava.buffer.BufferFactory Class Reference 24
5.5.1 Detailed Description 24
5.5.2 Member Function Documentation 24
5.5.2.1 getBufferName 24
5.6 popjava.buffer.BufferFactoryFinder Class Reference 24
5.6.1 Detailed Description 25
5.6.2 Member Function Documentation 25
5.6.2.1 findFactory 25
5.6.2.2 getInstance 25
5.6.2.3 getSupportingBuffer 25
5.6.2.4 loadBufferMap 25
5.7 popjava.buffer.BufferFactoryPlugin Class Reference 26
5.7.1 Detailed Description 26
5.7.2 Member Function Documentation 26
5.7.2.1 getBufferName 26
5.8 popjava.buffer.BufferPlugin Class Reference 26
5.8.1 Detailed Description 28

5.8.2 Member Function Documentation 28
5.8.2.1 extractHeader 28
5.8.2.2 get 28
5.8.2.3 getBoolean 29
5.8.2.4 getBooleanArray 29
5.8.2.5 getByteArray 29
5.8.2.6 getChar 29
5.8.2.7 getCharArray 29
5.8.2.8 getDouble 30
5.8.2.9 getDoubleArray 30
5.8.2.10 getFloat 30
5.8.2.11 getFloatArray 30
5.8.2.12 getInt 31
5.8.2.13 getIntArray 31
5.8.2.14 getLong 31
5.8.2.15 getLongArray 31
5.8.2.16 getShort 31
5.8.2.17 getShortArray 32
5.8.2.18 getString 32
5.8.2.19 getTranslatedInteger 32
5.8.2.20 packMessageHeader 32
5.8.2.21 put 32
5.8.2.22 put 33
5.8.2.23 put 33
5.8.2.24 putBoolean 33
5.8.2.25 putBooleanArray 33
5.8.2.26 putByteArray 33
5.8.2.27 putChar 34
5.8.2.28 putCharArray 34
5.8.2.29 putDouble 34
5.8.2.30 putDoubleArray 34
5.8.2.31 putFloat 34
5.8.2.32 putFloatArray 34
5.8.2.33 putInt 35
5.8.2.34 putIntArray 35
5.8.2.35 putLong 35
5.8.2.36 putLongArray 35
5.8.2.37 putShort 35
5.8.2.38 putShortArray 36
5.8.2.39 putString 36

5.9 popjava.buffer.BufferRaw Class Reference 36
5.9.1 Detailed Description 38
5.9.2 Constructor & Destructor Documentation 39
5.9.2.1 BufferRaw 39
5.9.3 Member Function Documentation 39
5.9.3.1 extractHeader 39
5.9.3.2 get 39
5.9.3.3 getBoolean 39
5.9.3.4 getBooleanArray 39
5.9.3.5 getByteArray 40
5.9.3.6 getChar 40
5.9.3.7 getCharArray 40
5.9.3.8 getDouble 40
5.9.3.9 getDoubleArray 41
5.9.3.10 getFloat 41
5.9.3.11 getFloatArray 41
5.9.3.12 getInt 41
5.9.3.13 getInt 41
5.9.3.14 getIntArray 42
5.9.3.15 getLong 42
5.9.3.16 getLongArray 42
5.9.3.17 getShort 42
5.9.3.18 getShortArray 43
5.9.3.19 getString 43
5.9.3.20 getTranslatedInteger 43
5.9.3.21 limit 43
5.9.3.22 packMessageHeader 44
5.9.3.23 position 44
5.9.3.24 position 44
5.9.3.25 put 44
5.9.3.26 put 44
5.9.3.27 put 45
5.9.3.28 putBoolean 45
5.9.3.29 putBooleanArray 45
5.9.3.30 putByteArray 45
5.9.3.31 putChar 45
5.9.3.32 putCharArray 46
5.9.3.33 putDouble 46
5.9.3.34 putDoubleArray 46
5.9.3.35 putFloat 46

5.9.3.36 putFloatArray . . . 46
5.9.3.37 putInt . . . 47
5.9.3.38 putInt . . . 47
5.9.3.39 putIntArray . . . 47
5.9.3.40 putLong . . . 47
5.9.3.41 putLongArray . . . 48
5.9.3.42 putShort . . . 48
5.9.3.43 putShortArray . . . 48
5.9.3.44 putString . . . 48
5.9.3.45 resize . . . 48
5.10 popjava.buffer.BufferRawFactory Class Reference . . . 49
5.10.1 Detailed Description . . . 49
5.11 popjava.buffer.BufferXDR Class Reference . . . 49
5.11.1 Detailed Description . . . 49
5.11.2 Constructor & Destructor Documentation . . . 50
5.11.2.1 BufferXDR . . . 50
5.11.2.2 BufferXDR . . . 50
5.11.3 Member Function Documentation . . . 50
5.11.3.1 putBoolean . . . 50
5.12 popjava.buffer.BufferXDRFactory Class Reference . . . 50
5.12.1 Detailed Description . . . 50
5.13 popjava.util.ClassUtil Class Reference . . . 51
5.13.1 Detailed Description . . . 51
5.13.2 Member Function Documentation . . . 51
5.13.2.1 getConstructor . . . 51
5.13.2.2 getDefaultPrimitiveValue . . . 52
5.13.2.3 getMethod . . . 52
5.13.2.4 getMethodSign . . . 52
5.13.2.5 getMethodSign . . . 52
5.14 popjava.combox.Combox Class Reference . . . 53
5.14.1 Detailed Description . . . 53
5.14.2 Constructor & Destructor Documentation . . . 54
5.14.2.1 Combox . . . 54
5.14.3 Member Function Documentation . . . 54
5.14.3.1 connect . . . 54
5.14.3.2 connect . . . 54
5.14.3.3 getBufferFactory . . . 54
5.14.3.4 receive . . . 54
5.14.3.5 send . . . 55
5.14.3.6 setBufferFactory . . . 55

5.15 popjava.combox.ComboxAcceptSocket Class Reference 55
5.15.1 Detailed Description 56
5.15.2 Constructor & Destructor Documentation 56
5.15.2.1 ComboxAcceptSocket 56
5.15.3 Member Function Documentation 56
5.15.3.1 getStatus 56
5.15.3.2 setStatus 56
5.16 popjava.combox.ComboxAllocateSocket Class Reference 56
5.16.1 Detailed Description 57
5.16.2 Member Function Documentation 57
5.16.2.1 getUrl 57
5.16.2.2 receive 57
5.16.2.3 send 57
5.17 popjava.combox.ComboxFactory Class Reference 58
5.17.1 Detailed Description 58
5.17.2 Member Function Documentation 58
5.17.2.1 createClientCombox 58
5.17.2.2 createClientCombox 59
5.17.2.3 createServerCombox 59
5.17.2.4 createServerCombox 59
5.17.2.5 getComboxName 59
5.18 popjava.combox.ComboxFactoryFinder Class Reference 60
5.18.1 Detailed Description 60
5.18.2 Member Function Documentation 60
5.18.2.1 findFactory 60
5.18.2.2 get 61
5.18.2.3 getFactoryCount 61
5.18.2.4 getInstance 61
5.18.2.5 loadComboxMap 61
5.19 popjava.combox.ComboxFactoryPlugin Class Reference 61
5.19.1 Detailed Description 62
5.19.2 Member Function Documentation 62
5.19.2.1 createClientCombox 62
5.19.2.2 createClientCombox 62
5.19.2.3 createServerCombox 63
5.19.2.4 createServerCombox 63
5.19.2.5 getComboxName 63
5.20 popjava.combox.ComboxPlugin Class Reference 63
5.20.1 Detailed Description 64
5.20.2 Member Function Documentation 64

5.20.2.1 connect 64
5.20.2.2 receive 64
5.20.2.3 send 64
5.21 popjava.combox.ComboxReceiveRequestSocket Class Reference 65
5.21.1 Detailed Description 65
5.21.2 Constructor & Destructor Documentation 66
5.21.2.1 ComboxReceiveRequestSocket 66
5.21.3 Member Function Documentation 66
5.21.3.1 getStatus 66
5.21.3.2 receiveRequest 66
5.21.3.3 setBuffer 66
5.21.3.4 setStatus 67
5.22 popjava.combox.ComboxServer Class Reference 67
5.22.1 Detailed Description 67
5.22.2 Constructor & Destructor Documentation 67
5.22.2.1 ComboxServer 67
5.22.3 Member Function Documentation 68
5.22.3.1 getRequestQueue 68
5.23 popjava.combox.ComboxServerPlugin Class Reference 68
5.23.1 Detailed Description 68
5.23.2 Constructor & Destructor Documentation 68
5.23.2.1 ComboxServerPlugin 68
5.23.3 Member Function Documentation 68
5.23.3.1 getRequestQueue 68
5.24 popjava.combox.ComboxServerSocket Class Reference 69
5.24.1 Detailed Description 69
5.24.2 Constructor & Destructor Documentation 69
5.24.2.1 ComboxServerSocket 69
5.24.3 Member Function Documentation 69
5.24.3.1 GetUrl 70
5.25 popjava.combox.ComboxSocket Class Reference 70
5.25.1 Detailed Description 70
5.25.2 Constructor & Destructor Documentation 71
5.25.2.1 ComboxSocket 71
5.25.3 Member Function Documentation 71
5.25.3.1 connect 71
5.25.3.2 receive 71
5.25.3.3 send 71
5.26 popjava.combox.ComboxSocketFactory Class Reference 72
5.26.1 Detailed Description 72

5.26.2 Member Function Documentation . . . 72
5.26.2.1 createClientCombox . . . 72
5.26.2.2 createClientCombox . . . 73
5.26.2.3 createServerCombox . . . 73
5.26.2.4 createServerCombox . . . 73
5.26.2.5 getComboxName . . . 74
5.27 popjava.util.Configuration Class Reference . . . 74
5.27.1 Detailed Description . . . 74
5.28 popjava.system.ConfigurationWorker Class Reference . . . 74
5.28.1 Detailed Description . . . 75
5.28.2 Constructor & Destructor Documentation . . . 75
5.28.2.1 ConfigurationWorker . . . 75
5.28.3 Member Function Documentation . . . 75
5.28.3.1 getValue . . . 75
5.29 popjava.annotation.POPParameter.Direction Enum Reference . . . 76
5.30 popjava.interfacebase.Interface Class Reference . . . 76
5.30.1 Detailed Description . . . 77
5.30.2 Constructor & Destructor Documentation . . . 77
5.30.2.1 Interface . . . 77
5.30.3 Member Function Documentation . . . 77
5.30.3.1 allocate . . . 77
5.30.3.2 bind . . . 78
5.30.3.3 deserialize . . . 78
5.30.3.4 getAccessPoint . . . 78
5.30.3.5 getOD . . . 78
5.30.3.6 isAlive . . . 79
5.30.3.7 popDispatch . . . 79
5.30.3.8 popResponse . . . 79
5.30.3.9 serialize . . . 79
5.30.3.10 setAccessPoint . . . 80
5.30.3.11 setOd . . . 80
5.31 popjava.dataswaper.IPOPBase Interface Reference . . . 80
5.31.1 Detailed Description . . . 80
5.31.2 Member Function Documentation . . . 80
5.31.2.1 deserialize . . . 80
5.31.2.2 serialize . . . 81
5.32 popjava.dataswaper.IPOPBaseConst Interface Reference . . . 81
5.32.1 Detailed Description . . . 81
5.33 popjava.dataswaper.IPOPBaseInput Interface Reference . . . 81
5.33.1 Detailed Description . . . 82

5.33.2 Member Function Documentation 82
5.33.2.1 deserialize 82
5.33.2.2 serialize 82
5.34 popjava.util.LogWriter Class Reference 82
5.34.1 Detailed Description 83
5.34.2 Member Function Documentation 83
5.34.2.1 deleteLogDir 83
5.34.2.2 writeDebugInfo 83
5.34.2.3 writeExceptionLog 83
5.34.2.4 writeLogfile 83
5.34.2.5 writeLogInfo 84
5.35 popjava.base.MessageHeader Class Reference 84
5.35.1 Detailed Description 85
5.35.2 Constructor & Destructor Documentation 85
5.35.2.1 MessageHeader 85
5.35.2.2 MessageHeader 85
5.35.3 Member Function Documentation 85
5.35.3.1 getClassId 85
5.35.3.2 getExceptionCode 86
5.35.3.3 getMethodId 86
5.35.3.4 getRequestType 86
5.35.3.5 getSenmatics 86
5.35.3.6 setClassId 86
5.35.3.7 setExceptionCode 87
5.35.3.8 setMethodId 87
5.35.3.9 setRequestType 87
5.35.3.10 setSenmatics 87
5.36 popjava.base.MethodInfo Class Reference 87
5.36.1 Detailed Description 88
5.36.2 Constructor & Destructor Documentation 88
5.36.2.1 MethodInfo 88
5.36.3 Member Function Documentation 88
5.36.3.1 equals 88
5.36.3.2 getClassId 88
5.36.3.3 getMethodId 89
5.37 popjava.baseobject.ObjectDescription Class Reference 89
5.37.1 Detailed Description 91
5.37.2 Member Function Documentation 92
5.37.2.1 getBatch 92
5.37.2.2 getCodeFile 92

5.37.2.3 getEncoding 92
5.37.2.4 getHostarch 92
5.37.2.5 getHostcore 92
5.37.2.6 getHostName 92
5.37.2.7 getHostuser 93
5.37.2.8 getJobUrl 93
5.37.2.9 getJVMParameters 93
5.37.2.10 getPlatform 93
5.37.2.11 getPowerMin 93
5.37.2.12 getProtocol 93
5.37.2.13 getSearchMaxDepth 94
5.37.2.14 getSearchMaxSize 94
5.37.2.15 getSearchWaitTime 94
5.37.2.16 getValue 94
5.37.2.17 getWallTime 94
5.37.2.18 isEmpty 94
5.37.2.19 isSearchSet 95
5.37.2.20 manual 95
5.37.2.21 merge 95
5.37.2.22 removeValue 95
5.37.2.23 setBandwidth 95
5.37.2.24 setBatch 96
5.37.2.25 setCodeFile 96
5.37.2.26 setDirectory 96
5.37.2.27 setEncoding 96
5.37.2.28 setHostarch 96
5.37.2.29 setHostcore 96
5.37.2.30 setHostname 97
5.37.2.31 setHostuser 97
5.37.2.32 setJobUrl 97
5.37.2.33 setJVMParamters 97
5.37.2.34 setMemory 97
5.37.2.35 setPlatform 98
5.37.2.36 setPower 98
5.37.2.37 setProtocol 98
5.37.2.38 setSearch 98
5.37.2.39 setValue 98
5.37.2.40 setWallTime 99
5.38 popjava.dataswaper.ObjectDescriptionInput Class Reference 99
5.38.1 Detailed Description 101

5.38.2 Constructor & Destructor Documentation 101
5.38.2.1 ObjectDescriptionInput 101
5.38.3 Member Function Documentation 101
5.38.3.1 getCodeFile 101
5.38.3.2 getEncoding 101
5.38.3.3 getHostName 101
5.38.3.4 getJobUrl 101
5.38.3.5 getPlatform 102
5.38.3.6 getProtocol 102
5.38.3.7 getValue 102
5.38.3.8 getWallTime 102
5.38.3.9 isEmpty 102
5.38.3.10 merge 103
5.38.3.11 removeValue 103
5.38.3.12 setBandwidth 103
5.38.3.13 setCodeFile 103
5.38.3.14 setEncoding 103
5.38.3.15 setHostname 103
5.38.3.16 setJobUrl 104
5.38.3.17 setMemory 104
5.38.3.18 setPlatform 104
5.38.3.19 setPower 104
5.38.3.20 setProtocol 104
5.38.3.21 setSearch 104
5.38.3.22 setValue 105
5.38.3.23 setWallTime 105
5.39 popjava.baseobject.ODElement Class Reference 105
5.39.1 Detailed Description 106
5.39.2 Constructor & Destructor Documentation 106
5.39.2.1 ODElement 106
5.39.3 Member Function Documentation 106
5.39.3.1 deserialize 106
5.39.3.2 getMinValue 106
5.39.3.3 getRequiredValue 106
5.39.3.4 isEmpty 107
5.39.3.5 serialize 107
5.39.3.6 set 107
5.39.3.7 set 107
5.39.3.8 setMinValue 107
5.39.3.9 setRequiredValue 107

5.40 popjava.PJMethodFilter Class Reference 108
5.40.1 Detailed Description 108
5.40.2 Member Function Documentation 108
5.40.2.1 isHandled 108
5.41 popjava.PJMethodHandler Class Reference 108
5.41.1 Detailed Description 109
5.41.2 Constructor & Destructor Documentation 109
5.41.2.1 PJMethodHandler 109
5.41.3 Member Function Documentation 109
5.41.3.1 bindObject 109
5.41.3.2 invoke 109
5.41.3.3 popConstructor 110
5.42 popjava.PJProxyFactory Class Reference 110
5.42.1 Detailed Description 111
5.42.2 Constructor & Destructor Documentation 111
5.42.2.1 PJProxyFactory 111
5.42.3 Member Function Documentation 111
5.42.3.1 bindPOPObject 111
5.42.3.2 newActiveFromBuffer 111
5.42.3.3 newPOPObject 112
5.42.3.4 newPOPObject 112
5.43 popjava.baseobject.POPAccessPoint Class Reference 113
5.43.1 Detailed Description 113
5.43.2 Constructor & Destructor Documentation 113
5.43.2.1 POPAccessPoint 113
5.43.2.2 POPAccessPoint 114
5.43.3 Member Function Documentation 114
5.43.3.1 addAccessPoint 114
5.43.3.2 get 114
5.43.3.3 isEmpty 114
5.43.3.4 setAccessString 114
5.43.3.5 size 115
5.44 popjava.serviceadapter.POPAppService Class Reference 115
5.44.1 Detailed Description 115
5.44.2 Constructor & Destructor Documentation 115
5.44.2.1 POPAppService 115
5.44.2.2 POPAppService 116
5.44.3 Member Function Documentation 116
5.44.3.1 queryService 116
5.44.3.2 queryService 116

5.44.3.3 registerService 116
5.44.3.4 unregisterService 116
5.45 popjava.annotation.POPAsyncConc Interface Reference 117
5.45.1 Detailed Description 117
5.46 popjava.annotation.POPAsyncMutex Interface Reference 117
5.47 popjava.annotation.POPAsyncSeq Interface Reference 117
5.48 popjava.buffer.POPBuffer Class Reference 117
5.48.1 Detailed Description 120
5.48.2 Constructor & Destructor Documentation 120
5.48.2.1 POPBuffer 120
5.48.3 Member Function Documentation 120
5.48.3.1 checkAndThrow 120
5.48.3.2 deserializeReferenceObject 120
5.48.3.3 extractHeader 121
5.48.3.4 get 121
5.48.3.5 getArray 121
5.48.3.6 getBoolean 121
5.48.3.7 getBooleanArray 122
5.48.3.8 getByteArray 122
5.48.3.9 getChar 122
5.48.3.10 getCharArray 122
5.48.3.11 getDouble 123
5.48.3.12 getDoubleArray 123
5.48.3.13 getFloat 123
5.48.3.14 getFloatArray 123
5.48.3.15 getHeader 124
5.48.3.16 getInt 124
5.48.3.17 getIntArray 124
5.48.3.18 getLong 124
5.48.3.19 getLongArray 124
5.48.3.20 getShort 125
5.48.3.21 getShortArray 125
5.48.3.22 getString 125
5.48.3.23 getTranslatedInteger 125
5.48.3.24 getValue 126
5.48.3.25 packMessageHeader 126
5.48.3.26 put 126
5.48.3.27 put 126
5.48.3.28 put 127
5.48.3.29 putArray 127

5.48.3.30 putBoolean . . . 127
5.48.3.31 putBooleanArray . . . 127
5.48.3.32 putByteArray . . . 128
5.48.3.33 putChar . . . 128
5.48.3.34 putCharArray . . . 128
5.48.3.35 putDouble . . . 128
5.48.3.36 putDoubleArray . . . 128
5.48.3.37 putFloat . . . 129
5.48.3.38 putFloatArray . . . 129
5.48.3.39 putInt . . . 129
5.48.3.40 putIntArray . . . 129
5.48.3.41 putLong . . . 129
5.48.3.42 putLongArray . . . 130
5.48.3.43 putShort . . . 130
5.48.3.44 putShortArray . . . 130
5.48.3.45 putString . . . 130
5.48.3.46 putValue . . . 130
5.48.3.47 serializeReferenceObject . . . 131
5.48.3.48 setHeader . . . 131
5.48.3.49 size . . . 131
5.48.3.50 toCharString . . . 131
5.48.3.51 toIntString . . . 131
5.49 popjava.annotation.POPClass Interface Reference . . . 132
5.50 popjava.annotation.processors.POPClassProcessor Class Reference . . . 132
5.50.1 Detailed Description . . . 132
5.51 popjava.serviceadapter.POPCodeManager Class Reference . . . 132
5.51.1 Detailed Description . . . 133
5.51.2 Constructor & Destructor Documentation . . . 133
5.51.2.1 POPCodeManager . . . 133
5.51.2.2 POPCodeManager . . . 133
5.51.3 Member Function Documentation . . . 133
5.51.3.1 getPlatform . . . 133
5.51.3.2 queryCode . . . 133
5.51.3.3 registerCode . . . 134
5.52 popjava.annotation.POPConfig Interface Reference . . . 134
5.53 popjava.base.POPErrorCode Class Reference . . . 134
5.53.1 Detailed Description . . . 135
5.54 popjava.base.POPException Class Reference . . . 135
5.54.1 Detailed Description . . . 136
5.54.2 Constructor & Destructor Documentation . . . 136

5.54.2.1 POPException 136
5.54.3 Member Function Documentation 136
5.54.3.1 createReflectException 136
5.54.3.2 createReflectMethodNotFoundException 137
5.54.3.3 deserialize 137
5.54.3.4 serialize 137
5.54.3.5 throwAccessPointNotAvailableException 137
5.54.3.6 throwBufferFormatException 138
5.54.3.7 throwBufferNotAvailableException 138
5.54.3.8 throwComboxNotAvailableException 138
5.54.3.9 throwNullObjectNotAllowException 138
5.54.3.10 throwObjectBindException 139
5.54.3.11 throwObjectNoResource 139
5.54.3.12 throwReflectException 139
5.54.3.13 throwReflectMethodNotFoundException 139
5.54.3.14 throwReflectSerializeException 140
5.55 popjava.PopJava Class Reference 140
5.55.1 Detailed Description 140
5.55.2 Member Function Documentation 140
5.55.2.1 newActive 140
5.55.2.2 newActive 141
5.55.2.3 newActive 141
5.55.2.4 newActiveFromBuffer 142
5.56 popjava.codemanager.POPJavaAppService Class Reference 142
5.56.1 Member Function Documentation 142
5.56.1.1 getPlatform 142
5.56.1.2 queryCode 143
5.56.1.3 registerCode 143
5.57 popjava.scripts.Popjavac Class Reference 143
5.58 popjava.system.POPJavaConfiguration Class Reference 143
5.58.1 Member Function Documentation 144
5.58.1.1 getPopAppCoreService 144
5.58.1.2 getPopJavaLocation 144
5.58.1.3 getPopPluginLocation 144
5.59 popjava.serviceadapter.POPJobManager Class Reference 144
5.59.1 Detailed Description 145
5.59.2 Constructor & Destructor Documentation 145
5.59.2.1 POPJobManager 145
5.59.2.2 POPJobManager 145
5.59.2.3 POPJobManager 146

5.59.3 Member Function Documentation 146
5.59.3.1 allocResource 146
5.59.3.2 cancelReservation 146
5.59.3.3 createObject 146
5.59.3.4 execObj 147
5.59.3.5 query 147
5.59.3.6 registerNode 147
5.60 popjava.serviceadapter.POPJobService Class Reference 148
5.60.1 Detailed Description 148
5.60.2 Constructor & Destructor Documentation 148
5.60.2.1 POPJobService 148
5.60.2.2 POPJobService 148
5.60.3 Member Function Documentation 148
5.60.3.1 createObject 148
5.61 popjava.scripts.Popjrun Class Reference 149
5.62 popjava.base.POPObject Class Reference 149
5.62.1 Detailed Description 151
5.62.2 Member Function Documentation 151
5.62.2.1 addSemantic 151
5.62.2.2 addSemantic 151
5.62.2.3 canKill 151
5.62.2.4 defineConstructor 152
5.62.2.5 defineMethod 152
5.62.2.6 deserialize 152
5.62.2.7 getAccessPoint 152
5.62.2.8 getClassId 152
5.62.2.9 getClassName 153
5.62.2.10 getConstructorByInfo 153
5.62.2.11 getMethodByInfo 153
5.62.2.12 getMethodInfo 154
5.62.2.13 getMethodInfo 154
5.62.2.14 getOd 154
5.62.2.15 getPOPCReference 154
5.62.2.16 getSemantic 154
5.62.2.17 getSemantic 155
5.62.2.18 hasDestructor 155
5.62.2.19 hasDestructor 155
5.62.2.20 initializeConstructorInfo 155
5.62.2.21 initializeMethodInfo 156
5.62.2.22 initializePOPObject 156

5.62.2.23 isDaemon . . . 156
5.62.2.24 serialize . . . 156
5.62.2.25 setClassId . . . 156
5.62.2.26 setClassName . . . 157
5.62.2.27 setOd . . . 157
5.63 popjava.annotation.POPObjectDescription Interface Reference . . . 157
5.63.1 Member Function Documentation . . . 157
5.63.1.1 jvmParameters . . . 157
5.64 popjava.serviceadapter.POPObjectMonitor Class Reference . . . 157
5.64.1 Detailed Description . . . 158
5.64.2 Constructor & Destructor Documentation . . . 158
5.64.2.1 POPObjectMonitor . . . 158
5.64.2.2 POPObjectMonitor . . . 158
5.64.3 Member Function Documentation . . . 158
5.64.3.1 checkObjects . . . 158
5.64.3.2 manageObject . . . 158
5.64.3.3 unManageObject . . . 159
5.65 popjava.annotation.POPParameter Interface Reference . . . 159
5.66 popjava.baseobject.POPReference Class Reference . . . 159
5.66.1 Detailed Description . . . 159
5.66.2 Member Function Documentation . . . 159
5.66.2.1 setAccessPoint . . . 159
5.67 popjava.serviceadapter.POPRemoteLog Class Reference . . . 160
5.67.1 Detailed Description . . . 160
5.67.2 Constructor & Destructor Documentation . . . 160
5.67.2.1 POPRemoteLog . . . 160
5.67.2.2 POPRemoteLog . . . 160
5.67.3 Member Function Documentation . . . 160
5.67.3.1 log . . . 160
5.68 popjava.system.POPRemoteLogThread Class Reference . . . 161
5.68.1 Detailed Description . . . 161
5.68.2 Constructor & Destructor Documentation . . . 161
5.68.2.1 POPRemoteLogThread . . . 161
5.68.3 Member Function Documentation . . . 161
5.68.3.1 getFilename . . . 161
5.68.3.2 run . . . 161
5.68.3.3 setRunning . . . 161
5.69 popjava.serviceadapter.POPServiceBase Class Reference . . . 162
5.69.1 Detailed Description . . . 162
5.69.2 Constructor & Destructor Documentation . . . 162

5.69.2.1 POPServiceBase 162
5.69.2.2 POPServiceBase 162
5.69.3 Member Function Documentation 162
5.69.3.1 start 163
5.69.3.2 stop 163
5.70 popjava.dataswaper.POPString Class Reference 163
5.70.1 Detailed Description 163
5.70.2 Constructor & Destructor Documentation 163
5.70.2.1 POPString 163
5.70.3 Member Function Documentation 164
5.70.3.1 getValue 164
5.70.3.2 setValue 164
5.71 popjava.annotation.POPSyncConc Interface Reference 164
5.72 popjava.annotation.POPSyncMutex Interface Reference 164
5.73 popjava.annotation.POPSyncSeq Interface Reference 164
5.74 popjava.system.POPSystem Class Reference 164
5.74.1 Detailed Description 165
5.74.2 Member Function Documentation 165
5.74.2.1 createAppCoreService 165
5.74.2.2 getDefaultAccessPoint 166
5.74.2.3 getDefaultOD 166
5.74.2.4 getEnviroment 166
5.74.2.5 getHostIP 166
5.74.2.6 getIPAsInt 166
5.74.2.7 getPlatform 167
5.74.2.8 initCodeService 167
5.74.2.9 initialize 167
5.74.2.10 initialize 168
5.75 popjava.base.POPSystemErrorCode Class Reference 168
5.75.1 Detailed Description 168
5.76 popjava.broker.POPThread Class Reference 168
5.76.1 Detailed Description 169
5.76.2 Member Function Documentation 169
5.76.2.1 getRequest 169
5.76.2.2 setRequest 169
5.77 popjava.broker.Request Class Reference 169
5.77.1 Detailed Description 171
5.77.2 Constructor & Destructor Documentation 171
5.77.2.1 Request 171
5.77.3 Member Function Documentation 171

5.77.3.1 getBroker . . . 171
5.77.3.2 getBuffer . . . 171
5.77.3.3 getClassId . . . 171
5.77.3.4 getCombox . . . 172
5.77.3.5 getMethodId . . . 172
5.77.3.6 getReceiveCombox . . . 172
5.77.3.7 getSenmatics . . . 172
5.77.3.8 getStatus . . . 172
5.77.3.9 init . . . 172
5.77.3.10 isConcurrent . . . 173
5.77.3.11 isMutex . . . 173
5.77.3.12 isSequential . . . 173
5.77.3.13 isSynchronous . . . 173
5.77.3.14 setBroker . . . 174
5.77.3.15 setBuffer . . . 174
5.77.3.16 setBuffer . . . 174
5.77.3.17 setClassId . . . 174
5.77.3.18 setCombox . . . 174
5.77.3.19 setMethodId . . . 174
5.77.3.20 setReceiveCombox . . . 175
5.77.3.21 setSenmatics . . . 175
5.77.3.22 setStatus . . . 175
5.78 popjava.broker.RequestQueue Class Reference . . . 175
5.78.1 Detailed Description . . . 176
5.78.2 Member Function Documentation . . . 176
5.78.2.1 add . . . 176
5.78.2.2 canPeek . . . 176
5.78.2.3 clear . . . 176
5.78.2.4 getMaxQueue . . . 177
5.78.2.5 peek . . . 177
5.78.2.6 remove . . . 177
5.78.2.7 setMaxQueue . . . 177
5.78.2.8 size . . . 177
5.79 popjava.scripts.ScriptUtils Class Reference . . . 178
5.80 popjava.base.Semantic Class Reference . . . 178
5.80.1 Detailed Description . . . 178
5.81 popjava.util.SystemUtil Class Reference . . . 178
5.81.1 Detailed Description . . . 179
5.81.2 Member Function Documentation . . . 179
5.81.2.1 runCmd . . . 179

5.82 popjava.annotation.POPConfig.Type Enum Reference 179
5.83 popjava.util.Util Class Reference 179
5.83.1 Detailed Description 180
5.83.2 Member Function Documentation 180
5.83.2.1 byteArrayToInt 180
5.83.2.2 generateRandomString 180
5.83.2.3 isLocal 180
5.83.2.4 isNoCaseStringEqual 181
5.83.2.5 isParameterNotOfDirection 181
5.83.2.6 isStringEqual 181
5.83.2.7 matchPlatform 181
5.83.2.8 removeStringFromArrayList 182
5.83.2.9 sameContact 182
5.83.2.10 splitTheCommand 182
5.84 popjava.system.XMLWorker Class Reference 182
5.84.1 Detailed Description 183
5.84.2 Member Function Documentation 183
5.84.2.1 isValid 183

Index **183**

Chapter 1

Namespace Index

1.1 Packages

Here are the packages with brief descriptions (if available):

popjava.interfacebase
Interface.java 11

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

- popjava.baseobject.AccessPoint 13
- popjava.codemanager.AppService 16
  - popjava.codemanager.POPJavaAppService 142
  - popjava.serviceadapter.POPAppService 115
- popjava.base.BindStatus 16
- popjava.broker.Broker 18
- popjava.buffer.BufferFactory 24
  - popjava.buffer.BufferFactoryPlugin 26
  - popjava.buffer.BufferRawFactory 49
  - popjava.buffer.BufferXDRFactory 50
- popjava.buffer.BufferFactoryFinder 24
- popjava.util.ClassUtil 51
- popjava.combox.Combox 53
  - popjava.combox.ComboxPlugin 63
  - popjava.combox.ComboxSocket 70
- popjava.combox.ComboxAllocateSocket 56
- popjava.combox.ComboxFactory 58
  - popjava.combox.ComboxFactoryPlugin 61
  - popjava.combox.ComboxSocketFactory 72
- popjava.combox.ComboxFactoryFinder 60
- popjava.combox.ComboxServer 67
  - popjava.combox.ComboxServerPlugin 68
  - popjava.combox.ComboxServerSocket 69
- popjava.util.Configuration 74
- popjava.annotation.POPParameter.Direction 76
- popjava.interfacebase.Interface 76
  - popjava.PJMethodHandler 108
- popjava.dataswaper.IPOPBase 80
  - popjava.base.POPException 135
  - popjava.base.POPObject 149
    - popjava.codemanager.POPJavaAppService 142
    - popjava.serviceadapter.POPServiceBase 162
      - popjava.serviceadapter.POPJobService 148
        - popjava.serviceadapter.POPJobManager 144
      - popjava.serviceadapter.POPRemoteLog 160
        - popjava.serviceadapter.POPCodeManager 132

popjava.serviceadapter.POPObjectMonitor 157
popjava.serviceadapter.POPAppService 115
popjava.baseobject.ObjectDescription 89
popjava.baseobject.POPAccessPoint 113
popjava.dataswaper.POPString 163
popjava.dataswaper.IPOPBaseConst 81
popjava.dataswaper.IPOPBaseInput 81
popjava.dataswaper.ObjectDescriptionInput 99
popjava.util.LogWriter 82
popjava.base.MessageHeader 84
MethodHandler
popjava.PJMethodHandler 108
popjava.base.MethodInfo 87
Object
popjava.buffer.POPBuffer 117
popjava.buffer.BufferPlugin 26
popjava.buffer.BufferRaw 36
popjava.buffer.BufferXDR 49
popjava.baseobject.ODElement 105
popjava.annotation.POPAsyncConc 117
popjava.annotation.POPAsyncMutex 117
popjava.annotation.POPAsyncSeq 117
popjava.annotation.POPClass 132
popjava.annotation.POPConfig 134
popjava.base.POPErrorCode 134
popjava.PopJava 140
popjava.scripts.Popjavac 143
popjava.system.POPJavaConfiguration 143
popjava.scripts.Popjrun 149
popjava.annotation.POPObjectDescription 157
popjava.annotation.POPParameter 159
popjava.baseobject.POPReference 159
popjava.annotation.POPSyncConc 164
popjava.annotation.POPSyncMutex 164
popjava.annotation.POPSyncSeq 164
popjava.system.POPSystem 164
popjava.base.POPSystemErrorCode 168
popjava.broker.Request 169
popjava.broker.RequestQueue 175
Runnable
popjava.broker.POPThread 168
popjava.combox.ComboxAcceptSocket 55
popjava.combox.ComboxReceiveRequestSocket 65
RuntimeException
popjava.base.POPException 135
popjava.scripts.ScriptUtils 178
popjava.base.Semantic 178
popjava.util.SystemUtil 178
Thread
popjava.system.POPRemoteLogThread 161
popjava.annotation.POPConfig.Type 179
popjava.util.Util 179
popjava.system.XMLWorker 182
popjava.system.ConfigurationWorker 74
AbstractProcessor
popjava.annotation.processors.POPClassProcessor 132
MethodFilter

popjava.PJMethodFilter 108
ProxyFactory
popjava.PJProxyFactory 110

Chapter 3

Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

popjava.baseobject.AccessPoint
This class represent an access to a broker-side parallel object 13
popjava.codemanager.AppService 16
popjava.base.BindStatus
This class represent the different bind status that a connection between a interface and a broker can have 16
popjava.broker.Broker
This class is the base class of all broker-side parallel object 18
popjava.buffer.BufferFactory
This abstract class defined all the methods needed by a BufferFactory 24
popjava.buffer.BufferFactoryFinder
This class is responsible to discover the buffer 24
popjava.buffer.BufferFactoryPlugin
This class defined the interface for new buffer factory plug-in 26
popjava.buffer.BufferPlugin
This class defined the interface for each new buffer plug-in 26
popjava.buffer.BufferRaw
This class is a RAW implementation of the buffer abstract class 36
popjava.buffer.BufferRawFactory
Implementation of the abstract BufferFactory for the RAW encoding 49
popjava.buffer.BufferXDR
This class is a XDR extension of the BufferRAW class 49
popjava.buffer.BufferXDRFactory
Implementation of the abstract BufferFactory for the RAW encoding 50
popjava.util.ClassUtil
This class gives some static methods to look inside a class 51
popjava.combox.Combox
This class is the base implementation for all Combox in the POP-Java library All other combox must inherit from this class 53
popjava.combox.ComboxAcceptSocket
This class is responsible to accept the new connection for the associated server combox socket 55
popjava.combox.ComboxAllocateSocket
This class is responsible to send an receive message on the server combox socket 56
popjava.combox.ComboxFactory
This abstract class regroup the method needed by a ComboxFactory 58
popjava.combox.ComboxFactoryFinder
This class is responsible to find the different combox available in POP-Java 60

popjava.combox.ComboxFactoryPlugin
This class defined the interface for new combox factory plug-in 61
popjava.combox.ComboxPlugin
This class defined the interface for each new combox plug-in 63
popjava.combox.ComboxReceiveRequestSocket
This class is responsible to receive the new request for the associated combox 65
popjava.combox.ComboxServer
This class represent the server side of a socket connection 67
popjava.combox.ComboxServerPlugin
This class defined the interface for all new combox server plug-in 68
popjava.combox.ComboxServerSocket
This class is an implementation of the combox with the protocol socket for the server side . . . 69
popjava.combox.ComboxSocket
This combox implement the protocol Socket 70
popjava.combox.ComboxSocketFactory
This class is the factory for all combox socket 72
popjava.util.Configuration
This class regroup some configuration values 74
popjava.system.ConfigurationWorker
POP-Java configuration class 74
popjava.annotation.POPParameter.Direction 76
popjava.interfacebase.Interface
Interface side of a POP-Java parallel object 76
popjava.dataswaper.IPOPBase
This interface declare the needed method for the serialization and the deserialization of an object 80
popjava.dataswaper.IPOPBaseConst
This type is used for communicate with the pop-c++ only 81
popjava.dataswaper.IPOPBaseInput
This type is used for communicate with the pop-c++ only 81
popjava.util.LogWriter
This class is used to write log file 82
popjava.base.MessageHeader
Message header is include in all communication between Interface and Broker side 84
popjava.base.MethodInfo
This class represents all the informations about a method in a parallel object 87
popjava.baseobject.ObjectDescription
This class represents the object description for a parallel object 89
popjava.dataswaper.ObjectDescriptionInput
Compatible implementation of the ObjectDescription POP-Java object for POP-C++ 99
popjava.baseobject.ODElement
This class represents an ODElement for the object description 105
popjava.PJMethodFilter
This class is a method filter for the PJMethodHandler 108
popjava.PJMethodHandler
This class is responsible to invoke methods on the parallel object 108
popjava.PJProxyFactory
POP-Java Proxy Factory : this class provide methods to create a proxy factory for a specified class 110
popjava.baseobject.POPAccessPoint
This class represents multiple access to the broker-side parallel object 113
popjava.serviceadapter.POPAppService
Partial POP-Java class implementation to be used with the POP-C++ runtime This class declares the necessary methods to use the AppService parallel object of POP-C++ 115
popjava.annotation.POPAsyncConc 117
popjava.annotation.POPAsyncMutex 117
popjava.annotation.POPAsyncSeq 117
popjava.buffer.POPBuffer
This abstract class defined all the required methods to implement a buffer 117

popjava.annotation.POPClass 132
popjava.annotation.processors.POPClassProcessor
Http://www.javaspecialists.eu/archive/Issue167.html 132
popjava.serviceadapter.POPCodeManager
Partial POP-Java class implementation to be used with the POP-C++ runtime This class declares the necessary methods to use the CodeMgr parallel object of POP-C++ 132
popjava.annotation.POPConfig 134
popjava.base.POPErrorCode
This class regroup all POP error code 134
popjava.base.POPException
This class is the base implementation for all POP exception 135
popjava.PopJava
This class is used to create parallel object 140
popjava.codemanager.POPJavaAppService 142
popjava.scripts.Popjavac 143
popjava.system.POPJavaConfiguration 143
popjava.serviceadapter.POPJobManager
Partial POP-Java class implementation to be used with the POP-C++ runtime This class declares the necessary methods to use the JobMgr parallel object of POP-C++ 144
popjava.serviceadapter.POPJobService
Partial POP-Java class implementation to be used with the POP-C++ runtime This class declares the necessary methods to use the JobMgr parallel object of POP-C++ 148
popjava.scripts.Popjrun 149
popjava.base.POPObject
This class is the base class of all POP-Java parallel classes 149
popjava.annotation.POPObjectDescription 157
popjava.serviceadapter.POPObjectMonitor
Partial POP-Java class implementation to be used with the POP-C++ runtime This class declares the necessary methods to use the ObjectMonitor parallel object of POP-C++ 157
popjava.annotation.POPParameter 159
popjava.baseobject.POPReference
This class defined a POPReference 159
popjava.serviceadapter.POPRemoteLog
Partial POP-Java class implementation to be used with the POP-C++ runtime This class declares the necessary methods to use the RemoteLog parallel object of POP-C++ 160
popjava.system.POPRemoteLogThread 161
popjava.serviceadapter.POPServiceBase
Partial POP-Java class implementation to be used with the POP-C++ runtime This class declares the necessary methods to use the paroc_service_base parallel object of POP-C++ 162
popjava.dataswaper.POPString
Compatible with the POP-C++ paroc_string implementation 163
popjava.annotation.POPSyncConc 164
popjava.annotation.POPSyncMutex 164
popjava.annotation.POPSyncSeq 164
popjava.system.POPSystem
This class is responsible for the initialization of a POP-Java application 164
popjava.base.POPSystemErrorCode
This class regroup all exception code 168
popjava.broker.POPThread
Base class of POPThread 168
popjava.broker.Request
This class symbolize a request between the interface-side and the broker-side 169
popjava.broker.RequestQueue
This class represents the request queue used in the broker-side Every requests are put into this request queue and are served in FIFO order 175
popjava.scripts.ScriptUtils 178
popjava.base.Semantic
This class class is used to store the different semantics used in the POP model 178

popjava.util.SystemUtil
This glass gives some static method to deal with the system 178
popjava.annotation.POPConfig.Type 179
popjava.util.Util
This class gives some static utility methods 179
popjava.system.XMLWorker
Base class to handle XML validation 182

Chapter 4

Namespace Documentation

4.1 Package popjava.interfacebase

Interface.java.

Classes

- class Interface

    Interface side of a POP-Java parallel object.

4.1.1 Detailed Description

Interface.java.

Chapter 5

Class Documentation

5.1 popjava.baseobject.AccessPoint Class Reference

This class represent an access to a broker-side parallel object.

Collaboration diagram for popjava.baseobject.AccessPoint:

Public Member Functions

- AccessPoint ()

  Create a new AccessPoint.
- AccessPoint (String protocol, String host, int port)

  Create new access point with given values.
- String getProtocol ()

  Get the protocol of this access point.
- String getHost ()

  Get the host of this access point.
- int getPort ()

  Get the port of this access point.
- void setPort (int port)

  Set the port for this access point.
- void setProtocol (String protocol)

  Set the protocol for this access point.
- void setHost (String host)

  Set the host form this access point.
- boolean isEmpty ()

  Check is the access point is empty.
- String toString ()

  Format the access point as a string value.

Static Public Member Functions

- static AccessPoint create (String accessString)

  Create an access point from a formatted string.

Static Public Attributes

- static final String **SocketProtocol** = "socket"
- static final String **WSProtocol** = "webservice"
- static final String **HTTPProtocol** = "http"
- static final int **DefaultPort** = 12008
- static final String **DefaultHost** = "localhost"

Protected Attributes

- String **protocol**
- int **port**
- String **host**

5.1.1 Detailed Description

This class represent an access to a broker-side parallel object.

5.1.2 Constructor & Destructor Documentation

5.1.2.1 popjava.baseobject.AccessPoint.AccessPoint (String *protocol,* String *host,* int *port*)

Create new access point with given values.

Parameters

protocol	Protocol of the access point
host	Host of the access point
port	Port on which the broker is listening to

5.1.3 Member Function Documentation

5.1.3.1 static AccessPoint popjava.baseobject.AccessPoint.create (String *accessString*) [static]

Create an access point from a formatted string.

Parameters

accessString	Formatted access string

Returns

the new access point created from the string

Here is the call graph for this function:

Here is the caller graph for this function:

5.1.3.2 String popjava.baseobject.AccessPoint.getHost ( )

Get the host of this access point.

Returns

host as a string value

Here is the caller graph for this function:

5.1.3.3 int popjava.baseobject.AccessPoint.getPort ( )

Get the port of this access point.

Returns

port as an int value

Here is the caller graph for this function:

5.1.3.4 String popjava.baseobject.AccessPoint.getProtocol ( )

Get the protocol of this access point.

Returns

protocol as a string value

Here is the caller graph for this function:

5.1.3.5 boolean popjava.baseobject.AccessPoint.isEmpty ( )

Check is the access point is empty.

Returns

true if the access point is not set

5.1.3.6 void popjava.baseobject.AccessPoint.setHost (String *host*)

Set the host form this access point.

Parameters

host	The host to set

Here is the caller graph for this function:

5.1.3.7 void popjava.baseobject.AccessPoint.setPort (int *port*)

Set the port for this access point.

Parameters

port	The port to set

Here is the caller graph for this function:

5.1.3.8 void popjava.baseobject.AccessPoint.setProtocol (String *protocol*)

Set the protocol for this access point.

Parameters

protocol	The protocol to set

Here is the caller graph for this function:

5.2 popjava.codemanager.AppService Interface Reference

Inheritance diagram for popjava.codemanager.AppService:

Collaboration diagram for popjava.codemanager.AppService:

Public Member Functions

- void **registerCode** (String objname, String platform, String codefile)
- int **queryCode** (String objname, String platform, POPString codefile)
- int **getPlatform** (String objname, POPString platform)
- POPAccessPoint **getAccessPoint** ()
- String **getPOPCAppID** ()
- void **exit** ()

5.3 popjava.base.BindStatus Class Reference

This class represent the different bind status that a connection between a interface and a broker can have.

Collaboration diagram for popjava.base.BindStatus:

Public Member Functions

- BindStatus ()

  Creates a new instance of BindStatus.
- int getCode ()

  Get the code associated with this bind status.
- void setCode (int code)

  Associate a platform with this bind status.
- String getPlatform ()

  Get the platform associated with this bind status.
- void setPlatform (String platform)

  Associate a platform with this bind status.
- String getInfo ()

  Get informations of this bind status.
- void setInfo (String info)

  Set informations to this bind status.

Static Public Attributes

- static final int **BindOK** = 0
- static final int **BindForwardSession** = 1
- static final int **BindForwardPermanent** = 2
- static final int **BindAllocRetry** = 3

Protected Attributes

- int **code**
- String **platform**
- String **info**

5.3.1 Detailed Description

This class represent the different bind status that a connection between a interface and a broker can have.

5.3.2 Member Function Documentation

5.3.2.1 int popjava.base.BindStatus.getCode ( )

Get the code associated with this bind status.

Returns

the associated code

Here is the caller graph for this function:

5.3.2.2 String popjava.base.BindStatus.getInfo ( )

Get informations of this bind status.

Returns

return informations as a string value

5.3.2.3 String popjava.base.BindStatus.getPlatform ( )

Get the platform associated with this bind status.

Returns

the associated platform

Here is the caller graph for this function:

5.3.2.4 void popjava.base.BindStatus.setCode (int *code*)

Associate a platform with this bind status.

Parameters

code	the associated code

5.3.2.5 void popjava.base.BindStatus.setInfo (String *info*)

Set informations to this bind status.

Parameters

info	The informations to set as a string value

5.3.2.6 void popjava.base.BindStatus.setPlatform (String *platform*)

Associate a platform with this bind status.

Parameters

platform	string value of the platform

5.4 popjava.broker.Broker Class Reference

This class is the base class of all broker-side parallel object.

Collaboration diagram for popjava.broker.Broker:

Public Member Functions

- boolean invoke (Request request) throws InterruptedException

    This method is responsible to dispatch the request between invokeConstructor and invokeMethod.
- void clearResourceAfterInvoke (Request request)

    Remove the request from the request queue after invocation.
- void serveRequest (final Request request) throws InterruptedException

    This method is responsible to handle the broker-side semantics for a request.
- boolean popCall (Request request)

    This method is responsible to handle the POP system call.
- synchronized void kill ()

    Kill the broker and its associated object.
- void treatRequests () throws InterruptedException

    Main loop of this broker.
- synchronized void onNewConnection ()

    Increment the connection counter.
- synchronized void onCloseConnection ()

    Decrement de connection counter and exit the broker if there is no more connection.
- boolean isDaemon ()

    Get information about the deamon mode of this broker.
- synchronized int getState ()

    Get information about the state of this borker.
- synchronized void setState (int state)

    Set state information of this broker.
- boolean initialize (ArrayList< String > argvs)

    Initialization of the broker-side.
- boolean sendException (Combox combox, POPException exception)

    Send exception to the interface-side.
- void sendResponse (Combox combox, POPBuffer buffer)

Send response to the interface-side.

- String getLogPrefix ()

  Return the prefix for log file.

Static Public Member Functions

- static Broker **getBroker** ()
- static POPAccessPoint getAccessPoint ()

  Return the access point of this broker.
- static void main (String[ ] argvs) throws InterruptedException

  Entry point for the Broker.

Static Public Attributes

- static final int **Running** = 0
- static final int **Exit** = 1
- static final int **Abort** = 2
- static final int **REQUEST_QUEUE_TIMEOUT_MS** = 600
- static final int **BasicCallMaxRange** = 10
- static final int **ConstructorSemanticId** = 21
- static final String **CallBackPrefix** = "-callback="
- static final String **CodeLocationPrefix** = "-codelocation="
- static final String **ObjectNamePrefix** = "-object="
- static final String **ActualObjectNamePrefix** = "-actualobject="
- static final String **AppServicePrefix** = "-appservice="

Protected Member Functions

- boolean findEndcoding (String encoding)

  Look for a specific encoding.
- Class<?> getPOPObjectClass (String className, URLClassLoader urlClassLoader) throws ClassNot-FoundException

  Return the class of the associated object.

Protected Attributes

- int **state**
- ComboxServer **comboxServer**
- POPBuffer **buffer**
- POPObject **popObject** = null
- POPObject **popInfo** = null
- int **connectionCount** = 0
- Semaphore **sequentialSemaphore** = new Semaphore(1, true)

Static Protected Attributes

- static POPAccessPoint **accessPoint** = new POPAccessPoint()

5.4.1 Detailed Description

This class is the base class of all broker-side parallel object.

The broker is responsible to receive the requests from the interface-side and to execute them on the real object

5.4.2 Member Function Documentation

5.4.2.1 void popjava.broker.Broker.clearResourceAfterInvoke (**Request** *request*)

Remove the request from the request queue after invocation.

Parameters

request	Request to be removed

Here is the caller graph for this function:

5.4.2.2 boolean popjava.broker.Broker.findEndcoding (String *encoding*) `[protected]`

Look for a specific encoding.

Parameters

encoding	Encoding to look for

Returns

true if the encoding is available

Here is the caller graph for this function:

5.4.2.3 static **POPAccessPoint** popjava.broker.Broker.getAccessPoint ( ) `[static]`

Return the access point of this broker.

Returns

Access point associated with this broker

Here is the caller graph for this function:

5.4.2.4 String popjava.broker.Broker.getLogPrefix ( )

Return the prefix for log file.

Returns

log prefix

5.4.2.5 Class<?> popjava.broker.Broker.getPOPObjectClass (String *className,* URLClassLoader *urlClassLoader*) throws ClassNotFoundException `[protected]`

Return the class of the associated object.

Parameters

className	Name of the class
urlClassLoader	Path of the class

Returns

Class object or null

Exceptions

ClassNotFoundException	thrown if the class is not found

5.4.2.6 synchronized int popjava.broker.Broker.getState ( )

Get information about the state of this borker.

Returns

current state

Here is the call graph for this function:

Here is the caller graph for this function:

5.4.2.7 boolean popjava.broker.Broker.initialize (ArrayList< String > *argvs*)

Initialization of the broker-side.

Parameters

argvs	Arguments

Returns

true if the initialization process succeed

Here is the call graph for this function:

Here is the caller graph for this function:

5.4.2.8 boolean popjava.broker.Broker.invoke (Request *request*) throws InterruptedException

This method is responsible to dispatch the request between invokeConstructor and invokeMethod.

Parameters

request	Request received from the interface-side

Returns

true if the request has been treated correctly

Exceptions

InterruptedException	

Here is the call graph for this function:

Here is the caller graph for this function:

5.4.2.9 boolean popjava.broker.Broker.isDaemon ( )

Get information about the deamon mode of this broker.

Returns

deamon mode

Here is the call graph for this function:

Here is the caller graph for this function:

5.4.2.10 static void popjava.broker.Broker.main (String[ ] *argvs*) throws InterruptedException [static]

Entry point for the Broker.

This method is called when a new Broker is setup in a JVM.

Parameters

argvs	arguments of the program

Exceptions

InterruptedException	

Here is the call graph for this function:

5.4.2.11 boolean popjava.broker.Broker.popCall (Request *request*)

This method is responsible to handle the POP system call.

Parameters

request	Request received from the interface-side

Returns

true if the request has been treated correctly

Here is the call graph for this function:

Here is the caller graph for this function:

5.4.2.12 boolean popjava.broker.Broker.sendException (Combox *combox,* POPException *exception*)

Send exception to the interface-side.

Parameters

combox	Combox to send the exception
exception	Exception to send

Returns

true if the exception has been sent

Here is the call graph for this function:

5.4.2.13 void popjava.broker.Broker.sendResponse (Combox *combox,* POPBuffer *buffer*)

Send response to the interface-side.

Parameters

combox	Combox to send the response
buffer	Buffer to send trough the combox

Here is the caller graph for this function:

5.4.2.14 void popjava.broker.Broker.serveRequest (final Request *request*) throws InterruptedException

This method is responsible to handle the broker-side semantics for a request.

Parameters

request	Request received from the interface-side

Exceptions

InterruptedException	

Here is the call graph for this function:

Here is the caller graph for this function:

5.4.2.15 synchronized void popjava.broker.Broker.setState (int *state*)

Set state information of this broker.

Parameters

state	state to set to this broker

Here is the caller graph for this function:

5.4.2.16 void popjava.broker.Broker.treatRequests ( ) throws InterruptedException

Main loop of this broker.

Exceptions

InterruptedException	

Here is the call graph for this function:

Here is the caller graph for this function:

5.5 popjava.buffer.BufferFactory Class Reference

This abstract class defined all the methods needed by a BufferFactory.

Inheritance diagram for popjava.buffer.BufferFactory:

Collaboration diagram for popjava.buffer.BufferFactory:

Public Member Functions

- abstract POPBuffer createBuffer ()

    Creates a new instance of BufferFactory.
- abstract String getBufferName ()

    Return buffer's names.

5.5.1 Detailed Description

This abstract class defined all the methods needed by a BufferFactory.

5.5.2 Member Function Documentation

5.5.2.1 abstract String popjava.buffer.BufferFactory.getBufferName ( ) `[pure virtual]`

Return buffer's names.

Returns

name of the buffers

Implemented in popjava.buffer.BufferRawFactory, popjava.buffer.BufferXDRFactory, and popjava.buffer.BufferFactoryPlugin.

5.6 popjava.buffer.BufferFactoryFinder Class Reference

This class is responsible to discover the buffer.

Collaboration diagram for popjava.buffer.BufferFactoryFinder:

Public Member Functions

- void loadBufferMap (String pluginLocation)

    Read the plugin file.
- BufferFactory findFactory (String factoryName)

    Find a specific factory.
- String getSupportingBuffer ()

    Get a formatted string of supporting buffer.

Static Public Member Functions

- static BufferFactoryFinder getInstance ()

    Get the unique instance of the BufferFactoryFinder.

Protected Member Functions

- BufferFactoryFinder ()

  Default constructor.

5.6.1 Detailed Description

This class is responsible to discover the buffer.

5.6.2 Member Function Documentation

5.6.2.1 BufferFactory popjava.buffer.BufferFactoryFinder.findFactory (String *factoryName*)

Find a specific factory.

Parameters

factoryName	Name of the factory

Returns

The factory found or null

5.6.2.2 static BufferFactoryFinder popjava.buffer.BufferFactoryFinder.getInstance ( ) [static]

Get the unique instance of the BufferFactoryFinder.

Returns

the unique instance of the BufferFactoryFinder

Here is the call graph for this function:

Here is the caller graph for this function:

5.6.2.3 String popjava.buffer.BufferFactoryFinder.getSupportingBuffer ( )

Get a formatted string of supporting buffer.

Returns

formatted string of supporting buffer

Here is the caller graph for this function:

5.6.2.4 void popjava.buffer.BufferFactoryFinder.loadBufferMap (String *pluginLocation*)

Read the plugin file.

Parameters

pluginLocation	Location of the plugin file

Here is the call graph for this function:

Here is the caller graph for this function:

5.7 popjava.buffer.BufferFactoryPlugin Class Reference

This class defined the interface for new buffer factory plug-in.

Inheritance diagram for popjava.buffer.BufferFactoryPlugin:

Collaboration diagram for popjava.buffer.BufferFactoryPlugin:

Public Member Functions

- POPBuffer createBuffer ()

  Creates a new instance of BufferFactory.
- String getBufferName ()

  Return buffer's names.

5.7.1 Detailed Description

This class defined the interface for new buffer factory plug-in.

5.7.2 Member Function Documentation

5.7.2.1 String popjava.buffer.BufferFactoryPlugin.getBufferName ( ) `[virtual]`

Return buffer's names.

Returns

name of the buffers

Implements popjava.buffer.BufferFactory.

5.8 popjava.buffer.BufferPlugin Class Reference

This class defined the interface for each new buffer plug-in.

Inheritance diagram for popjava.buffer.BufferPlugin:

Collaboration diagram for popjava.buffer.BufferPlugin:

Public Member Functions

- byte[] **array** ()
- MessageHeader extractHeader ()

  Retrieve the message header from the buffer.
- byte get ()

  Retrieve a byte from the buffer.
- boolean getBoolean ()

  Retrieve a boolean from the buffer.
- boolean[] getBooleanArray (int length)

  Retrieve a boolean array from the buffer.

- byte[] getByteArray (int length)

  Retrieve a byte array from the buffer.
- char getChar ()

  Retrieve a char from the buffer.
- double getDouble ()

  Retrieve a double from the buffer.
- double[] getDoubleArray (int length)

  Retrieve a double array from the buffer.
- float getFloat ()

  Retrieve a float from the buffer.
- float[] getFloatArray (int length)

  Retrieve a float array from the buffer.
- int getInt ()

  Retrieve a int from the buffer.
- int[] getIntArray (int length)

  Retrieve a int array from the buffer.
- long getLong ()

  Retrieve a long from the buffer.
- long[] getLongArray (int length)

  Retrieve a long array from the buffer.
- String getString ()

  Retrieve a string from the buffer.
- void put (byte value)

  Insert a byte in the buffer.
- void put (byte[] data)

  Insert a byte array into the buffer.
- void put (byte[] data, int offset, int length)

  Insert a byte array into a specific place in the buffer.
- void putBoolean (boolean value)

  Insert a boolean in the buffer.
- void putBooleanArray (boolean[] value)

  Insert a boolean array into the buffer.
- void putByteArray (byte[] value)

  Insert a byte array into the buffer.
- void putChar (char value)

  Insert a char into the buffer.
- void putDouble (double value)

  Insert a double value into the buffer.
- void putDoubleArray (double[] value)

  Insert a double array into the buffer.
- void putFloat (float value)

  Insert a float value into the buffer.
- void putFloatArray (float[] value)

  Insert a float array into the buffer.
- void putInt (int value)

  Insert a int into the buffer.
- void putIntArray (int[] value)

  Insert a int array into the buffer.
- void putLong (long value)

  Insert a long into the buffer.
- void putLongArray (long[] value)

Insert a long array into the buffer.
- void putString (String value)

    Insert a string into the buffer.
- void reset ()

    Erase the buffer and set the pointer to the beginning.
- void resetToReceive ()

    Reset the buffer before reception of a new message.
- int getTranslatedInteger (byte[] value)

    Get a integer value of the byte array.
- int packMessageHeader ()

    Pack the message header into the buffer.
- short getShort ()

    Retrieve a short from the buffer.
- short[] getShortArray (int length)

    Retrieve a short array from the buffer.
- void putShort (short value)

    Insert a short into the buffer.
- void putShortArray (short[] value)

    Insert a short array into the buffer.
- char[] getCharArray (int length)

    Retrieve a char array from the buffer.
- void putCharArray (char[] value)

    Insert a char array into the buffer.

Additional Inherited Members

5.8.1 Detailed Description

This class defined the interface for each new buffer plug-in.

5.8.2 Member Function Documentation

5.8.2.1 MessageHeader popjava.buffer.BufferPlugin.extractHeader ( ) [virtual]

Retrieve the message header from the buffer.

Returns

message header retrieved in the buffer

Implements popjava.buffer.POPBuffer.

5.8.2.2 byte popjava.buffer.BufferPlugin.get ( ) [virtual]

Retrieve a byte from the buffer.

Returns

byte retrieved in the buffer

Implements popjava.buffer.POPBuffer.

5.8.2.3 boolean popjava.buffer.BufferPlugin.getBoolean ( ) `[virtual]`

Retrieve a boolean from the buffer.

Returns

boolean retrieved in the buffer

Implements popjava.buffer.POPBuffer.

5.8.2.4 boolean [ ] popjava.buffer.BufferPlugin.getBooleanArray (int *length*) `[virtual]`

Retrieve a boolean array from the buffer.

Parameters

length	length of the array to retrieve

Returns

boolean array retrieved in the buffer

Implements popjava.buffer.POPBuffer.

5.8.2.5 byte [ ] popjava.buffer.BufferPlugin.getByteArray (int *length*) `[virtual]`

Retrieve a byte array from the buffer.

Parameters

length	length of the array to retrieve

Returns

byte array retrieved in the buffer

Implements popjava.buffer.POPBuffer.

5.8.2.6 char popjava.buffer.BufferPlugin.getChar ( ) `[virtual]`

Retrieve a char from the buffer.

Returns

char retrieved in the buffer

Implements popjava.buffer.POPBuffer.

5.8.2.7 char [ ] popjava.buffer.BufferPlugin.getCharArray (int *length*) `[virtual]`

Retrieve a char array from the buffer.

Parameters

length	length of the array to retrieve

Returns

char array retrieved in the buffer

Implements popjava.buffer.POPBuffer.

5.8.2.8 double popjava.buffer.BufferPlugin.getDouble ( ) [virtual]

Retrieve a double from the buffer.

Returns

double retrieved in the buffer

Implements popjava.buffer.POPBuffer.

5.8.2.9 double [] popjava.buffer.BufferPlugin.getDoubleArray (int *length*) [virtual]

Retrieve a double array from the buffer.

Parameters

length	length of the array to retrieve

Returns

double array retrieved in the buffer

Implements popjava.buffer.POPBuffer.

5.8.2.10 float popjava.buffer.BufferPlugin.getFloat ( ) [virtual]

Retrieve a float from the buffer.

Returns

float retrieved in the buffer

Implements popjava.buffer.POPBuffer.

5.8.2.11 float [] popjava.buffer.BufferPlugin.getFloatArray (int *length*) [virtual]

Retrieve a float array from the buffer.

Parameters

length	length of the array to retrieve

Returns

float array retrieved in the buffer

Implements popjava.buffer.POPBuffer.

5.8.2.12 int popjava.buffer.BufferPlugin.getInt ( ) [virtual]

Retrieve a int from the buffer.

Returns

int retrieved in the buffer

Implements popjava.buffer.POPBuffer.

5.8.2.13 int [] popjava.buffer.BufferPlugin.getIntArray (int *length*) [virtual]

Retrieve a int array from the buffer.

Parameters

length	length of the array to retrieve

Returns

int array retrieved in the buffer

Implements popjava.buffer.POPBuffer.

5.8.2.14 long popjava.buffer.BufferPlugin.getLong ( ) [virtual]

Retrieve a long from the buffer.

Returns

long retrieved in the buffer

Implements popjava.buffer.POPBuffer.

5.8.2.15 long [] popjava.buffer.BufferPlugin.getLongArray (int *length*) [virtual]

Retrieve a long array from the buffer.

Parameters

length	length of the array to retrieve

Returns

long array retrieved in the buffer

Implements popjava.buffer.POPBuffer.

5.8.2.16 short popjava.buffer.BufferPlugin.getShort ( ) [virtual]

Retrieve a short from the buffer.

Returns

short retrieved in the buffer

Implements popjava.buffer.POPBuffer.

5.8.2.17 short [] popjava.buffer.BufferPlugin.getShortArray (int *length*) [virtual]

Retrieve a short array from the buffer.

Parameters

length	length of the array to retrieve

Returns

short array retrieved in the buffer

Implements popjava.buffer.POPBuffer.

5.8.2.18 String popjava.buffer.BufferPlugin.getString ( ) [virtual]

Retrieve a string from the buffer.

Returns

string retrieved in the buffer

Implements popjava.buffer.POPBuffer.

5.8.2.19 int popjava.buffer.BufferPlugin.getTranslatedInteger (byte[] *value*) [virtual]

Get a integer value of the byte array.

Parameters

value	The byte array to translate

Returns

The integer

Implements popjava.buffer.POPBuffer.

5.8.2.20 int popjava.buffer.BufferPlugin.packMessageHeader ( ) [virtual]

Pack the message header into the buffer.

Returns

number of byte used for the message header

Implements popjava.buffer.POPBuffer.

5.8.2.21 void popjava.buffer.BufferPlugin.put (byte *value*) [virtual]

Insert a byte in the buffer.

Parameters

value	byte value to insert

Implements popjava.buffer.POPBuffer.

5.8.2.22 void popjava.buffer.BufferPlugin.put (byte[] *data*) [virtual]

Insert a byte array into the buffer.

Parameters

data	byte array to insert

Implements popjava.buffer.POPBuffer.

5.8.2.23 void popjava.buffer.BufferPlugin.put (byte[] *data*, int *offset*, int *length*) [virtual]

Insert a byte array into a specific place in the buffer.

Parameters

data	byte array to insert
offset	offset for insertion
length	length of the array

Implements popjava.buffer.POPBuffer.

5.8.2.24 void popjava.buffer.BufferPlugin.putBoolean (boolean *value*) [virtual]

Insert a boolean in the buffer.

Parameters

value	boolean value to insert

Implements popjava.buffer.POPBuffer.

5.8.2.25 void popjava.buffer.BufferPlugin.putBooleanArray (boolean[] *value*) [virtual]

Insert a boolean array into the buffer.

Parameters

value	boolean array to insert

Implements popjava.buffer.POPBuffer.

5.8.2.26 void popjava.buffer.BufferPlugin.putByteArray (byte[] *value*) [virtual]

Insert a byte array into the buffer.

Parameters

value	byte array to insert

Implements popjava.buffer.POPBuffer.

5.8.2.27 void popjava.buffer.BufferPlugin.putChar (char *value*) [virtual]

Insert a char into the buffer.

Parameters

value	char value to insert

Implements popjava.buffer.POPBuffer.

5.8.2.28 void popjava.buffer.BufferPlugin.putCharArray (char[] *value*) [virtual]

Insert a char array into the buffer.

Parameters

value	char array to insert

Implements popjava.buffer.POPBuffer.

5.8.2.29 void popjava.buffer.BufferPlugin.putDouble (double *value*) [virtual]

Insert a double value into the buffer.

Parameters

value	double value to insert

Implements popjava.buffer.POPBuffer.

5.8.2.30 void popjava.buffer.BufferPlugin.putDoubleArray (double[] *value*) [virtual]

Insert a double array into the buffer.

Parameters

value	double array to insert

Implements popjava.buffer.POPBuffer.

5.8.2.31 void popjava.buffer.BufferPlugin.putFloat (float *value*) [virtual]

Insert a float value into the buffer.

Parameters

value	float value to insert

Implements popjava.buffer.POPBuffer.

5.8.2.32 void popjava.buffer.BufferPlugin.putFloatArray (float[] *value*) [virtual]

Insert a float array into the buffer.

Parameters

value	float array to insert

Implements popjava.buffer.POPBuffer.

5.8.2.33 void popjava.buffer.BufferPlugin.putInt (int *value*) [virtual]

Insert a int into the buffer.

Parameters

value	int value to insert

Implements popjava.buffer.POPBuffer.

5.8.2.34 void popjava.buffer.BufferPlugin.putIntArray (int[] *value*) [virtual]

Insert a int array into the buffer.

Parameters

value	int array to insert

Implements popjava.buffer.POPBuffer.

5.8.2.35 void popjava.buffer.BufferPlugin.putLong (long *value*) [virtual]

Insert a long into the buffer.

Parameters

value	long value to insert

Implements popjava.buffer.POPBuffer.

5.8.2.36 void popjava.buffer.BufferPlugin.putLongArray (long[] *value*) [virtual]

Insert a long array into the buffer.

Parameters

value	long array to insert

Implements popjava.buffer.POPBuffer.

5.8.2.37 void popjava.buffer.BufferPlugin.putShort (short *value*) [virtual]

Insert a short into the buffer.

Parameters

value	short value to insert

Implements popjava.buffer.POPBuffer.

5.8.2.38 void popjava.buffer.BufferPlugin.putShortArray (short[] *value*) [virtual]

Insert a short array into the buffer.

Parameters

value	short array to insert

Implements popjava.buffer.POPBuffer.

5.8.2.39 void popjava.buffer.BufferPlugin.putString (String *value*) [virtual]

Insert a string into the buffer.

Parameters

value	string value to insert

Implements popjava.buffer.POPBuffer.

5.9 popjava.buffer.BufferRaw Class Reference

This class is a RAW implementation of the buffer abstract class.

Inheritance diagram for popjava.buffer.BufferRaw:

Collaboration diagram for popjava.buffer.BufferRaw:

Public Member Functions

- BufferRaw ()
  Default constructor.
- BufferRaw (MessageHeader messageHeader)
  Constructor with given values.
- byte[] **array** ()
- MessageHeader extractHeader ()
  Retrieve the message header from the buffer.
- boolean getBoolean ()
  Retrieve a boolean from the buffer.
- float getFloat ()
  Retrieve a float from the buffer.
- int getInt ()
  Retrieve a int from the buffer.
- int getInt (int index)
  Get int value at the specified index.
- char getChar ()
  Retrieve a char from the buffer.
- double getDouble ()
  Retrieve a double from the buffer.
- long getLong ()
  Retrieve a long from the buffer.
- String getString ()
  Retrieve a string from the buffer.

- void put (byte value)

  Insert a byte in the buffer.
- void put (byte[ ] data)

  Insert a byte array into the buffer.
- void put (byte[ ] data, int offset, int length)

  Insert a byte array into a specific place in the buffer.
- void putBoolean (boolean value)

  Insert a boolean in the buffer.
- void putChar (char value)

  Insert a char into the buffer.
- void putFloat (float value)

  Insert a float value into the buffer.
- void putInt (int value)

  Insert a int into the buffer.
- void putInt (int index, int value)

  Insert int value at a specified index in the buffer.
- void putDouble (double value)

  Insert a double value into the buffer.
- void putLong (long value)

  Insert a long into the buffer.
- void putString (String data)

  Insert a string into the buffer.
- void reset ()

  Erase the buffer and set the pointer to the beginning.
- void resetToReceive ()

  Reset the buffer before reception of a new message.
- int getTranslatedInteger (byte[ ] value)

  Get a integer value of the byte array.
- String **toIntString** ()
- String **toCharString** ()
- int position ()

  Get the current buffer's position.
- void position (int index)

  Set the pointer to the index.
- void resize (int moreCapacity)

  Resize the current buffer to store more data.
- void **resize** (int position, int moreCapacity)
- void putBooleanArray (boolean[ ] value)

  Insert a boolean array into the buffer.
- void putDoubleArray (double[ ] value)

  Insert a double array into the buffer.
- void putFloatArray (float[ ] value)

  Insert a float array into the buffer.
- void putIntArray (int[ ] value)

  Insert a int array into the buffer.
- void putLongArray (long[ ] value)

  Insert a long array into the buffer.
- byte get ()

  Retrieve a byte from the buffer.
- boolean[ ] getBooleanArray (int length)

  Retrieve a boolean array from the buffer.

- byte[] getByteArray (int length)

  Retrieve a byte array from the buffer.
- double[] getDoubleArray (int length)

  Retrieve a double array from the buffer.
- float[] getFloatArray (int length)

  Retrieve a float array from the buffer.
- int[] getIntArray (int length)

  Retrieve a int array from the buffer.
- long[] getLongArray (int length)

  Retrieve a long array from the buffer.
- void putByteArray (byte[] value)

  Insert a byte array into the buffer.
- int packMessageHeader ()

  Pack the message header into the buffer.
- short getShort ()

  Retrieve a short from the buffer.
- void putShort (short value)

  Insert a short into the buffer.
- short[] getShortArray (int length)

  Retrieve a short array from the buffer.
- void putShortArray (short[] value)

  Insert a short array into the buffer.
- char[] getCharArray (int length)

  Retrieve a char array from the buffer.
- void putCharArray (char[] value)

  Insert a char array into the buffer.

Static Public Attributes

- static final int BufferLength = 20000

  Size of the buffer.

Protected Member Functions

- int limit ()

  Return the buffer's limit.

Protected Attributes

- ByteBuffer buffer

  Byte buffer to store data.

Additional Inherited Members

5.9.1 Detailed Description

This class is a RAW implementation of the buffer abstract class.

5.9.2 Constructor & Destructor Documentation

5.9.2.1 popjava.buffer.BufferRaw.BufferRaw (**MessageHeader** *messageHeader*)

Constructor with given values.

Parameters

messageHeader	Message header to be associated with this buffer

Here is the call graph for this function:

5.9.3 Member Function Documentation

5.9.3.1 **MessageHeader** popjava.buffer.BufferRaw.extractHeader ( ) `[virtual]`

Retrieve the message header from the buffer.

Returns

message header retrieved in the buffer

Implements popjava.buffer.POPBuffer.

Here is the call graph for this function:

5.9.3.2 byte popjava.buffer.BufferRaw.get ( ) `[virtual]`

Retrieve a byte from the buffer.

Returns

byte retrieved in the buffer

Implements popjava.buffer.POPBuffer.

5.9.3.3 boolean popjava.buffer.BufferRaw.getBoolean ( ) `[virtual]`

Retrieve a boolean from the buffer.

Returns

boolean retrieved in the buffer

Implements popjava.buffer.POPBuffer.

5.9.3.4 boolean [] popjava.buffer.BufferRaw.getBooleanArray (int *length*) `[virtual]`

Retrieve a boolean array from the buffer.

Parameters

length	length of the array to retrieve

Returns

boolean array retrieved in the buffer

Implements popjava.buffer.POPBuffer.

Here is the call graph for this function:

5.9.3.5 byte [ ] popjava.buffer.BufferRaw.getByteArray (int *length*) [virtual]

Retrieve a byte array from the buffer.

Parameters

length	length of the array to retrieve

Returns

byte array retrieved in the buffer

Implements popjava.buffer.POPBuffer.

Here is the call graph for this function:

5.9.3.6 char popjava.buffer.BufferRaw.getChar ( ) [virtual]

Retrieve a char from the buffer.

Returns

char retrieved in the buffer

Implements popjava.buffer.POPBuffer.

Here is the call graph for this function:

5.9.3.7 char [ ] popjava.buffer.BufferRaw.getCharArray (int *length*) [virtual]

Retrieve a char array from the buffer.

Parameters

length	length of the array to retrieve

Returns

char array retrieved in the buffer

Implements popjava.buffer.POPBuffer.

5.9.3.8 double popjava.buffer.BufferRaw.getDouble ( ) [virtual]

Retrieve a double from the buffer.

Returns

double retrieved in the buffer

Implements popjava.buffer.POPBuffer.

5.9.3.9 double [ ] popjava.buffer.BufferRaw.getDoubleArray (int *length*) `[virtual]`

Retrieve a double array from the buffer.

Parameters

length	length of the array to retrieve

Returns

double array retrieved in the buffer

Implements popjava.buffer.POPBuffer.

Here is the call graph for this function:

5.9.3.10 float popjava.buffer.BufferRaw.getFloat ( ) `[virtual]`

Retrieve a float from the buffer.

Returns

float retrieved in the buffer

Implements popjava.buffer.POPBuffer.

5.9.3.11 float [ ] popjava.buffer.BufferRaw.getFloatArray (int *length*) `[virtual]`

Retrieve a float array from the buffer.

Parameters

length	length of the array to retrieve

Returns

float array retrieved in the buffer

Implements popjava.buffer.POPBuffer.

Here is the call graph for this function:

5.9.3.12 int popjava.buffer.BufferRaw.getInt ( ) `[virtual]`

Retrieve a int from the buffer.

Returns

int retrieved in the buffer

Implements popjava.buffer.POPBuffer.

Here is the caller graph for this function:

5.9.3.13 int popjava.buffer.BufferRaw.getInt (int *index*)

Get int value at the specified index.

Parameters

index	index of the value

Returns

the int value

5.9.3.14 int [ ] popjava.buffer.BufferRaw.getIntArray (int *length*) [virtual]

Retrieve a int array from the buffer.

Parameters

length	length of the array to retrieve

Returns

int array retrieved in the buffer

Implements popjava.buffer.POPBuffer.

Here is the call graph for this function:

Here is the caller graph for this function:

5.9.3.15 long popjava.buffer.BufferRaw.getLong ( ) [virtual]

Retrieve a long from the buffer.

Returns

long retrieved in the buffer

Implements popjava.buffer.POPBuffer.

5.9.3.16 long [ ] popjava.buffer.BufferRaw.getLongArray (int *length*) [virtual]

Retrieve a long array from the buffer.

Parameters

length	length of the array to retrieve

Returns

long array retrieved in the buffer

Implements popjava.buffer.POPBuffer.

Here is the call graph for this function:

5.9.3.17 short popjava.buffer.BufferRaw.getShort ( ) [virtual]

Retrieve a short from the buffer.

Returns

short retrieved in the buffer

Implements popjava.buffer.POPBuffer.

5.9.3.18 short [ ] popjava.buffer.BufferRaw.getShortArray (int *length*) `[virtual]`

Retrieve a short array from the buffer.

Parameters

length	length of the array to retrieve

Returns

short array retrieved in the buffer

Implements popjava.buffer.POPBuffer.

Here is the call graph for this function:

5.9.3.19 String popjava.buffer.BufferRaw.getString ( ) `[virtual]`

Retrieve a string from the buffer.

Returns

string retrieved in the buffer

Implements popjava.buffer.POPBuffer.

Here is the call graph for this function:

5.9.3.20 int popjava.buffer.BufferRaw.getTranslatedInteger (byte[ ] *value*) `[virtual]`

Get a integer value of the byte array.

Parameters

value	The byte array to translate

Returns

The integer

Implements popjava.buffer.POPBuffer.

Reimplemented in popjava.buffer.BufferXDR.

5.9.3.21 int popjava.buffer.BufferRaw.limit ( ) `[protected]`

Return the buffer's limit.

Returns

the limit of this buffer

5.9.3.22 int popjava.buffer.BufferRaw.packMessageHeader ( ) `[virtual]`

Pack the message header into the buffer.

Returns

number of byte used for the message header

Implements popjava.buffer.POPBuffer.

Here is the call graph for this function:

5.9.3.23 int popjava.buffer.BufferRaw.position ( )

Get the current buffer's position.

Returns

the buffer's position

Here is the caller graph for this function:

5.9.3.24 void popjava.buffer.BufferRaw.position (int *index*)

Set the pointer to the index.

Parameters

index	index to set the pointer

Here is the call graph for this function:

5.9.3.25 void popjava.buffer.BufferRaw.put (byte *value*) `[virtual]`

Insert a byte in the buffer.

Parameters

value	byte value to insert

Implements popjava.buffer.POPBuffer.

Here is the call graph for this function:

Here is the caller graph for this function:

5.9.3.26 void popjava.buffer.BufferRaw.put (byte[] *data*) `[virtual]`

Insert a byte array into the buffer.

Parameters

data	byte array to insert

Implements popjava.buffer.POPBuffer.

Here is the call graph for this function:

5.9.3.27 void popjava.buffer.BufferRaw.put (byte[] *data*, int *offset*, int *length*) [virtual]

Insert a byte array into a specific place in the buffer.

Parameters

data	byte array to insert
offset	offset for insertion
length	length of the array

Implements popjava.buffer.POPBuffer.

Here is the call graph for this function:

5.9.3.28 void popjava.buffer.BufferRaw.putBoolean (boolean *value*) [virtual]

Insert a boolean in the buffer.

Parameters

value	boolean value to insert

Implements popjava.buffer.POPBuffer.

Reimplemented in popjava.buffer.BufferXDR.

Here is the call graph for this function:

5.9.3.29 void popjava.buffer.BufferRaw.putBooleanArray (boolean[] *value*) [virtual]

Insert a boolean array into the buffer.

Parameters

value	boolean array to insert

Implements popjava.buffer.POPBuffer.

Here is the call graph for this function:

5.9.3.30 void popjava.buffer.BufferRaw.putByteArray (byte[] *value*) [virtual]

Insert a byte array into the buffer.

Parameters

value	byte array to insert

Implements popjava.buffer.POPBuffer.

Here is the call graph for this function:

5.9.3.31 void popjava.buffer.BufferRaw.putChar (char *value*) [virtual]

Insert a char into the buffer.

Parameters

value	char value to insert

Implements popjava.buffer.POPBuffer.

Here is the call graph for this function:

5.9.3.32 void popjava.buffer.BufferRaw.putCharArray (char[] *value*) [virtual]

Insert a char array into the buffer.

Parameters

value	char array to insert

Implements popjava.buffer.POPBuffer.

Here is the call graph for this function:

5.9.3.33 void popjava.buffer.BufferRaw.putDouble (double *value*) [virtual]

Insert a double value into the buffer.

Parameters

value	double value to insert

Implements popjava.buffer.POPBuffer.

Here is the call graph for this function:

5.9.3.34 void popjava.buffer.BufferRaw.putDoubleArray (double[] *value*) [virtual]

Insert a double array into the buffer.

Parameters

value	double array to insert

Implements popjava.buffer.POPBuffer.

Here is the call graph for this function:

5.9.3.35 void popjava.buffer.BufferRaw.putFloat (float *value*) [virtual]

Insert a float value into the buffer.

Parameters

value	float value to insert

Implements popjava.buffer.POPBuffer.

Here is the call graph for this function:

5.9.3.36 void popjava.buffer.BufferRaw.putFloatArray (float[] *value*) [virtual]

Insert a float array into the buffer.

Parameters

value	float array to insert

Implements popjava.buffer.POPBuffer.

Here is the call graph for this function:

5.9.3.37 void popjava.buffer.BufferRaw.putInt (int *value*) [virtual]

Insert a int into the buffer.

Parameters

value	int value to insert

Implements popjava.buffer.POPBuffer.

Here is the call graph for this function:

Here is the caller graph for this function:

5.9.3.38 void popjava.buffer.BufferRaw.putInt (int *index,* int *value*)

Insert int value at a specified index in the buffer.

Parameters

index	index to put the value
value	the int value to be inserted

Here is the call graph for this function:

5.9.3.39 void popjava.buffer.BufferRaw.putIntArray (int[] *value*) [virtual]

Insert a int array into the buffer.

Parameters

value	int array to insert

Implements popjava.buffer.POPBuffer.

Here is the call graph for this function:

Here is the caller graph for this function:

5.9.3.40 void popjava.buffer.BufferRaw.putLong (long *value*) [virtual]

Insert a long into the buffer.

Parameters

value	long value to insert

Implements popjava.buffer.POPBuffer.

Here is the call graph for this function:

5.9.3.41 void popjava.buffer.BufferRaw.putLongArray (long[] *value*) [virtual]

Insert a long array into the buffer.

Parameters

value	long array to insert

Implements popjava.buffer.POPBuffer.

Here is the call graph for this function:

5.9.3.42 void popjava.buffer.BufferRaw.putShort (short *value*) [virtual]

Insert a short into the buffer.

Parameters

value	short value to insert

Implements popjava.buffer.POPBuffer.

Here is the call graph for this function:

5.9.3.43 void popjava.buffer.BufferRaw.putShortArray (short[] *value*) [virtual]

Insert a short array into the buffer.

Parameters

value	short array to insert

Implements popjava.buffer.POPBuffer.

Here is the call graph for this function:

5.9.3.44 void popjava.buffer.BufferRaw.putString (String *value*) [virtual]

Insert a string into the buffer.

Parameters

value	string value to insert

Implements popjava.buffer.POPBuffer.

Here is the call graph for this function:

Here is the caller graph for this function:

5.9.3.45 void popjava.buffer.BufferRaw.resize (int *moreCapacity*)

Resize the current buffer to store more data.

Parameters

moreCapacity	The additional capacity to add on the current buffer

Here is the call graph for this function:

Here is the caller graph for this function:

5.10 popjava.buffer.BufferRawFactory Class Reference

Implementation of the abstract BufferFactory for the RAW encoding.

Inheritance diagram for popjava.buffer.BufferRawFactory:

Collaboration diagram for popjava.buffer.BufferRawFactory:

Public Member Functions

- POPBuffer createBuffer ()

  Create a new RAW factory.
- String getBufferName ()

  Get the identifier of this factory.

Public Attributes

- final String BufferName = "raw"

  Identifier of this buffer.

5.10.1 Detailed Description

Implementation of the abstract BufferFactory for the RAW encoding.

5.11 popjava.buffer.BufferXDR Class Reference

This class is a XDR extension of the BufferRAW class.

Inheritance diagram for popjava.buffer.BufferXDR:

Collaboration diagram for popjava.buffer.BufferXDR:

Public Member Functions

- BufferXDR ()

  Default constructor.
- BufferXDR (MessageHeader messageHeader)

  Constructor with given values.
- void putBoolean (boolean value)

  Insert a boolean value.
- int getTranslatedInteger (byte[ ] value)

  Transfirm an integer.

Additional Inherited Members

5.11.1 Detailed Description

This class is a XDR extension of the BufferRAW class.

5.11.2 Constructor & Destructor Documentation

5.11.2.1 popjava.buffer.BufferXDR.BufferXDR ( )

Default constructor.

Create a new instance of XDR buffer

5.11.2.2 popjava.buffer.BufferXDR.BufferXDR (MessageHeader *messageHeader*)

Constructor with given values.

Parameters

messageHeader	Message header to be associated with this buffer

5.11.3 Member Function Documentation

5.11.3.1 void popjava.buffer.BufferXDR.putBoolean (boolean *value*) [virtual]

Insert a boolean value.

Parameters

value	The boolean value to be inserted

Reimplemented from popjava.buffer.BufferRaw.

5.12 popjava.buffer.BufferXDRFactory Class Reference

Implementation of the abstract BufferFactory for the RAW encoding.

Inheritance diagram for popjava.buffer.BufferXDRFactory:

Collaboration diagram for popjava.buffer.BufferXDRFactory:

Public Member Functions

- POPBuffer createBuffer ()

  Create a new XDR Buffer.
- String getBufferName ()

  Get the identifier of this buffer factory.

Public Attributes

- final String BufferName = "xdr"

  Identifier of this buffer.

5.12.1 Detailed Description

Implementation of the abstract BufferFactory for the RAW encoding.

5.13 popjava.util.ClassUtil Class Reference

This class gives some static methods to look inside a class.

Collaboration diagram for popjava.util.ClassUtil:

Static Public Member Functions

- static Class<?>[] **getObjectTypes** (Object...objects)
- static Constructor<?> getConstructor (Class<?> c, Class<?>...parameterTypes) throws NoSuchMethod-Exception

  Retrieve a specific constructor in the given class.
- static Method getMethod (Class<?> c, String methodName, Class<?>...parameterTypes) throws NoSuch-MethodException

  Retrieve a specific method in the given class.
- static String getMethodSign (Method m)

  Get the signature of a method.
- static String getMethodSign (Constructor<?> c)

  Get the signature of a constructor.
- static String **getMethodSign** (String name, Class<?>[] parameterTypes)
- static Object getDefaultPrimitiveValue (Class<?> c)

  Get a default object of a primitive class.

5.13.1 Detailed Description

This class gives some static methods to look inside a class.

5.13.2 Member Function Documentation

5.13.2.1 static Constructor<?> popjava.util.ClassUtil.getConstructor (Class<?> *c*, Class<?>... *parameterTypes*) throws NoSuchMethodException [static]

Retrieve a specific constructor in the given class.

Parameters

c	The class to look in
parameterTypes	Parameters of the constructor to retrieve

Returns

The retrieved constructor

Exceptions

NoSuchMethodException	Thrown if the constructor is not found

Here is the call graph for this function:

Here is the caller graph for this function:

5.13.2.2 static Object popjava.util.ClassUtil.getDefaultPrimitiveValue (Class< ? > *c*) [static]

Get a default object of a primitive class.

Parameters

c	The primitive class

Returns

Object with default value

Here is the caller graph for this function:

5.13.2.3 static Method popjava.util.ClassUtil.getMethod (Class< ? > *c,* String *methodName,* Class< ? >... *parameterTypes*) throws NoSuchMethodException [static]

Retrieve a specific method in the given class.

Parameters

c	The class to look in
methodName	The name of the method to retrieve
parameterTypes	Parameters of the method to retrieve

Returns

The retrieved method

Exceptions

NoSuchMethodException	Thrown if the method is not found

Here is the call graph for this function:

5.13.2.4 static String popjava.util.ClassUtil.getMethodSign (Method *m*) [static]

Get the signature of a method.

Parameters

m	The method

Returns

Signature of the given method as a string value

Here is the caller graph for this function:

5.13.2.5 static String popjava.util.ClassUtil.getMethodSign (Constructor< ? > *c*) [static]

Get the signature of a constructor.

Parameters

c	The constructor

Returns

Signature of the constructor as a string value

Here is the call graph for this function:

5.14 popjava.combox.Combox Class Reference

This class is the base implementation for all Combox in the POP-Java library All other combox must inherit from this class.

Inheritance diagram for popjava.combox.Combox:

Collaboration diagram for popjava.combox.Combox:

Public Member Functions

- Combox ()

  Default constructor.
- Combox (POPAccessPoint accesspoint, int timeout)

  Constructor with given values.
- boolean connect (POPAccessPoint accesspoint, int timeout)

  Connect the current combox to the other side combox.
- abstract int send (POPBuffer buffer)

  Send the buffer to the other side.
- abstract int receive (POPBuffer buffer)

  Receive buffer from the other side.
- abstract void close ()

  Close the connection.
- abstract boolean connect ()

  Connect to the other side.
- void setBufferFactory (BufferFactory bufferFactory)

  Associate a buffer factory to the combox.
- BufferFactory getBufferFactory ()

  Get the associated buffer factory.

Protected Attributes

- int **timeOut** = 0
- POPAccessPoint **accessPoint**
- boolean **available** = false
- BufferFactory **bufferFactory**

5.14.1 Detailed Description

This class is the base implementation for all Combox in the POP-Java library All other combox must inherit from this class.

5.14.2 Constructor & Destructor Documentation

5.14.2.1 popjava.combox.Combox.Combox (**POPAccessPoint** *accesspoint,* int *timeout*)

Constructor with given values.

Parameters

accesspoint	Access point to create the combox
timeout	Connection time out

5.14.3 Member Function Documentation

5.14.3.1 boolean popjava.combox.Combox.connect (**POPAccessPoint** *accesspoint,* int *timeout*)

Connect the current combox to the other side combox.

Parameters

accesspoint	Access point of the other side combox
timeout	Connection time out

Returns

true if the connection is established

Here is the call graph for this function:

5.14.3.2 abstract boolean popjava.combox.Combox.connect ( ) `[pure virtual]`

Connect to the other side.

Returns

true if the connection succeed

Implemented in popjava.combox.ComboxSocket, and popjava.combox.ComboxPlugin.

Here is the caller graph for this function:

5.14.3.3 **BufferFactory** popjava.combox.Combox.getBufferFactory ( )

Get the associated buffer factory.

Returns

The associated buffer factory

Here is the caller graph for this function:

5.14.3.4 abstract int popjava.combox.Combox.receive (**POPBuffer** *buffer*) `[pure virtual]`

Receive buffer from the other side.

Parameters

buffer	Buffer to receive

Returns

Number of byte received

Implemented in popjava.combox.ComboxSocket, and popjava.combox.ComboxPlugin.

5.14.3.5 abstract int popjava.combox.Combox.send (**POPBuffer** *buffer*) `[pure virtual]`

Send the buffer to the other side.

Parameters

buffer	The buffer to send

Returns

Number of byte sent

Implemented in popjava.combox.ComboxSocket, and popjava.combox.ComboxPlugin.

5.14.3.6 void popjava.combox.Combox.setBufferFactory (**BufferFactory** *bufferFactory*)

Associate a buffer factory to the combox.

Parameters

bufferFactory	The buffer factory to associate

Here is the caller graph for this function:

5.15 popjava.combox.ComboxAcceptSocket Class Reference

This class is responsible to accept the new connection for the associated server combox socket.

Inheritance diagram for popjava.combox.ComboxAcceptSocket:

Collaboration diagram for popjava.combox.ComboxAcceptSocket:

Public Member Functions

- ComboxAcceptSocket (Broker broker, RequestQueue requestQueue, ServerSocket socket)

  Create a new instance of the ComboxAccept socket.
- void run ()

  Start the local thread.
- void close ()

  Close the current connection.
- synchronized int getStatus ()

  Get the current status.
- synchronized void setStatus (int status)

  Set the current status.

Static Public Attributes

- static final int **Running** = 0

- static final int **Exit** = 1
- static final int **Abort** = 2

Protected Attributes

- Broker **broker**
- RequestQueue **requestQueue**
- ServerSocket **serverSocket**
- int **status** = Exit
- LinkedList< Socket > **concurentConnections** = new LinkedList<Socket>()

5.15.1 Detailed Description

This class is responsible to accept the new connection for the associated server combox socket.

5.15.2 Constructor & Destructor Documentation

5.15.2.1 popjava.combox.ComboxAcceptSocket.ComboxAcceptSocket (Broker *broker*, RequestQueue *requestQueue*, ServerSocket *socket*)

Create a new instance of the ComboxAccept socket.

Parameters

broker	The associated broker
requestQueue	The associated request queue
socket	The associated combox socket

5.15.3 Member Function Documentation

5.15.3.1 synchronized int popjava.combox.ComboxAcceptSocket.getStatus ( )

Get the current status.

Returns

The current status

5.15.3.2 synchronized void popjava.combox.ComboxAcceptSocket.setStatus (int *status*)

Set the current status.

Parameters

status	The new status

Here is the caller graph for this function:

5.16 popjava.combox.ComboxAllocateSocket Class Reference

This class is responsible to send an receive message on the server combox socket.

Collaboration diagram for popjava.combox.ComboxAllocateSocket:

Public Member Functions

- ComboxAllocateSocket ()

  Create a new instance of the ComboxAllocateSocket.
- void startToAcceptOneConnection ()

  Start the socket and wait for a connection.
- String getUrl ()

  Get URL of this socket.
- void close ()

  Close the current connection.
- int send (POPBuffer buffer)

  Send a message to the other-side.
- int receive (POPBuffer buffer)

  Receive a new message from the other-side.
- boolean **isComboxConnected** ()

Protected Attributes

- ServerSocket **serverSocket** = null

5.16.1 Detailed Description

This class is responsible to send an receive message on the server combox socket.

5.16.2 Member Function Documentation

5.16.2.1 String popjava.combox.ComboxAllocateSocket.getUrl ( )

Get URL of this socket.

Returns

The URL as a string value

Here is the call graph for this function:

5.16.2.2 int popjava.combox.ComboxAllocateSocket.receive (POPBuffer *buffer*)

Receive a new message from the other-side.

Parameters

buffer	Buffer to receive the message

Returns

Number of byte read

Here is the call graph for this function:

5.16.2.3 int popjava.combox.ComboxAllocateSocket.send (POPBuffer *buffer*)

Send a message to the other-side.

Parameters

buffer	Buffer to be send

Returns

Number of byte sent

Here is the call graph for this function:

5.17 popjava.combox.ComboxFactory Class Reference

This abstract class regroup the method needed by a ComboxFactory.

Inheritance diagram for popjava.combox.ComboxFactory:

Collaboration diagram for popjava.combox.ComboxFactory:

Public Member Functions

- abstract Combox createClientCombox (POPAccessPoint accessPoint)

  Create a new client combox with the given access point.
- abstract Combox createClientCombox (POPAccessPoint accessPoint, int timeout)

  Create a new client combox with the given access point and a specified timeout.
- abstract ComboxServer createServerCombox (AccessPoint accessPoint, POPBuffer buffer, Broker broker)

  Create a new server combox with the given access point, buffer and broker.
- abstract ComboxServer createServerCombox (AccessPoint accessPoint, int timeout, POPBuffer buffer, Broker broker)

  Create a new server combox with the given access point, buffer and broker and a connection timeout.
- abstract String getComboxName ()

  Get the combox name.

5.17.1 Detailed Description

This abstract class regroup the method needed by a ComboxFactory.

5.17.2 Member Function Documentation

5.17.2.1 abstract **Combox** popjava.combox.ComboxFactory.createClientCombox (**POPAccessPoint** ***accessPoint***) `[pure virtual]`

Create a new client combox with the given access point.

Parameters

accessPoint	The access point to connect the combox

Returns

The combox created

Implemented in popjava.combox.ComboxSocketFactory, and popjava.combox.ComboxFactoryPlugin.

5.17.2.2 abstract **Combox** popjava.combox.ComboxFactory.createClientCombox (**POPAccessPoint** *accessPoint,* int *timeout*) `[pure virtual]`

Create a new client combox with the given access point and a specified timeout.

Parameters

accessPoint	The access point to connect the combox
timeout	The connection timeout

Returns

The combox created

Implemented in popjava.combox.ComboxSocketFactory, and popjava.combox.ComboxFactoryPlugin.

5.17.2.3 abstract **ComboxServer** popjava.combox.ComboxFactory.createServerCombox (**AccessPoint** *accessPoint,* **POPBuffer** *buffer,* **Broker** *broker*) `[pure virtual]`

Create a new server combox with the given access point, buffer and broker.

Parameters

accessPoint	The access point for the server
buffer	The buffer for sending and receiving
broker	The broker associated with this combox

Returns

The combox server created

Implemented in popjava.combox.ComboxSocketFactory, and popjava.combox.ComboxFactoryPlugin.

5.17.2.4 abstract **ComboxServer** popjava.combox.ComboxFactory.createServerCombox (**AccessPoint** *accessPoint,* int *timeout,* **POPBuffer** *buffer,* **Broker** *broker*) `[pure virtual]`

Create a new server combox with the given access point, buffer and broker and a connection timeout.

Parameters

accessPoint	The access point for the server
timeout	The connection timeout
buffer	The buffer for sending and receiving
broker	The broker associated with this combox

Returns

The combox server created

Implemented in popjava.combox.ComboxSocketFactory, and popjava.combox.ComboxFactoryPlugin.

5.17.2.5 abstract String popjava.combox.ComboxFactory.getComboxName ( ) `[pure virtual]`

Get the combox name.

Returns

name of the combox

Implemented in popjava.combox.ComboxFactoryPlugin, and popjava.combox.ComboxSocketFactory.

5.18 popjava.combox.ComboxFactoryFinder Class Reference

This class is responsible to find the different combox available in POP-Java.

Collaboration diagram for popjava.combox.ComboxFactoryFinder:

Public Member Functions

- void loadComboxMap (String pluginLocation)

  Load all the combox in the pop_combox.xml file.
- ComboxFactory findFactory (String factoryName)

  Find a specific factory with the given name.
- int getFactoryCount ()

  Get the number of factory.
- ComboxFactory get (int index)

  Get the factory at the specified index.

Static Public Member Functions

- static ComboxFactoryFinder getInstance ()

  Get the unique instance of the factory finder.

Protected Member Functions

- ComboxFactoryFinder ()

  Default constructor.

5.18.1 Detailed Description

This class is responsible to find the different combox available in POP-Java.

5.18.2 Member Function Documentation

5.18.2.1 ComboxFactory popjava.combox.ComboxFactoryFinder.findFactory (String *factoryName*)

Find a specific factory with the given name.

Parameters

factoryName	Name of the factory

Returns

The combox factory or null if not found

5.18.2.2 ComboxFactory popjava.combox.ComboxFactoryFinder.get (int *index*)

Get the factory at the specified index.

Parameters

index	Index of the factory

Returns

The factory at the specified index or null if out of bound index

Here is the call graph for this function:

5.18.2.3 int popjava.combox.ComboxFactoryFinder.getFactoryCount ( )

Get the number of factory.

Returns

Number of factory

Here is the caller graph for this function:

5.18.2.4 static ComboxFactoryFinder popjava.combox.ComboxFactoryFinder.getInstance ( ) [static]

Get the unique instance of the factory finder.

Returns

The unique ComboxFactoryFinder instance

Here is the call graph for this function:

5.18.2.5 void popjava.combox.ComboxFactoryFinder.loadComboxMap (String *pluginLocation*)

Load all the combox in the pop_combox.xml file.

Parameters

pluginLocation	Location of the plugin file

Here is the call graph for this function:

Here is the caller graph for this function:

5.19 popjava.combox.ComboxFactoryPlugin Class Reference

This class defined the interface for new combox factory plug-in.

Inheritance diagram for popjava.combox.ComboxFactoryPlugin:

Collaboration diagram for popjava.combox.ComboxFactoryPlugin:

Public Member Functions

- Combox createClientCombox (POPAccessPoint accessPoint)

  Create a new client combox with the given access point.
- Combox createClientCombox (POPAccessPoint accessPoint, int timeout)

  Create a new client combox with the given access point and a specified timeout.
- ComboxServer createServerCombox (AccessPoint accessPoint, POPBuffer buffer, Broker broker)

  Create a new server combox with the given access point, buffer and broker.
- ComboxServer createServerCombox (AccessPoint accessPoint, int timeout, POPBuffer buffer, Broker broker)

  Create a new server combox with the given access point, buffer and broker and a connection timeout.
- String getComboxName ()

  Get the combox name.

5.19.1 Detailed Description

This class defined the interface for new combox factory plug-in.

5.19.2 Member Function Documentation

5.19.2.1 Combox popjava.combox.ComboxFactoryPlugin.createClientCombox (POPAccessPoint *accessPoint*) `[virtual]`

Create a new client combox with the given access point.

Parameters

accessPoint	The access point to connect the combox

Returns

The combox created

Implements popjava.combox.ComboxFactory.

5.19.2.2 Combox popjava.combox.ComboxFactoryPlugin.createClientCombox (POPAccessPoint *accessPoint,* int *timeout*) `[virtual]`

Create a new client combox with the given access point and a specified timeout.

Parameters

accessPoint	The access point to connect the combox
timeout	The connection timeout

Returns

The combox created

Implements popjava.combox.ComboxFactory.

5.19.2.3 ComboxServer popjava.combox.ComboxFactoryPlugin.createServerCombox (AccessPoint *accessPoint*, POPBuffer *buffer*, Broker *broker*) [virtual]

Create a new server combox with the given access point, buffer and broker.

Parameters

accessPoint	The access point for the server
buffer	The buffer for sending and receiving
broker	The broker associated with this combox

Returns

The combox server created

Implements popjava.combox.ComboxFactory.

5.19.2.4 ComboxServer popjava.combox.ComboxFactoryPlugin.createServerCombox (AccessPoint *accessPoint*, int *timeout*, POPBuffer *buffer*, Broker *broker*) [virtual]

Create a new server combox with the given access point, buffer and broker and a connection timeout.

Parameters

accessPoint	The access point for the server
timeout	The connection timeout
buffer	The buffer for sending and receiving
broker	The broker associated with this combox

Returns

The combox server created

Implements popjava.combox.ComboxFactory.

5.19.2.5 String popjava.combox.ComboxFactoryPlugin.getComboxName ( ) [virtual]

Get the combox name.

Returns

name of the combox

Implements popjava.combox.ComboxFactory.

5.20 popjava.combox.ComboxPlugin Class Reference

This class defined the interface for each new combox plug-in.

Inheritance diagram for popjava.combox.ComboxPlugin:

Collaboration diagram for popjava.combox.ComboxPlugin:

Public Member Functions

- void close ()

  Close the connection.
- boolean connect ()

  Connect to the other side.
- int receive (POPBuffer buffer)

  Receive buffer from the other side.
- int send (POPBuffer buffer)

  Send the buffer to the other side.

Additional Inherited Members

5.20.1 Detailed Description

This class defined the interface for each new combox plug-in.

5.20.2 Member Function Documentation

5.20.2.1 boolean popjava.combox.ComboxPlugin.connect ( ) `[virtual]`

Connect to the other side.

Returns

true if the connection succeed

Implements popjava.combox.Combox.

5.20.2.2 int popjava.combox.ComboxPlugin.receive (**POPBuffer** *buffer*) `[virtual]`

Receive buffer from the other side.

Parameters

buffer	Buffer to receive

Returns

Number of byte received

Implements popjava.combox.Combox.

5.20.2.3 int popjava.combox.ComboxPlugin.send (**POPBuffer** *buffer*) `[virtual]`

Send the buffer to the other side.

Parameters

buffer	The buffer to send

Returns

Number of byte sent

Implements popjava.combox.Combox.

5.21 popjava.combox.ComboxReceiveRequestSocket Class Reference

This class is responsible to receive the new request for the associated combox.

Inheritance diagram for popjava.combox.ComboxReceiveRequestSocket:

Collaboration diagram for popjava.combox.ComboxReceiveRequestSocket:

Public Member Functions

- ComboxReceiveRequestSocket (Broker broker, RequestQueue requestQueue, Socket socket) throws IO-Exception

  Crate a new instance of ComboxReceiveRequestSocket.
- void run ()

  Start the thread.
- boolean receiveRequest (Request request)

  Get request from the buffer.
- void close ()

  Close the current connection.
- synchronized int getStatus ()

  Get the status of the current connection.
- synchronized void setStatus (int status)

  Set the current status.
- void setBuffer (String bufferType)

  Associate a buffer with this receiving combox.

Static Public Attributes

- static final int **Running** = 0
- static final int **Exit** = 1
- static final int **Abort** = 2

Protected Member Functions

- void finalize () throws Throwable

  Method called before destruction of the instance.

Protected Attributes

- ComboxSocket **combox**
- RequestQueue **requestQueue**
- Broker **broker**
- int **status** = Exit

5.21.1 Detailed Description

This class is responsible to receive the new request for the associated combox.

5.21.2 Constructor & Destructor Documentation

5.21.2.1 popjava.combox.ComboxReceiveRequestSocket.ComboxReceiveRequestSocket (Broker *broker*, RequestQueue *requestQueue*, Socket *socket*) throws IOException

Crate a new instance of ComboxReceiveRequestSocket.

Parameters

broker	The associated broker
requestQueue	The associated request queue
socket	The associated socket

Exceptions

IOException	Thrown if any exception occurred during the process

5.21.3 Member Function Documentation

5.21.3.1 synchronized int popjava.combox.ComboxReceiveRequestSocket.getStatus ( )

Get the status of the current connection.

Returns

Current connection status

Here is the caller graph for this function:

5.21.3.2 boolean popjava.combox.ComboxReceiveRequestSocket.receiveRequest (Request *request*)

Get request from the buffer.

Parameters

request	The request

Returns

true if the new request if complete or false if it's incomplete

Here is the call graph for this function:

Here is the caller graph for this function:

5.21.3.3 void popjava.combox.ComboxReceiveRequestSocket.setBuffer (String *bufferType*)

Associate a buffer with this receiving combox.

Parameters

bufferType	Type of the buffer

Here is the call graph for this function:

5.21.3.4 synchronized void popjava.combox.ComboxReceiveRequestSocket.setStatus (int *status*)

Set the current status.

Parameters

status	The new status

Here is the caller graph for this function:

5.22 popjava.combox.ComboxServer Class Reference

This class represent the server side of a socket connection.

Inheritance diagram for popjava.combox.ComboxServer:

Collaboration diagram for popjava.combox.ComboxServer:

Public Member Functions

- ComboxServer (AccessPoint accessPoint, int timeout, Broker broker)

  Default constructor.
- RequestQueue getRequestQueue ()

  Get the associated request queue.

Static Public Attributes

- static final int **Running** = 0
- static final int **Exit** = 1
- static final int **Abort** = 2

Protected Attributes

- int **status** = Exit
- RequestQueue **requestQueue** = new RequestQueue()
- Broker **broker**
- int **timeOut** = 0
- AccessPoint **accessPoint**

5.22.1 Detailed Description

This class represent the server side of a socket connection.

5.22.2 Constructor & Destructor Documentation

5.22.2.1 popjava.combox.ComboxServer.ComboxServer (AccessPoint *accessPoint*, int *timeout*, Broker *broker*)

Default constructor.

Parameters

accessPoint	Access point of the combox server
timeout	Connection timeout
broker	Associated broker

5.22.3 Member Function Documentation

5.22.3.1 RequestQueue popjava.combox.ComboxServer.getRequestQueue ( )

Get the associated request queue.

Returns

The associated request queue

5.23 popjava.combox.ComboxServerPlugin Class Reference

This class defined the interface for all new combox server plug-in.

Inheritance diagram for popjava.combox.ComboxServerPlugin:

Collaboration diagram for popjava.combox.ComboxServerPlugin:

Public Member Functions

- ComboxServerPlugin (AccessPoint accessPoint, int timeout, Broker broker)

  Default constructor.
- RequestQueue getRequestQueue ()

  Get the associated request queue.

Additional Inherited Members

5.23.1 Detailed Description

This class defined the interface for all new combox server plug-in.

5.23.2 Constructor & Destructor Documentation

5.23.2.1 popjava.combox.ComboxServerPlugin.ComboxServerPlugin (AccessPoint *accessPoint*, int *timeout*, Broker *broker*)

Default constructor.

Create a new combox server plug-in

Parameters

accessPoint	Access point of the combox server
timeout	Connection timeout
broker	Associated broker

5.23.3 Member Function Documentation

5.23.3.1 RequestQueue popjava.combox.ComboxServerPlugin.getRequestQueue ( )

Get the associated request queue.

Returns

The associated request queue

5.24 popjava.combox.ComboxServerSocket Class Reference

This class is an implementation of the combox with the protocol socket for the server side.

Inheritance diagram for popjava.combox.ComboxServerSocket:

Collaboration diagram for popjava.combox.ComboxServerSocket:

Public Member Functions

- ComboxServerSocket (AccessPoint accessPoint, int timeout, POPBuffer buffer, Broker broker)
  Default constructor.
- String GetUrl ()
  Get the URL of the combox.
- void createServer ()
  Create and start the combox server.

Static Public Attributes

- static int **BufferLength** = 1024

Protected Attributes

- ServerSocket **serverSocket** = null

5.24.1 Detailed Description

This class is an implementation of the combox with the protocol socket for the server side.

5.24.2 Constructor & Destructor Documentation

5.24.2.1 popjava.combox.ComboxServerSocket.ComboxServerSocket (**AccessPoint** *accessPoint,* int *timeout,* **POPBuffer** *buffer,* **Broker** *broker*)

Default constructor.

Create a new instance of a socket combox

Parameters

accessPoint	Access point of the combox
timeout	Connection timeout
buffer	Buffer associated with this combox
broker	Broker associated with this combox

Here is the call graph for this function:

5.24.3 Member Function Documentation

5.24.3.1 String popjava.combox.ComboxServerSocket.GetUrl ( )

Get the URL of the combox.

Returns

URL as a string value

Here is the call graph for this function:

5.25 popjava.combox.ComboxSocket Class Reference

This combox implement the protocol Socket.

Inheritance diagram for popjava.combox.ComboxSocket:

Collaboration diagram for popjava.combox.ComboxSocket:

Public Member Functions

- ComboxSocket (Socket socket) throws IOException

  Create a new combox on the given socket.
- **ComboxSocket** (POPAccessPoint accesspoint, int timeout)
- void close ()

  Close the connection.
- boolean connect ()

  Connect to the other side.
- int receive (POPBuffer buffer)

  Receive buffer from the other side.
- int send (POPBuffer buffer)

  Send the buffer to the other side.

Static Public Attributes

- static int **BufferLength** = 1024 ∗ 1024 ∗ 8

Protected Member Functions

- void **finalize** () throws Throwable

Protected Attributes

- Socket **peerConnection** = null
- byte[ ] **receivedBuffer**
- InputStream **inputStream** = null
- OutputStream **outputStream** = null

5.25.1 Detailed Description

This combox implement the protocol Socket.

5.25.2 Constructor & Destructor Documentation

5.25.2.1 popjava.combox.ComboxSocket.ComboxSocket (Socket *socket*) throws IOException

Create a new combox on the given socket.

Parameters

socket	The socket to create the combox

Exceptions

IOException	Thrown is any IO exception occurred during the creation

5.25.3 Member Function Documentation

5.25.3.1 boolean popjava.combox.ComboxSocket.connect ( ) `[virtual]`

Connect to the other side.

Returns

true if the connection succeed

Implements popjava.combox.Combox.

Here is the call graph for this function:

5.25.3.2 int popjava.combox.ComboxSocket.receive (POPBuffer *buffer*) `[virtual]`

Receive buffer from the other side.

Parameters

buffer	Buffer to receive

Returns

Number of byte received

Implements popjava.combox.Combox.

Here is the call graph for this function:

Here is the caller graph for this function:

5.25.3.3 int popjava.combox.ComboxSocket.send (POPBuffer *buffer*) `[virtual]`

Send the buffer to the other side.

Parameters

buffer	The buffer to send

Returns

Number of byte sent

Implements popjava.combox.Combox.

Here is the call graph for this function:

Here is the caller graph for this function:

5.26 popjava.combox.ComboxSocketFactory Class Reference

This class is the factory for all combox socket.

Inheritance diagram for popjava.combox.ComboxSocketFactory:

Collaboration diagram for popjava.combox.ComboxSocketFactory:

Public Member Functions

- String getComboxName ()

  Get the combox name.
- Combox createClientCombox (POPAccessPoint accessPoint)

  Create a new client combox with the given access point.
- Combox createClientCombox (POPAccessPoint accessPoint, int timeout)

  Create a new client combox with the given access point and a specified timeout.
- ComboxServer createServerCombox (AccessPoint accessPoint, POPBuffer buffer, Broker broker)

  Create a new server combox with the given access point, buffer and broker.
- ComboxServer createServerCombox (AccessPoint accessPoint, int timeout, POPBuffer buffer, Broker broker)

  Create a new server combox with the given access point, buffer and broker and a connection timeout.

Static Public Attributes

- static final String Protocol = "socket"

  Name of the implemented protocol.

5.26.1 Detailed Description

This class is the factory for all combox socket.

5.26.2 Member Function Documentation

5.26.2.1 Combox popjava.combox.ComboxSocketFactory.createClientCombox (POPAccessPoint *accessPoint*) [virtual]

Create a new client combox with the given access point.

Parameters

accessPoint	The access point to connect the combox

Returns

The combox created

Implements popjava.combox.ComboxFactory.

5.26.2.2 Combox popjava.combox.ComboxSocketFactory.createClientCombox (POPAccessPoint *accessPoint*, int *timeout*) [virtual]

Create a new client combox with the given access point and a specified timeout.

Parameters

accessPoint	The access point to connect the combox
timeout	The connection timeout

Returns

The combox created

Implements popjava.combox.ComboxFactory.

5.26.2.3 ComboxServer popjava.combox.ComboxSocketFactory.createServerCombox (AccessPoint *accessPoint*, POPBuffer *buffer*, Broker *broker*) [virtual]

Create a new server combox with the given access point, buffer and broker.

Parameters

accessPoint	The access point for the server
buffer	The buffer for sending and receiving
broker	The broker associated with this combox

Returns

The combox server created

Implements popjava.combox.ComboxFactory.

5.26.2.4 ComboxServer popjava.combox.ComboxSocketFactory.createServerCombox (AccessPoint *accessPoint*, int *timeout*, POPBuffer *buffer*, Broker *broker*) [virtual]

Create a new server combox with the given access point, buffer and broker and a connection timeout.

Parameters

accessPoint	The access point for the server
timeout	The connection timeout
buffer	The buffer for sending and receiving
broker	The broker associated with this combox

Returns

The combox server created

Implements popjava.combox.ComboxFactory.

5.26.2.5 String popjava.combox.ComboxSocketFactory.getComboxName ( ) [virtual]

Get the combox name.

Returns

name of the combox

Implements popjava.combox.ComboxFactory.

Here is the caller graph for this function:

5.27 popjava.util.Configuration Class Reference

This class regroup some configuration values.

Collaboration diagram for popjava.util.Configuration:

Public Member Functions

- Configuration ()

  Default constructor.

Static Public Attributes

- static final boolean Debug = false

  Creates a new instance of POPConfiguration.
- static final boolean **DebugCombox** = false
- static final int **RESERVE_TIMEOUT** = 60000
- static final int **ALLOC_TIMEOUT** = 30000
- static final int **CONNECTION_TIMEOUT** = 30000
- static final String **DefaultEncoding** = "xdr"
- static final String **SelectedEncoding** = "raw"
- static final String **DefaultProtocol** = "socket"
- static final boolean **ACTIVATE_JMX** = false
- static final boolean **CONNECT_TO_POPCPP** = false
- static final boolean **REDIRECT_OUTPUT_TO_ROOT** = true
- static final boolean **USE_NATIVE_SSH_IF_POSSIBLE** = true

5.27.1 Detailed Description

This class regroup some configuration values.

5.28 popjava.system.ConfigurationWorker Class Reference

POP-Java configuration class.

Inheritance diagram for popjava.system.ConfigurationWorker:

Collaboration diagram for popjava.system.ConfigurationWorker:

Public Member Functions

- ConfigurationWorker () throws Exception

    Constructs a new ConfigurationWorker and retrieve POP-Java base location.
- String getValue (String name)

    Retrieve a configuration item in the configuration file by its name.

Static Public Attributes

- static final String POPJ_LOCATION_ITEM = "popj_location"

    POP-Java location configuration item name.
- static final String POPJ_PLUGIN_ITEM = "popj_plugin_location"

    POP-Java plug-in location configuration item name.
- static final String POPJ_BROKER_COMMAND_ITEM = "popj_broker_command"

    POP-Java broker command configuration item name.
- static final String POPC_APPCORESERVICE_ITEM = "popc_appcoreservice_location"

    POP-Java application core service location configuration name.

Additional Inherited Members

5.28.1 Detailed Description

POP-Java configuration class.

Provide access trough the different configuration parameters stored in the XML configuration file.

Author

clementval

5.28.2 Constructor & Destructor Documentation

5.28.2.1 popjava.system.ConfigurationWorker.ConfigurationWorker ( ) throws Exception

Constructs a new ConfigurationWorker and retrieve POP-Java base location.

Exceptions

Exception	thrown if the configuration file is not valid with its XML schema

Here is the call graph for this function:

5.28.3 Member Function Documentation

5.28.3.1 String popjava.system.ConfigurationWorker.getValue (String *name*)

Retrieve a configuration item in the configuration file by its name.

Parameters

name	name of the item to retrieve the value

Returns

Value of the item or null if not found

5.29 popjava.annotation.POPParameter.Direction Enum Reference

Collaboration diagram for popjava.annotation.POPParameter.Direction:

Public Attributes

- **IN**
- **OUT**
- **INOUT**

5.30 popjava.interfacebase.Interface Class Reference

Interface side of a POP-Java parallel object.

Inheritance diagram for popjava.interfacebase.Interface:

Collaboration diagram for popjava.interfacebase.Interface:

Public Member Functions

- Interface ()

    Default Interface constructor.
- Interface (POPAccessPoint accessPoint) throws POPException

    Create an Interface by giving the access point of the parallel object.
- boolean serialize (POPBuffer buffer)

    Serialization of the Interface into the buffer.
- boolean deserialize (POPBuffer buffer)

    Deserialize an Interface from a buffer.
- POPAccessPoint getAccessPoint ()

    Return the access point of the parallel object associated with this interface.
- void setAccessPoint (POPAccessPoint accessPoint)

    Set the access point associated with this interface.
- ObjectDescription getOD ()

    Return the object description associated with this interface.
- void setOd (ObjectDescription od)

    Associate an object description with this interface.
- void **release** ()
- boolean allocate (String objectName) throws POPException

    Allocate resource for the associated parallel object.
- int **addRef** ()
- int **decRef** ()
- boolean isAlive ()

    Check if the parallel object associated with this interface is still alive.
- void kill ()

    Kill the associated parallel object.
- void close ()

    Close the combox associated with this interface.

Protected Member Functions

- boolean bind (POPAccessPoint accesspoint) throws POPException

    Bind the interface with a parallel object (Broker-side)
- void popDispatch (POPBuffer buffer)

    Send the buffer content to the broker-side.
- int popResponse (POPBuffer buffer) throws POPException

    Receive response from the broker-side.
- void finalize () throws Throwable

    Close everything.

Protected Attributes

- Combox **combox**
- POPAccessPoint **popAccessPoint** = new POPAccessPoint()
- ObjectDescription **od** = new ObjectDescription()

5.30.1 Detailed Description

Interface side of a POP-Java parallel object.

This object is the local representative of the parallel object

5.30.2 Constructor & Destructor Documentation

5.30.2.1 popjava.interfacebase.Interface.Interface (POPAccessPoint *accessPoint*) throws POPException

Create an Interface by giving the access point of the parallel object.

Parameters

accessPoint	Access point of the parallel object

Exceptions

POPException	thrown of the interface cannot be bind with the parallel object

Here is the call graph for this function:

5.30.3 Member Function Documentation

5.30.3.1 boolean popjava.interfacebase.Interface.allocate (String *objectName*) throws POPException

Allocate resource for the associated parallel object.

Parameters

objectName	Name of the object

Returns

True if the interface can allocate some resources

Exceptions

POPException	thrown if any exception occurred during the allocating process

Here is the call graph for this function:

Here is the caller graph for this function:

5.30.3.2 boolean popjava.interfacebase.Interface.bind (**POPAccessPoint** *accesspoint*) throws **POPException** [protected]

Bind the interface with a parallel object (Broker-side)

Parameters

accesspoint	Access point of the parallel object (Broker-side)s

Returns

true if the interface is binded to the broker-side

Exceptions

POPException	thrown if any exception occurred during the binding process

Here is the call graph for this function:

Here is the caller graph for this function:

5.30.3.3 boolean popjava.interfacebase.Interface.deserialize (**POPBuffer** *buffer*)

Deserialize an Interface from a buffer.

Parameters

buffer	Buffer to deserialize from

Returns

True if the deserialization has finished without any problems

Here is the call graph for this function:

5.30.3.4 **POPAccessPoint** popjava.interfacebase.Interface.getAccessPoint ( )

Return the access point of the parallel object associated with this interface.

Returns

Access point of the associated parallel object

5.30.3.5 **ObjectDescription** popjava.interfacebase.Interface.getOD ( )

Return the object description associated with this interface.

Returns

ObjectDescription of this interface

Here is the caller graph for this function:

5.30.3.6 boolean popjava.interfacebase.Interface.isAlive ( )

Check if the parallel object associated with this interface is still alive.

Returns

true if the parallel object is alive

Here is the call graph for this function:

5.30.3.7 void popjava.interfacebase.Interface.popDispatch (POPBuffer *buffer*) [protected]

Send the buffer content to the broker-side.

Parameters

buffer	Buffer to send

Here is the caller graph for this function:

5.30.3.8 int popjava.interfacebase.Interface.popResponse (POPBuffer *buffer*) throws POPException [protected]

Receive response from the broker-side.

Parameters

buffer	

Returns

Exceptions

POPException	

Here is the call graph for this function:

Here is the caller graph for this function:

5.30.3.9 boolean popjava.interfacebase.Interface.serialize (POPBuffer *buffer*)

Serialization of the Interface into the buffer.

Parameters

buffer	Buffer to serialize in

Returns

true if the serialization is finished without any problems

Here is the call graph for this function:

5.30.3.10 void popjava.interfacebase.Interface.setAccessPoint (POPAccessPoint *accessPoint*)

Set the access point associated with this interface.

Parameters

accessPoint	Access point to associate

5.30.3.11 void popjava.interfacebase.Interface.setOd (ObjectDescription *od*)

Associate an object description with this interface.

Parameters

od	Object descritption to associate

Here is the caller graph for this function:

5.31 popjava.dataswaper.IPOPBase Interface Reference

This interface declare the needed method for the serialization and the deserialization of an object.

Inheritance diagram for popjava.dataswaper.IPOPBase:

Collaboration diagram for popjava.dataswaper.IPOPBase:

Public Member Functions

- boolean serialize (POPBuffer buffer)

  Serialize an object into the buffer.
- boolean deserialize (POPBuffer buffer)

  Deserialize an object from the buffer.

5.31.1 Detailed Description

This interface declare the needed method for the serialization and the deserialization of an object.

5.31.2 Member Function Documentation

5.31.2.1 boolean popjava.dataswaper.IPOPBase.deserialize (POPBuffer *buffer*)

Deserialize an object from the buffer.

Parameters

buffer	The buffer to deserialize from

Returns

true if the deserialization process succeed

Implemented in popjava.base.POPObject, popjava.baseobject.ObjectDescription, popjava.base.POPException, popjava.baseobject.POPAccessPoint, and popjava.dataswaper.POPString.

Here is the caller graph for this function:

5.31.2.2 boolean popjava.dataswaper.IPOPBase.serialize (POPBuffer *buffer*)

Serialize an object into the buffer.

Parameters

buffer	The buffer to serialize in

Returns

true if the serialization process succeed

Implemented in popjava.base.POPObject, popjava.baseobject.ObjectDescription, popjava.base.POPException, popjava.baseobject.POPAccessPoint, and popjava.dataswaper.POPString.

Here is the caller graph for this function:

5.32 popjava.dataswaper.IPOPBaseConst Interface Reference

This type is used for communicate with the pop-c++ only.

Collaboration diagram for popjava.dataswaper.IPOPBaseConst:

Public Member Functions

- boolean **serialize** (POPBuffer buffer)

5.32.1 Detailed Description

This type is used for communicate with the pop-c++ only.

It is compatible with the in type Be careful when use this type

5.33 popjava.dataswaper.IPOPBaseInput Interface Reference

This type is used for communicate with the pop-c++ only.

Inheritance diagram for popjava.dataswaper.IPOPBaseInput:

Collaboration diagram for popjava.dataswaper.IPOPBaseInput:

Public Member Functions

- boolean serialize (POPBuffer buffer)

    Serialize an object into the buffer.
- boolean deserialize (POPBuffer buffer)

    Deserialize an object from the buffer.

5.33.1 Detailed Description

This type is used for communicate with the pop-c++ only.

It is compatible with the in type Be careful when use this type

5.33.2 Member Function Documentation

5.33.2.1 boolean popjava.dataswaper.IPOPBaseInput.deserialize (**POPBuffer** *buffer*)

Deserialize an object from the buffer.

Parameters

buffer	The buffer to deserialize from

Returns

true if the deserialization process succeed

Implemented in popjava.dataswaper.ObjectDescriptionInput.

Here is the caller graph for this function:

5.33.2.2 boolean popjava.dataswaper.IPOPBaseInput.serialize (**POPBuffer** *buffer*)

Serialize an object into the buffer.

Parameters

buffer	The buffer to serialize in

Returns

true if the serialization process succeed

Implemented in popjava.dataswaper.ObjectDescriptionInput.

Here is the caller graph for this function:

5.34 popjava.util.LogWriter Class Reference

This class is used to write log file.

Collaboration diagram for popjava.util.LogWriter:

Static Public Member Functions

- static void writeLogInfo (String info, String filePath)

  Write a new log information line in the file.
- static synchronized void **printDebug** (String message)
- static synchronized void writeDebugInfo (String info)

  Write a new debug information line in the file.
- static void writeExceptionLog (Throwable e)

  Writes an exception to the same log as writeDebugInfo.

- static synchronized void writeLogfile (String info, String path)

  Write new log information into a file.
- static boolean deleteLogDir ()

  Remove all file in the log directory.

Static Public Attributes

- static String LogFolder

  Log folder where the log files will be written.
- static String Prefix = ""

  Prefix of the log file.

5.34.1 Detailed Description

This class is used to write log file.

5.34.2 Member Function Documentation

5.34.2.1 static boolean popjava.util.LogWriter.deleteLogDir ( ) `[static]`

Remove all file in the log directory.

Returns

true if the action is succeed

5.34.2.2 static synchronized void popjava.util.LogWriter.writeDebugInfo (String *info*) `[static]`

Write a new debug information line in the file.

Parameters

info	Information to write

Here is the caller graph for this function:

5.34.2.3 static void popjava.util.LogWriter.writeExceptionLog (Throwable *e*) `[static]`

Writes an exception to the same log as writeDebugInfo.

The complete backtrace is logged.

Parameters

e	The exception to be logged

Here is the caller graph for this function:

5.34.2.4 static synchronized void popjava.util.LogWriter.writeLogfile (String *info*, String *path*) `[static]`

Write new log information into a file.

Parameters

info	Information to write
path	Path of the file

Here is the caller graph for this function:

5.34.2.5 static void popjava.util.LogWriter.writeLogInfo (String *info*, String *filePath*) `[static]`

Write a new log information line in the file.

Parameters

info	Information to write
filePath	Path of the log file

Here is the call graph for this function:

5.35 popjava.base.MessageHeader Class Reference

Message header is include in all communication between Interface and Broker side.

Collaboration diagram for popjava.base.MessageHeader:

Public Member Functions

- MessageHeader (int classId, int methodId, int semantics)

  Initialize a new message header with parameters.
- MessageHeader ()

  Initialize a new message header for sending a response.
- MessageHeader (int exceptionCode)

  Initialize a new message header for sending an exception.
- int getRequestType ()

  Get the request type.
- void setRequestType (int requestType)

  Set the request type in the header message.
- int getClassId ()

  Get the class identifier stored in this message header.
- void setClassId (int classId)

  Set the class identifier in the message header.
- int getMethodId ()

  Get the method identifier set in this message header.
- void setMethodId (int methodId)

  Set the method identifier in the message header.
- int getSenmatics ()

  Get the semantics stored in this message header.
- void setSenmatics (int senmatics)

  Set the semantic in the message header.
- int getExceptionCode ()

  Get the exception code stored in this message header.
- void setExceptionCode (int exceptionCode)

  Set the exception code in this message header.
- String toString ()

  Format message header as a string value.

Static Public Attributes

- static final int **Request** = 0
- static final int **Response** = 1
- static final int **Exception** = 2
- static final int **BindStatusCall** = 0
- static final int **AddRefCall** = 1
- static final int **DecRefCall** = 2
- static final int **GetEncodingCall** = 3
- static final int **KillCall** = 4
- static final int **ObjectAliveCall** = 5
- static final int **HeaderLength** = 20

Protected Attributes

- int **requestType**
- int **classId**
- int **methodId**
- int **semantics**
- int **exceptionCode**

5.35.1 Detailed Description

Message header is include in all communication between Interface and Broker side.

5.35.2 Constructor & Destructor Documentation

5.35.2.1 popjava.base.MessageHeader.MessageHeader (int *classId,* int *methodId,* int *semantics*)

Initialize a new message header with parameters.

Parameters

classId	Identifier of the parallel class
methodId	Identifier of the method in the parallel class
semantics	Invocation semantic of the method in the parallel class

5.35.2.2 popjava.base.MessageHeader.MessageHeader (int *exceptionCode*)

Initialize a new message header for sending an exception.

Parameters

exceptionCode	code of the exception to be sent

5.35.3 Member Function Documentation

5.35.3.1 int popjava.base.MessageHeader.getClassId ( )

Get the class identifier stored in this message header.

Returns

The class identifier

Here is the caller graph for this function:

5.35.3.2 int popjava.base.MessageHeader.getExceptionCode ( )

Get the exception code stored in this message header.

Returns

The exception code stored in the message header

Here is the caller graph for this function:

5.35.3.3 int popjava.base.MessageHeader.getMethodId ( )

Get the method identifier set in this message header.

Returns

the method identifier

Here is the caller graph for this function:

5.35.3.4 int popjava.base.MessageHeader.getRequestType ( )

Get the request type.

Returns

The request type stored in the message header

Here is the caller graph for this function:

5.35.3.5 int popjava.base.MessageHeader.getSenmatics ( )

Get the semantics stored in this message header.

Returns

The semantic stored in the message header

Here is the caller graph for this function:

5.35.3.6 void popjava.base.MessageHeader.setClassId (int *classId*)

Set the class identifier in the message header.

Parameters

classId	The class identifier to be set

Here is the caller graph for this function:

5.35.3.7 void popjava.base.MessageHeader.setExceptionCode (int *exceptionCode*)

Set the exception code in this message header.

Parameters

exceptionCode	

Here is the caller graph for this function:

5.35.3.8 void popjava.base.MessageHeader.setMethodId (int *methodId*)

Set the method identifier in the message header.

Parameters

methodId	The method identifier to be set

Here is the caller graph for this function:

5.35.3.9 void popjava.base.MessageHeader.setRequestType (int *requestType*)

Set the request type in the header message.

Request type can be Request, Response or Exception

Parameters

requestType	type of the request

Here is the caller graph for this function:

5.35.3.10 void popjava.base.MessageHeader.setSenmatics (int *senmatics*)

Set the semantic in the message header.

Parameters

senmatics	Semantic to be set

Here is the caller graph for this function:

5.36 popjava.base.MethodInfo Class Reference

This class represents all the informations about a method in a parallel object.

Collaboration diagram for popjava.base.MethodInfo:

Public Member Functions

- MethodInfo (int classId, int methodId)

    Create a new MethodInfo with the given values.
- int getMethodId ()

    Get the method unique identifier stored in this object.

- int getClassId ()

    Get the class unique identifier stored in this object.
- boolean equals (Object obj)

    Check if if the given object is equals to this MethodInfo.
- String toString ()

    Format the MethodInfo as a string value.

5.36.1 Detailed Description

This class represents all the informations about a method in a parallel object.

This class is used to retrieve the method to invoke on a parallel object

5.36.2 Constructor & Destructor Documentation

5.36.2.1 popjava.base.MethodInfo.MethodInfo (int *classId,* int *methodId*)

Create a new MethodInfo with the given values.

Parameters

classId	The class unique identifier
methodId	The method unique identifier

Here is the caller graph for this function:

5.36.3 Member Function Documentation

5.36.3.1 boolean popjava.base.MethodInfo.equals (Object *obj*)

Check if if the given object is equals to this MethodInfo.

Parameters

obj	The object to compare with

Returns

true is they are equal

Here is the call graph for this function:

Here is the caller graph for this function:

5.36.3.2 int popjava.base.MethodInfo.getClassId ( )

Get the class unique identifier stored in this object.

Returns

The class unique identifier

Here is the caller graph for this function:

5.36.3.3 int popjava.base.MethodInfo.getMethodId ( )

Get the method unique identifier stored in this object.

Returns

The method unique identifier

Here is the caller graph for this function:

5.37 popjava.baseobject.ObjectDescription Class Reference

This class represents the object description for a parallel object.

Inheritance diagram for popjava.baseobject.ObjectDescription:

Collaboration diagram for popjava.baseobject.ObjectDescription:

Public Member Functions

- ObjectDescription ()

  Create a new empty instance of ObjectDescription.
- void setDirectory (String d)

  Set the directory OD.
- void setPower (float required, float min)

  Set the power OD by ODElement.
- void setMemory (float required, float min)

  Set the memory OD by ODElement.
- void setBandwidth (float required, float min)

  Set the bandwidth OD by ODELement.
- void setWallTime (float walltime)

  Set the walltime OD.
- void manual (boolean a)

  Set the manual OD.
- void setSearch (int maxdepth, int maxsize, int waittime)

  Set the search OD values.
- int getSearchMaxDepth ()

  Get the OD search maximum depth value.
- int getSearchMaxSize ()

  Get the OD search maximum size value.
- int getSearchWaitTime ()

  Get the OD search waiting time value.
- boolean isSearchSet ()

  Say if the OD search is set.
- void setHostname (String hostname)

  Set the OD host name value.
- void setHostarch (String arch)

  Set the OD host architecture value.
- String getHostarch ()

  Get the OD host architecture value.
- void setHostcore (String core)

  Set the OD host core value.

- String getHostcore ()

  Get the OD host core value.
- void setHostuser (String user)

  Set the OD host user value.
- String getHostuser ()

  Get the OD host user value.
- void setBatch (String batch)

  Set the OD batch value.
- String getBatch ()

  Get the OD batch value.
- void setJobUrl (String jobUrl)

  Set the OD JobUrl value.
- void setCodeFile (String codeFile)

  Set the OD Code file value.
- void setProtocol (String protocol)

  Set the OD protocol value.
- void setEncoding (String encoding)

  Set the OD encoding value.
- void setPlatform (String platform)

  Set the OD platform value.
- void setJVMParamters (String parameters)

  Sets the jvm parameters that should be used when creating this object.
- float getPowerMin ()

  Get the OD power value.
- float **getPowerReq** ()
- float **getMemoryMin** ()
- float **getMemoryReq** ()
- float **getBandwidthMin** ()
- float **getBandwidthReq** ()
- float getWallTime ()

  Get the OD walltime value.
- String getHostName ()

  Get the OD hostname value.
- String getJobUrl ()

  Get the OD JobUrl value.
- String getProtocol ()

  Get the OD protocol value.
- String getEncoding ()

  Get the OD encoding value.
- String getJVMParameters ()

  Returns the parameters that should be used when creating the JVM for this object.
- String getPlatform ()

  Get he OD platform value.
- String getCodeFile ()

  Get the OD code file value.
- void setValue (String key, String value)

  Set a specific attribute in the list.
- String getValue (String key)

  Get a specific attribute from the list.
- void removeValue (String key)

  Remove a specific attribute from the list.

- void removeAllAttributes ()

  Remove all attributes from the list.
- boolean isEmpty ()

  Check if the current object is empty.
- boolean deserialize (POPBuffer buffer)

  Deserialize the object description from the buffer.
- boolean serialize (POPBuffer buffer)

  Serialize the object description into the buffer.
- void merge (ObjectDescription od)

  Merge another object description with this object description.
- String toString ()

  Format the object description as a string value.

Protected Member Functions

- void finalize () throws Throwable

  Method called before destruction.

Protected Attributes

- boolean **isLocalJob**
- boolean **isManual**
- int **max_depth**
- int **wait_time**
- int **max_size**
- boolean **searchSet**
- String **hostarch**
- String **hostcore**
- String **hostuser**
- float **power_min**
- float **power_req**
- float **bandwidth_min**
- float **bandwidth_req**
- float **memory_min**
- float **memory_req**
- float **wallTime**
- String **encoding**
- String **protocol**
- String **platform**
- String **hostName**
- String **jobUrl**
- String **codeFile**
- String **cwd**
- String **batch**
- String **jvmParamters**

5.37.1 Detailed Description

This class represents the object description for a parallel object.

The object description is the resource requirements for a specific parallel object.

5.37.2 Member Function Documentation

5.37.2.1 String popjava.baseobject.ObjectDescription.getBatch ( )

Get the OD batch value.

Returns

batch value set in this OD

5.37.2.2 String popjava.baseobject.ObjectDescription.getCodeFile ( )

Get the OD code file value.

Returns

codefile set in this OD

Here is the caller graph for this function:

5.37.2.3 String popjava.baseobject.ObjectDescription.getEncoding ( )

Get the OD encoding value.

Returns

encoding set in this OD

Here is the caller graph for this function:

5.37.2.4 String popjava.baseobject.ObjectDescription.getHostarch ( )

Get the OD host architecture value.

Returns

host architecture value set in the OD

Here is the caller graph for this function:

5.37.2.5 String popjava.baseobject.ObjectDescription.getHostcore ( )

Get the OD host core value.

Returns

host core value set in this OD

Here is the caller graph for this function:

5.37.2.6 String popjava.baseobject.ObjectDescription.getHostName ( )

Get the OD hostname value.

Returns

hostname set in this OD

Here is the caller graph for this function:

5.37.2.7 String popjava.baseobject.ObjectDescription.getHostuser ( )

Get the OD host user value.

Returns

host user value set in this OD

Here is the caller graph for this function:

5.37.2.8 String popjava.baseobject.ObjectDescription.getJobUrl ( )

Get the OD JobUrl value.

Returns

joburl set in this OD

Here is the caller graph for this function:

5.37.2.9 String popjava.baseobject.ObjectDescription.getJVMParameters ( )

Returns the parameters that should be used when creating the JVM for this object.

Returns

5.37.2.10 String popjava.baseobject.ObjectDescription.getPlatform ( )

Get he OD platform value.

Returns

platform set in this OD

Here is the caller graph for this function:

5.37.2.11 float popjava.baseobject.ObjectDescription.getPowerMin ( )

Get the OD power value.

Returns

power value set in this OD Get the OD memory value
memory value set in this OD Get the OD bandwith value
bandwith value set in this OD

Here is the caller graph for this function:

5.37.2.12 String popjava.baseobject.ObjectDescription.getProtocol ( )

Get the OD protocol value.

Returns

protocol set in this OD

Here is the caller graph for this function:

5.37.2.13 int popjava.baseobject.ObjectDescription.getSearchMaxDepth ( )

Get the OD search maximum depth value.

Returns

maximum depth value set in the OD

Here is the caller graph for this function:

5.37.2.14 int popjava.baseobject.ObjectDescription.getSearchMaxSize ( )

Get the OD search maximum size value.

Returns

maximum size value set in the OD

Here is the caller graph for this function:

5.37.2.15 int popjava.baseobject.ObjectDescription.getSearchWaitTime ( )

Get the OD search waiting time value.

Returns

waiting time value set in the OD

Here is the caller graph for this function:

5.37.2.16 String popjava.baseobject.ObjectDescription.getValue (String *key*)

Get a specific attribute from the list.

Parameters

key	Key of the specific attribute

Returns

Value of the attribute or an empty string

5.37.2.17 float popjava.baseobject.ObjectDescription.getWallTime ( )

Get the OD walltime value.

Returns

walltime value set in this OD

Here is the caller graph for this function:

5.37.2.18 boolean popjava.baseobject.ObjectDescription.isEmpty ( )

Check if the current object is empty.

Returns

true if empty

5.37.2.19 boolean popjava.baseobject.ObjectDescription.isSearchSet ( )

Say if the OD search is set.

Returns

true if the OD search is set

5.37.2.20 void popjava.baseobject.ObjectDescription.manual (boolean *a*)

Set the manual OD.

Parameters

a	true = manual

Here is the caller graph for this function:

5.37.2.21 void popjava.baseobject.ObjectDescription.merge (**ObjectDescription** *od*)

Merge another object description with this object description.

Parameters

od	The object description to be merged with this one

Here is the call graph for this function:

Here is the caller graph for this function:

5.37.2.22 void popjava.baseobject.ObjectDescription.removeValue (String *key*)

Remove a specific attribute from the list.

Parameters

key	Key of the attribute to be removed

5.37.2.23 void popjava.baseobject.ObjectDescription.setBandwidth (float *required,* float *min*)

Set the bandwidth OD by ODELement.

Parameters

bandwidth	ODElement specifying the required and minimum values Set the bandwidth OD by values
required	The required bandwidth
min	The minimum bandwidth

Here is the caller graph for this function:

5.37.2.24 void popjava.baseobject.ObjectDescription.setBatch (String *batch*)

Set the OD batch value.

Parameters

batch	batch value

Here is the caller graph for this function:

5.37.2.25 void popjava.baseobject.ObjectDescription.setCodeFile (String *codeFile*)

Set the OD Code file value.

Parameters

codeFile	Get the OD code file value

Here is the caller graph for this function:

5.37.2.26 void popjava.baseobject.ObjectDescription.setDirectory (String *d*)

Set the directory OD.

Parameters

d	Specific directory

Here is the caller graph for this function:

5.37.2.27 void popjava.baseobject.ObjectDescription.setEncoding (String *encoding*)

Set the OD encoding value.

Parameters

encoding	encoding to be used to communicate with the object

Here is the caller graph for this function:

5.37.2.28 void popjava.baseobject.ObjectDescription.setHostarch (String *arch*)

Set the OD host architecture value.

Parameters

arch	host architecture to execute the object

Here is the caller graph for this function:

5.37.2.29 void popjava.baseobject.ObjectDescription.setHostcore (String *core*)

Set the OD host core value.

Parameters

core	core value

Here is the caller graph for this function:

5.37.2.30 void popjava.baseobject.ObjectDescription.setHostname (String *hostname*)

Set the OD host name value.

Parameters

hostname	host name to execute the object

Here is the caller graph for this function:

5.37.2.31 void popjava.baseobject.ObjectDescription.setHostuser (String *user*)

Set the OD host user value.

Parameters

user	USer to execute the object

Here is the caller graph for this function:

5.37.2.32 void popjava.baseobject.ObjectDescription.setJobUrl (String *jobUrl*)

Set the OD JobUrl value.

Parameters

jobUrl	job manager access point

Here is the caller graph for this function:

5.37.2.33 void popjava.baseobject.ObjectDescription.setJVMParamters (String *parameters*)

Sets the jvm parameters that should be used when creating this object.

Parameters

parameters	

5.37.2.34 void popjava.baseobject.ObjectDescription.setMemory (float *required,* float *min*)

Set the memory OD by ODElement.

Parameters

memory	ODElement specifying the required and minimum values Set the memory OD by values
required	The required memory
min	The minimum memory

Here is the caller graph for this function:

5.37.2.35 void popjava.baseobject.ObjectDescription.setPlatform (String *platform*)

Set the OD platform value.

Parameters

platform	platform on which the object must be executed

Here is the caller graph for this function:

5.37.2.36 void popjava.baseobject.ObjectDescription.setPower (float *required,* float *min*)

Set the power OD by ODElement.

Parameters

power	ODElement specifying the required and minimum values Set the power OD by values
required	The required power
min	The minimum power

Here is the caller graph for this function:

5.37.2.37 void popjava.baseobject.ObjectDescription.setProtocol (String *protocol*)

Set the OD protocol value.

Parameters

protocol	protocol to be used to communicate with the object

Here is the caller graph for this function:

5.37.2.38 void popjava.baseobject.ObjectDescription.setSearch (int *maxdepth,* int *maxsize,* int *waittime*)

Set the search OD values.

Parameters

maxdepth	The maximum depth for the search algorithm
maxsize	The maximum size of a search request
waittime	The waiting time of the search algorithm (0 = take the first answer)

Here is the caller graph for this function:

5.37.2.39 void popjava.baseobject.ObjectDescription.setValue (String *key,* String *value*)

Set a specific attribute in the list.

Parameters

key	Key for this attribute
value	value for this attribute

Here is the caller graph for this function:

5.37.2.40 void popjava.baseobject.ObjectDescription.setWallTime (float *walltime*)

Set the walltime OD.

Parameters

walltime	time allocated for the wall execution

Here is the caller graph for this function:

5.38 popjava.dataswaper.ObjectDescriptionInput Class Reference

Compatible implementation of the ObjectDescription POP-Java object for POP-C++.

Inheritance diagram for popjava.dataswaper.ObjectDescriptionInput:

Collaboration diagram for popjava.dataswaper.ObjectDescriptionInput:

Public Member Functions

- ObjectDescriptionInput ()

  Create a new empty instance of ObjectDescriptionInput.
- ObjectDescriptionInput (ObjectDescription od)

  Create a new instance of ObjectDescriptionInput from an ObjectDescritption.
- void setPower (float required, float min)

  Set the power OD by ODElement.
- void setMemory (float required, float min)

  Set the memory OD by ODElement.
- void setBandwidth (float required, float min)

  Set the bandwidth OD by ODELement.
- void setWallTime (float walltime)

  Set the walltime OD.
- void setHostname (String hostname)

  Set the OD host name value.
- void setJobUrl (String jobUrl)

  Set the OD JobUrl value.
- void setCodeFile (String codeFile)

  Set the OD Code file value.
- void setProtocol (String protocol)

  Set the OD protocol value.
- void setEncoding (String encoding)

  Set the OD encoding value.
- void setPlatform (String platform)

  Set the OD platform value.
- float getWallTime ()

  Get the OD power value.
- String getHostName ()

  Get the OD hostname value.
- String getJobUrl ()

  Get the OD JobUrl value.
- String getProtocol ()

  Get the OD protocol value.

- String getEncoding ()

  Get the OD encoding value.
- String getPlatform ()

  Get he OD platform value.
- String getCodeFile ()

  Get the OD code file value.
- void setValue (String key, String value)

  Set a specific attribute in the list.
- String getValue (String key)

  Get a specific attribute from the list.
- void removeValue (String key)

  Remove a specific attribute from the list.
- void removeAllAttributes ()

  Remove all attributes from the list.
- boolean isEmpty ()

  Check if the current object is empty.
- void setSearch (int depth, int size, int waittime)

  Set the search OD values.
- boolean serialize (POPBuffer buffer)

  Serialize the object description into the buffer.
- void merge (ObjectDescription od)

  Merge another object description with this object description.
- String toString ()

  Format the object description as a string value.
- boolean deserialize (POPBuffer buffer)

  Deserialize the object description from the buffer.

Protected Member Functions

- void finalize () throws Throwable

  Method called before destruction.

Protected Attributes

- float **power_min**
- float **power_req**
- float **bandwidth_min**
- float **bandwidth_req**
- float **memory_min**
- float **memory_req**
- float **wallTime**
- boolean **isManual**
- String **cwd**
- int **searchMaxDepth**
- int **searchMaxReq**
- int **searchWaitingtime**
- String **url**
- String **user**
- String **core**
- String **batch**
- String **encoding**

- String **arch**
- String **hostName**
- String **jobUrl**
- String **codeFile**
- String **platform**
- String **protocol**

5.38.1 Detailed Description

Compatible implementation of the ObjectDescription POP-Java object for POP-C++.

5.38.2 Constructor & Destructor Documentation

5.38.2.1 popjava.dataswaper.ObjectDescriptionInput.ObjectDescriptionInput (**ObjectDescription** *od*)

Create a new instance of ObjectDescriptionInput from an ObjectDescritption.

Parameters

od	The base object description

Here is the call graph for this function:

5.38.3 Member Function Documentation

5.38.3.1 String popjava.dataswaper.ObjectDescriptionInput.getCodeFile ( )

Get the OD code file value.

Returns

codefile set in this OD

5.38.3.2 String popjava.dataswaper.ObjectDescriptionInput.getEncoding ( )

Get the OD encoding value.

Returns

encoding set in this OD

5.38.3.3 String popjava.dataswaper.ObjectDescriptionInput.getHostName ( )

Get the OD hostname value.

Returns

hostname set in this OD

5.38.3.4 String popjava.dataswaper.ObjectDescriptionInput.getJobUrl ( )

Get the OD JobUrl value.

Returns

joburl set in this OD

5.38.3.5 String popjava.dataswaper.ObjectDescriptionInput.getPlatform ( )

Get he OD platform value.

Returns

platform set in this OD

5.38.3.6 String popjava.dataswaper.ObjectDescriptionInput.getProtocol ( )

Get the OD protocol value.

Returns

protocol set in this OD

5.38.3.7 String popjava.dataswaper.ObjectDescriptionInput.getValue (String *key*)

Get a specific attribute from the list.

Parameters

key	Key of the specific attribute

Returns

Value of the attribute or an empty string

5.38.3.8 float popjava.dataswaper.ObjectDescriptionInput.getWallTime ( )

Get the OD power value.

Returns

power value set in this OD Get the OD memory value
memory value set in this OD Get the OD bandwith value
bandwith value set in this OD Get the OD walltime value
walltime value set in this OD

5.38.3.9 boolean popjava.dataswaper.ObjectDescriptionInput.isEmpty ( )

Check if the current object is empty.

Returns

true if empty

5.38.3.10 void popjava.dataswaper.ObjectDescriptionInput.merge (ObjectDescription *od*)

Merge another object description with this object description.

Parameters

od	The object description to be merged with this one

Here is the call graph for this function:

5.38.3.11 void popjava.dataswaper.ObjectDescriptionInput.removeValue (String *key*)

Remove a specific attribute from the list.

Parameters

key	Key of the attribute to be removed

5.38.3.12 void popjava.dataswaper.ObjectDescriptionInput.setBandwidth (float *required,* float *min*)

Set the bandwidth OD by ODELement.

Parameters

bandwidth	ODElement specifying the required and minimum values Set the bandwidth OD by values
required	The required bandwidth
min	The minimum bandwidth

Here is the caller graph for this function:

5.38.3.13 void popjava.dataswaper.ObjectDescriptionInput.setCodeFile (String *codeFile*)

Set the OD Code file value.

Parameters

codeFile	Get the OD code file value

5.38.3.14 void popjava.dataswaper.ObjectDescriptionInput.setEncoding (String *encoding*)

Set the OD encoding value.

Parameters

encoding	encoding to be used to communicate with the object

5.38.3.15 void popjava.dataswaper.ObjectDescriptionInput.setHostname (String *hostname*)

Set the OD host name value.

Parameters

hostname	host name to execute the object

5.38.3.16 void popjava.dataswaper.ObjectDescriptionInput.setJobUrl (String *jobUrl*)

Set the OD JobUrl value.

Parameters

jobUrl	job manager access point

5.38.3.17 void popjava.dataswaper.ObjectDescriptionInput.setMemory (float *required,* float *min*)

Set the memory OD by ODElement.

Parameters

memory	ODElement specifying the required and minimum values Set the memory OD by values
required	The required memory
min	The minimum memory

Here is the caller graph for this function:

5.38.3.18 void popjava.dataswaper.ObjectDescriptionInput.setPlatform (String *platform*)

Set the OD platform value.

Parameters

platform	platform on which the object must be executed

5.38.3.19 void popjava.dataswaper.ObjectDescriptionInput.setPower (float *required,* float *min*)

Set the power OD by ODElement.

Parameters

power	ODElement specifying the required and minimum values Set the power OD by values
required	The required power
min	The minimum power

Here is the caller graph for this function:

5.38.3.20 void popjava.dataswaper.ObjectDescriptionInput.setProtocol (String *protocol*)

Set the OD protocol value.

Parameters

protocol	protocol to be used to communicate with the object

5.38.3.21 void popjava.dataswaper.ObjectDescriptionInput.setSearch (int *depth,* int *size,* int *waittime*)

Set the search OD values.

Parameters

depth	The maximum depth for the search algorithm
size	The maximum size of a search request
waittime	The waiting time of the search algorithm (0 = take the first answer)

5.38.3.22 void popjava.dataswaper.ObjectDescriptionInput.setValue (String *key*, String *value*)

Set a specific attribute in the list.

Parameters

key	Key for this attribute
value	value for this attribute

5.38.3.23 void popjava.dataswaper.ObjectDescriptionInput.setWallTime (float *walltime*)

Set the walltime OD.

Parameters

walltime	time allocated for the wall execution

5.39 popjava.baseobject.ODElement Class Reference

This class represents an ODElement for the object description.

Collaboration diagram for popjava.baseobject.ODElement:

Public Member Functions

- ODElement ()

  Constructor a POPODElement, the require value and min value are 0.
- ODElement (float un, float deux)

  Create a new ODElement with given values.
- void serialize (POPBuffer buffer)

  Serialize the ODElement into the buffer.
- void setRequiredValue (float requiredValue)

  Set the required value for this ODElement.
- void setMinValue (float minValue)

  Set the minimum value of this element.
- float getRequiredValue ()

  Get the required value of this ODElement.
- float getMinValue ()

  Get the minimum value of this ODElement.
- void set (float requiredValue, float minValue)

  Set the values of the ODElement.
- void set (ODElement od)

  Set values with an ODElement.
- boolean isEmpty ()

  Check if the current object is empty.
- String toString ()

  Format the ODElement as a string value.

Static Public Member Functions

- static ODElement deserialize (POPBuffer buffer)

    Deserilize the ODElement from the buffer.

5.39.1 Detailed Description

This class represents an ODElement for the object description.

An ODElement is an element that has a required and a minimum value. For example, the power required for an object is set trough an ODElement. The power must have a required and a minimum value.

5.39.2 Constructor & Destructor Documentation

5.39.2.1 popjava.baseobject.ODElement.ODElement (float *un,* float *deux*)

Create a new ODElement with given values.

Parameters

requiredValue	Required value for this OD element
minValue	Minimum value for this OD element

5.39.3 Member Function Documentation

5.39.3.1 static **ODElement** popjava.baseobject.ODElement.deserialize (**POPBuffer** *buffer*) `[static]`

Deserilize the ODElement from the buffer.

Parameters

buffer	Buffer to deserialize from

Returns

the ODElement deserilized

Here is the call graph for this function:

5.39.3.2 float popjava.baseobject.ODElement.getMinValue ( )

Get the minimum value of this ODElement.

Returns

the minimum value

5.39.3.3 float popjava.baseobject.ODElement.getRequiredValue ( )

Get the required value of this ODElement.

Returns

the required value

5.39.3.4 boolean popjava.baseobject.ODElement.isEmpty ( )

Check if the current object is empty.

Returns

true if the current object is empty

Here is the caller graph for this function:

5.39.3.5 void popjava.baseobject.ODElement.serialize (POPBuffer *buffer*)

Serialize the ODElement into the buffer.

Parameters

buffer	The buffer to serialize in

Here is the call graph for this function:

5.39.3.6 void popjava.baseobject.ODElement.set (float *requiredValue,* float *minValue*)

Set the values of the ODElement.

Parameters

requiredValue	Required value
minValue	Minimum value

5.39.3.7 void popjava.baseobject.ODElement.set (ODElement *od*)

Set values with an ODElement.

Parameters

od	The ODElement with new values

5.39.3.8 void popjava.baseobject.ODElement.setMinValue (float *minValue*)

Set the minimum value of this element.

Parameters

minValue	

5.39.3.9 void popjava.baseobject.ODElement.setRequiredValue (float *requiredValue*)

Set the required value for this ODElement.

Parameters

requiredValue	Required value

5.40 popjava.PJMethodFilter Class Reference

This class is a method filter for the PJMethodHandler.

Inheritance diagram for popjava.PJMethodFilter:

Collaboration diagram for popjava.PJMethodFilter:

Public Member Functions

- PJMethodFilter ()

  Default constructor.
- boolean isHandled (Method m)

  Check if a method is handled by the method handler.

5.40.1 Detailed Description

This class is a method filter for the PJMethodHandler.

5.40.2 Member Function Documentation

5.40.2.1 boolean popjava.PJMethodFilter.isHandled (Method *m*)

Check if a method is handled by the method handler.

Parameters

m	The method to check

Returns

true if the method is handled

5.41 popjava.PJMethodHandler Class Reference

This class is responsible to invoke methods on the parallel object.

Inheritance diagram for popjava.PJMethodHandler:

Collaboration diagram for popjava.PJMethodHandler:

Public Member Functions

- PJMethodHandler ()

  Creates a new instance of PJComboxMethodHandler.
- PJMethodHandler (POPObject popObject)

  Associate an POPObject with this handler.
- boolean popConstructor (Class<?> targetClass, Object...argvs) throws POPException, NoSuchMethod-Exception

  Construct a parallel object.
- boolean bindObject (POPAccessPoint accesspoint) throws POPException

  Bind the interface-side with the broker-side.
- Object invoke (Object self, Method m, Method proceed, Object[] argvs) throws Throwable

Invoke a method on an object.

- String toString ()

  Format a string of this object.

Protected Attributes

- final int constructorSemanticId = 21

  Default semantic of a constructor.
- POPObject **popObjectInfo** = null

Additional Inherited Members

5.41.1 Detailed Description

This class is responsible to invoke methods on the parallel object.

5.41.2 Constructor & Destructor Documentation

5.41.2.1 popjava.PJMethodHandler.PJMethodHandler (**POPObject** *popObject*)

Associate an POPObject with this handler.

Parameters

popObject	The POPObject to associate

5.41.3 Member Function Documentation

5.41.3.1 boolean popjava.PJMethodHandler.bindObject (**POPAccessPoint** *accesspoint*) throws **POPException**

Bind the interface-side with the broker-side.

Parameters

accesspoint	Access point of the broker-side

Returns

true if the binding is succeed

Exceptions

POPException	throw an exception if the binding is not succeed

Here is the call graph for this function:

Here is the caller graph for this function:

5.41.3.2 Object popjava.PJMethodHandler.invoke (Object *self,* Method *m,* Method *proceed,* Object[ ] *argvs*) throws Throwable

Invoke a method on an object.

Parameters

self	The object to call the method
m	The method to be called
proceed	The method to proceed the call
argvs	Arguments of the methods

Returns

Any object if the method has a return value

Exceptions

Throw	any exception if the method throws any exception

Here is the call graph for this function:

5.41.3.3 boolean popjava.PJMethodHandler.popConstructor (Class<?> *targetClass*, Object... *argvs*) throws POPException, NoSuchMethodException

Construct a parallel object.

Parameters

targetClass	Class to be created
argvs	Arguments of the constructor

Returns

true if the object is instantiate

Exceptions

POPException	Thrown if any problem occurred during the parallel object creation
NoSuchMethodException	Thrown if the constructor is not found

Here is the call graph for this function:

Here is the caller graph for this function:

5.42 popjava.PJProxyFactory Class Reference

POP-Java Proxy Factory : this class provide methods to create a proxy factory for a specified class.

Inheritance diagram for popjava.PJProxyFactory:

Collaboration diagram for popjava.PJProxyFactory:

Public Member Functions

- PJProxyFactory (Class<?> targetClass)

  Create a new proxy factory for the specified class.
- Object newPOPObject (Object...argvs) throws POPException

  Create a new object from the factory.
- Object newPOPObject (ObjectDescription od, Object...argvs) throws POPException

Create a new object from specific class and object description.

- Object bindPOPObject (POPAccessPoint accessPoint) throws POPException

  Bind an Interface to her parallel object (her associated Broker)
- Object newActiveFromBuffer (POPBuffer buffer) throws POPException

  Recover a parallel object from the buffer.

Protected Attributes

- Class<?> targetClass

  Target class to create a proxy.

5.42.1 Detailed Description

POP-Java Proxy Factory : this class provide methods to create a proxy factory for a specified class.

This class uses the Javassit library.

5.42.2 Constructor & Destructor Documentation

5.42.2.1 popjava.PJProxyFactory.PJProxyFactory (Class<?> *targetClass*)

Create a new proxy factory for the specified class.

Parameters

targetClass	: Class to be created by the Factory

5.42.3 Member Function Documentation

5.42.3.1 Object popjava.PJProxyFactory.bindPOPObject (POPAccessPoint *accessPoint*) throws POPException

Bind an Interface to her parallel object (her associated Broker)

Parameters

accessPoint	: The accesspoint of the broker

Returns

ProxyObject which represent the Interface side

Exceptions

POPException	: if anything goes wrong

Here is the call graph for this function:

Here is the caller graph for this function:

5.42.3.2 Object popjava.PJProxyFactory.newActiveFromBuffer (POPBuffer *buffer*) throws POPException

Recover a parallel object from the buffer.

Parameters

buffer	: buffer from which the object is recovered

Returns

the object recovered

Exceptions

POPException	

Here is the call graph for this function:

Here is the caller graph for this function:

5.42.3.3 Object popjava.PJProxyFactory.newPOPObject (Object... *argvs*) throws POPException

Create a new object from the factory.

Parameters

argvs	: arguments to pass trough the constructor of the specific object

Returns

the instance of the object

Exceptions

POPException	

Here is the caller graph for this function:

5.42.3.4 Object popjava.PJProxyFactory.newPOPObject (ObjectDescription *od,* Object... *argvs*) throws POPException

Create a new object from specific class and object description.

Parameters

od	: Object description with the resource requirements
argvs	: arguments to pass trough the constructor of the specific object

Returns

the instance of the object

Exceptions

POPException	

Here is the call graph for this function:

5.43 popjava.baseobject.POPAccessPoint Class Reference

This class represents multiple access to the broker-side parallel object.

Inheritance diagram for popjava.baseobject.POPAccessPoint:

Collaboration diagram for popjava.baseobject.POPAccessPoint:

Public Member Functions

- POPAccessPoint ()

  Create a new POPAccessPoint()
- POPAccessPoint (boolean initialize)

  Create a new POPAccessPoint an make some initialization tasks.
- POPAccessPoint (String accessString)

  Create a new POPAccessPoint with a formatted string.
- boolean serialize (POPBuffer buffer)

  Serialize the object into the buffer to be sent over the network.
- boolean deserialize (POPBuffer buffer)

  Deserialize the object from the buffer received from the network.
- void addAccessPoint (AccessPoint accessPoint)

  Add an access point to the POPAccessPoint.
- boolean isEmpty ()

  Check if the current object is empty.
- String toString ()

  Format the POPAccessPoint to a string value.
- void setAccessString (String accessString)

  Add an access point by a formatted string.
- int size ()

  Get the number of different access points.
- AccessPoint get (int index)

  Get the access point at specified index.

Protected Attributes

- ArrayList< AccessPoint > accessPoints = new ArrayList<AccessPoint>()

  The list of the different access points.

5.43.1 Detailed Description

This class represents multiple access to the broker-side parallel object.

5.43.2 Constructor & Destructor Documentation

5.43.2.1 popjava.baseobject.POPAccessPoint.POPAccessPoint (boolean *initialize*)

Create a new POPAccessPoint an make some initialization tasks.

Parameters

initialize	Set to false if you don't want the initialization

Here is the call graph for this function:

5.43.2.2 popjava.baseobject.POPAccessPoint.POPAccessPoint (String *accessString*)

Create a new POPAccessPoint with a formatted string.

Parameters

accessString	Formatted string to create the POPAccessPoint

Here is the call graph for this function:

5.43.3 Member Function Documentation

5.43.3.1 void popjava.baseobject.POPAccessPoint.addAccessPoint (AccessPoint *accessPoint*)

Add an access point to the POPAccessPoint.

Parameters

accessPoint	New access point to be added

Here is the caller graph for this function:

5.43.3.2 AccessPoint popjava.baseobject.POPAccessPoint.get (int *index*)

Get the access point at specified index.

Parameters

index	index of the access point to return

Returns

the access points at the specified index

Here is the caller graph for this function:

5.43.3.3 boolean popjava.baseobject.POPAccessPoint.isEmpty ( )

Check if the current object is empty.

Returns

true is the current object is not set

Here is the caller graph for this function:

5.43.3.4 void popjava.baseobject.POPAccessPoint.setAccessString (String *accessString*)

Add an access point by a formatted string.

Parameters

accessString	Formatted string to be added as an access point

Here is the call graph for this function:

Here is the caller graph for this function:

5.43.3.5 int popjava.baseobject.POPAccessPoint.size ( )

Get the number of different access points.

Returns

Number of access points

Here is the caller graph for this function:

5.44 popjava.serviceadapter.POPAppService Class Reference

Partial POP-Java class implementation to be used with the POP-C++ runtime This class declares the necessary methods to use the AppService parallel object of POP-C++.

Inheritance diagram for popjava.serviceadapter.POPAppService:

Collaboration diagram for popjava.serviceadapter.POPAppService:

Public Member Functions

- POPAppService ()
  Default constructor of POPAppService.
- POPAppService (String challenge, boolean daemon, String codelocation)
  Constructor of POPAppService with parameters.
- boolean queryService (String name, POPServiceBase service)
  Ask the parallel object about the existence of a service in the runtime.
- boolean queryService (String name, POPAccessPoint service)
  Ask the parallel object about the existence of a service in the runtime.
- boolean registerService (String name, POPServiceBase newservice)
  Call the parallel object to register a new service in the runtime.
- boolean unregisterService (String name)
  Call the parallel object to unregister a service in the POP-C++ runtime.
- String **getPOPCAppID** ()

Additional Inherited Members

5.44.1 Detailed Description

Partial POP-Java class implementation to be used with the POP-C++ runtime This class declares the necessary methods to use the AppService parallel object of POP-C++.

5.44.2 Constructor & Destructor Documentation

5.44.2.1 popjava.serviceadapter.POPAppService.POPAppService ( )

Default constructor of POPAppService.

Create a POP-C++ object AppCoreService

Here is the call graph for this function:

5.44.2.2 popjava.serviceadapter.POPAppService.POPAppService (String *challenge*, boolean *daemon*, String *codelocation*)

Constructor of POPAppService with parameters.

Parameters

challenge	challenge string to stop the parallel object
daemon	say if the parallel object is running as a deamon
codelocation	path of the executable code

5.44.3 Member Function Documentation

5.44.3.1 boolean popjava.serviceadapter.POPAppService.queryService (String *name*, POPServiceBase *service*)

Ask the parallel object about the existence of a service in the runtime.

Parameters

name	Name of the service
service	Access Point of the service

Returns

true if the service exists

5.44.3.2 boolean popjava.serviceadapter.POPAppService.queryService (String *name*, POPAccessPoint *service*)

Ask the parallel object about the existence of a service in the runtime.

Parameters

name	Name of the service
service	Access Point of the service

Returns

true if the service exists

5.44.3.3 boolean popjava.serviceadapter.POPAppService.registerService (String *name*, POPServiceBase *newservice*)

Call the parallel object to register a new service in the runtime.

Parameters

name	Name of the new service
newservice	Reference of the new service

Returns

true if the service has been register correctly

5.44.3.4 boolean popjava.serviceadapter.POPAppService.unregisterService (String *name*)

Call the parallel object to unregister a service in the POP-C++ runtime.

Parameters

name	Name of the service to unregister

Returns

true if the service has been unregister correctly

5.45 popjava.annotation.POPAsyncConc Interface Reference

Collaboration diagram for popjava.annotation.POPAsyncConc:

5.45.1 Detailed Description

Author

Beat Wolf

5.46 popjava.annotation.POPAsyncMutex Interface Reference

Collaboration diagram for popjava.annotation.POPAsyncMutex:

5.47 popjava.annotation.POPAsyncSeq Interface Reference

Collaboration diagram for popjava.annotation.POPAsyncSeq:

5.48 popjava.buffer.POPBuffer Class Reference

This abstract class defined all the required methods to implement a buffer.

Inheritance diagram for popjava.buffer.POPBuffer:

Collaboration diagram for popjava.buffer.POPBuffer:

Public Member Functions

- POPBuffer ()

  Default constructor.
- abstract void reset ()

  Erase the buffer and set the pointer to the beginning.
- abstract void put (byte value)

  Insert a byte in the buffer.
- abstract void putBoolean (boolean value)

  Insert a boolean in the buffer.
- abstract void putChar (char value)

  Insert a char into the buffer.
- abstract void putInt (int value)

  Insert a int into the buffer.
- abstract void putLong (long value)

  Insert a long into the buffer.

- abstract void putShort (short value)

  Insert a short into the buffer.
- abstract void putFloat (float value)

  Insert a float value into the buffer.
- abstract void putDouble (double value)

  Insert a double value into the buffer.
- abstract void put (byte[] data)

  Insert a byte array into the buffer.
- abstract void put (byte[] data, int offset, int length)

  Insert a byte array into a specific place in the buffer.
- abstract void putByteArray (byte[] value)

  Insert a byte array into the buffer.
- abstract void putCharArray (char[] value)

  Insert a char array into the buffer.
- abstract void putBooleanArray (boolean[] value)

  Insert a boolean array into the buffer.
- abstract void putIntArray (int[] value)

  Insert a int array into the buffer.
- abstract void putShortArray (short[] value)

  Insert a short array into the buffer.
- abstract void putLongArray (long[] value)

  Insert a long array into the buffer.
- abstract void putFloatArray (float[] value)

  Insert a float array into the buffer.
- abstract void putDoubleArray (double[] value)

  Insert a double array into the buffer.
- abstract byte[] getByteArray (int length)

  Retrieve a byte array from the buffer.
- abstract char[] getCharArray (int length)

  Retrieve a char array from the buffer.
- abstract boolean[] getBooleanArray (int length)

  Retrieve a boolean array from the buffer.
- abstract int[] getIntArray (int length)

  Retrieve a int array from the buffer.
- abstract long[] getLongArray (int length)

  Retrieve a long array from the buffer.
- abstract short[] getShortArray (int length)

  Retrieve a short array from the buffer.
- abstract float[] getFloatArray (int length)

  Retrieve a float array from the buffer.
- abstract double[] getDoubleArray (int length)

  Retrieve a double array from the buffer.
- abstract void putString (String value)

  Insert a string into the buffer.
- abstract byte get ()

  Retrieve a byte from the buffer.
- abstract boolean getBoolean ()

  Retrieve a boolean from the buffer.
- abstract char getChar ()

  Retrieve a char from the buffer.
- abstract int getInt ()

Retrieve a int from the buffer.
- abstract long getLong ()

  Retrieve a long from the buffer.
- abstract short getShort ()

  Retrieve a short from the buffer.
- abstract float getFloat ()

  Retrieve a float from the buffer.
- abstract double getDouble ()

  Retrieve a double from the buffer.
- abstract String getString ()

  Retrieve a string from the buffer.
- abstract byte[ ] **array** ()
- abstract int getTranslatedInteger (byte[ ] value)

  Get a integer value of the byte array.
- abstract MessageHeader extractHeader ()

  Retrieve the message header from the buffer.
- abstract void resetToReceive ()

  Reset the buffer before reception of a new message.
- abstract int packMessageHeader ()

  Pack the message header into the buffer.
- POPBuffer (MessageHeader messageHeader)

  Constructor with given values.
- void setHeader (MessageHeader messageHeader)

  Associate a message header with this buffer.
- MessageHeader getHeader ()

  Get the message header associated with this buffer.
- int size ()

  Get the current size of the buffer.
- Object getValue (Class<?> c) throws POPException

  Retrieve an object from the buffer.
- void putValue (Object o, Class<?>c) throws POPException

  Insert an object into the buffer.
- void putArray (Object o) throws POPException

  Insert an array into the buffer.
- Object getArray (Class<?> arrayType) throws POPException

  Retrieve an array from the buffer.
- void serializeReferenceObject (Class<?> type, Object obj) throws POPException

  Insert an object reference into the buffer.
- void deserializeReferenceObject (Class<?> type, Object obj) throws POPException

  Retrieve an object reference from the buffer.
- String toIntString ()

  Return an empty string.
- String toCharString ()

  Return an empty string.

Static Public Member Functions

- static void checkAndThrow (int systemErrorCode, POPBuffer buffer) throws POPException

  Check error code and throw the right exception.

Protected Attributes

- MessageHeader messageHeader

  Each buffer send must contains a message header.
- int size = 0

  Size of the buffer in byte.

5.48.1 Detailed Description

This abstract class defined all the required methods to implement a buffer.

The buffer is responsible to encode and decode the data before sending them or receiving them over the network.

5.48.2 Constructor & Destructor Documentation

5.48.2.1 popjava.buffer.POPBuffer.POPBuffer (**MessageHeader** *messageHeader*)

Constructor with given values.

Parameters

messageHeader	Message header to be associated with this buffer

5.48.3 Member Function Documentation

5.48.3.1 static void popjava.buffer.POPBuffer.checkAndThrow (int *systemErrorCode,* **POPBuffer** *buffer*) throws **POPException** [static]

Check error code and throw the right exception.

Parameters

systemError-Code	Code of the error
buffer	Buffer from which retrieve the additional informations

Exceptions

POPException	thrown if any problem occurred

Here is the call graph for this function:

Here is the caller graph for this function:

5.48.3.2 void popjava.buffer.POPBuffer.deserializeReferenceObject (Class<?> *type,* Object *obj*) throws **POPException**

Retrieve an object reference from the buffer.

Parameters

type	Class of the object
obj	Object to be retrieved

Exceptions

POPException	thrown if the deserialization process is not going well

Here is the call graph for this function:

Here is the caller graph for this function:

5.48.3.3 abstract MessageHeader popjava.buffer.POPBuffer.extractHeader ( ) `[pure virtual]`

Retrieve the message header from the buffer.

Returns

message header retrieved in the buffer

Implemented in popjava.buffer.BufferRaw, and popjava.buffer.BufferPlugin.

Here is the caller graph for this function:

5.48.3.4 abstract byte popjava.buffer.POPBuffer.get ( ) `[pure virtual]`

Retrieve a byte from the buffer.

Returns

byte retrieved in the buffer

Implemented in popjava.buffer.BufferRaw, and popjava.buffer.BufferPlugin.

5.48.3.5 Object popjava.buffer.POPBuffer.getArray (Class<?> *arrayType*) throws POPException

Retrieve an array from the buffer.

Parameters

arrayType	Class of the array to retrieve

Returns

Array retrieved in the buffer

Exceptions

POPException	thrown if the serialization process is not going well

Here is the call graph for this function:

Here is the caller graph for this function:

5.48.3.6 abstract boolean popjava.buffer.POPBuffer.getBoolean ( ) `[pure virtual]`

Retrieve a boolean from the buffer.

Returns

boolean retrieved in the buffer

Implemented in popjava.buffer.BufferRaw, and popjava.buffer.BufferPlugin.

Here is the caller graph for this function:

5.48.3.7 abstract boolean [] popjava.buffer.POPBuffer.getBooleanArray (int *length*) [pure virtual]

Retrieve a boolean array from the buffer.

Parameters

length	length of the array to retrieve

Returns

boolean array retrieved in the buffer

Implemented in popjava.buffer.BufferRaw, and popjava.buffer.BufferPlugin.

Here is the caller graph for this function:

5.48.3.8 abstract byte [] popjava.buffer.POPBuffer.getByteArray (int *length*) [pure virtual]

Retrieve a byte array from the buffer.

Parameters

length	length of the array to retrieve

Returns

byte array retrieved in the buffer

Implemented in popjava.buffer.BufferRaw, and popjava.buffer.BufferPlugin.

Here is the caller graph for this function:

5.48.3.9 abstract char popjava.buffer.POPBuffer.getChar ( ) [pure virtual]

Retrieve a char from the buffer.

Returns

char retrieved in the buffer

Implemented in popjava.buffer.BufferRaw, and popjava.buffer.BufferPlugin.

Here is the caller graph for this function:

5.48.3.10 abstract char [] popjava.buffer.POPBuffer.getCharArray (int *length*) [pure virtual]

Retrieve a char array from the buffer.

Parameters

length	length of the array to retrieve

Returns

char array retrieved in the buffer

Implemented in popjava.buffer.BufferRaw, and popjava.buffer.BufferPlugin.

Here is the caller graph for this function:

5.48.3.11 abstract double popjava.buffer.POPBuffer.getDouble ( ) [pure virtual]

Retrieve a double from the buffer.

Returns

double retrieved in the buffer

Implemented in popjava.buffer.BufferRaw, and popjava.buffer.BufferPlugin.

Here is the caller graph for this function:

5.48.3.12 abstract double [] popjava.buffer.POPBuffer.getDoubleArray (int *length*) [pure virtual]

Retrieve a double array from the buffer.

Parameters

length	length of the array to retrieve

Returns

double array retrieved in the buffer

Implemented in popjava.buffer.BufferRaw, and popjava.buffer.BufferPlugin.

Here is the caller graph for this function:

5.48.3.13 abstract float popjava.buffer.POPBuffer.getFloat ( ) [pure virtual]

Retrieve a float from the buffer.

Returns

float retrieved in the buffer

Implemented in popjava.buffer.BufferRaw, and popjava.buffer.BufferPlugin.

Here is the caller graph for this function:

5.48.3.14 abstract float [] popjava.buffer.POPBuffer.getFloatArray (int *length*) [pure virtual]

Retrieve a float array from the buffer.

Parameters

length	length of the array to retrieve

Returns

float array retrieved in the buffer

Implemented in popjava.buffer.BufferRaw, and popjava.buffer.BufferPlugin.

Here is the caller graph for this function:

5.48.3.15 MessageHeader popjava.buffer.POPBuffer.getHeader ( )

Get the message header associated with this buffer.

Returns

Message header associated with the buffer

5.48.3.16 abstract int popjava.buffer.POPBuffer.getInt ( ) [pure virtual]

Retrieve a int from the buffer.

Returns

int retrieved in the buffer

Implemented in popjava.buffer.BufferRaw, and popjava.buffer.BufferPlugin.

Here is the caller graph for this function:

5.48.3.17 abstract int [ ] popjava.buffer.POPBuffer.getIntArray (int *length*) [pure virtual]

Retrieve a int array from the buffer.

Parameters

length	length of the array to retrieve

Returns

int array retrieved in the buffer

Implemented in popjava.buffer.BufferRaw, and popjava.buffer.BufferPlugin.

Here is the caller graph for this function:

5.48.3.18 abstract long popjava.buffer.POPBuffer.getLong ( ) [pure virtual]

Retrieve a long from the buffer.

Returns

long retrieved in the buffer

Implemented in popjava.buffer.BufferRaw, and popjava.buffer.BufferPlugin.

Here is the caller graph for this function:

5.48.3.19 abstract long [ ] popjava.buffer.POPBuffer.getLongArray (int *length*) [pure virtual]

Retrieve a long array from the buffer.

Parameters

length	length of the array to retrieve

Returns

long array retrieved in the buffer

Implemented in popjava.buffer.BufferRaw, and popjava.buffer.BufferPlugin.

Here is the caller graph for this function:

5.48.3.20 abstract short popjava.buffer.POPBuffer.getShort ( ) `[pure virtual]`

Retrieve a short from the buffer.

Returns

short retrieved in the buffer

Implemented in popjava.buffer.BufferRaw, and popjava.buffer.BufferPlugin.

Here is the caller graph for this function:

5.48.3.21 abstract short [ ] popjava.buffer.POPBuffer.getShortArray (int *length*) `[pure virtual]`

Retrieve a short array from the buffer.

Parameters

length	length of the array to retrieve

Returns

short array retrieved in the buffer

Implemented in popjava.buffer.BufferRaw, and popjava.buffer.BufferPlugin.

Here is the caller graph for this function:

5.48.3.22 abstract String popjava.buffer.POPBuffer.getString ( ) `[pure virtual]`

Retrieve a string from the buffer.

Returns

string retrieved in the buffer

Implemented in popjava.buffer.BufferRaw, and popjava.buffer.BufferPlugin.

Here is the caller graph for this function:

5.48.3.23 abstract int popjava.buffer.POPBuffer.getTranslatedInteger (byte[ ] *value*) `[pure virtual]`

Get a integer value of the byte array.

Parameters

value	The byte array to translate

Returns

The integer

Implemented in popjava.buffer.BufferRaw, popjava.buffer.BufferPlugin, and popjava.buffer.BufferXDR.

Here is the caller graph for this function:

5.48.3.24 Object popjava.buffer.POPBuffer.getValue (Class<?> *c*) throws POPException

Retrieve an object from the buffer.

Parameters

c	Class of the object to retrieve

Returns

Object retrieved in the buffer

Exceptions

POPException	thrown if the deserialization process is not going well

Here is the call graph for this function:

Here is the caller graph for this function:

5.48.3.25 abstract int popjava.buffer.POPBuffer.packMessageHeader ( ) `[pure virtual]`

Pack the message header into the buffer.

Returns

number of byte used for the message header

Implemented in popjava.buffer.BufferRaw, and popjava.buffer.BufferPlugin.

Here is the caller graph for this function:

5.48.3.26 abstract void popjava.buffer.POPBuffer.put (byte *value*) `[pure virtual]`

Insert a byte in the buffer.

Parameters

value	byte value to insert

Implemented in popjava.buffer.BufferRaw, and popjava.buffer.BufferPlugin.

Here is the caller graph for this function:

5.48.3.27 abstract void popjava.buffer.POPBuffer.put (byte[ ] *data*) `[pure virtual]`

Insert a byte array into the buffer.

Parameters

data	byte array to insert

Implemented in popjava.buffer.BufferRaw, and popjava.buffer.BufferPlugin.

5.48.3.28 abstract void popjava.buffer.POPBuffer.put (byte[] *data,* int *offset,* int *length*) `[pure virtual]`

Insert a byte array into a specific place in the buffer.

Parameters

data	byte array to insert
offset	offset for insertion
length	length of the array

Implemented in popjava.buffer.BufferRaw, and popjava.buffer.BufferPlugin.

5.48.3.29 void popjava.buffer.POPBuffer.putArray (Object *o*) throws POPException

Insert an array into the buffer.

Parameters

o	Array to be inserted

Exceptions

POPException	thrown if the serialization process is not going well

Here is the call graph for this function:

Here is the caller graph for this function:

5.48.3.30 abstract void popjava.buffer.POPBuffer.putBoolean (boolean *value*) `[pure virtual]`

Insert a boolean in the buffer.

Parameters

value	boolean value to insert

Implemented in popjava.buffer.BufferRaw, popjava.buffer.BufferPlugin, and popjava.buffer.BufferXDR.

Here is the caller graph for this function:

5.48.3.31 abstract void popjava.buffer.POPBuffer.putBooleanArray (boolean[] *value*) `[pure virtual]`

Insert a boolean array into the buffer.

Parameters

value	boolean array to insert

Implemented in popjava.buffer.BufferRaw, and popjava.buffer.BufferPlugin.

Here is the caller graph for this function:

5.48.3.32 abstract void popjava.buffer.POPBuffer.putByteArray (byte[] *value*) [pure virtual]

Insert a byte array into the buffer.

Parameters

value	byte array to insert

Implemented in popjava.buffer.BufferRaw, and popjava.buffer.BufferPlugin.

Here is the caller graph for this function:

5.48.3.33 abstract void popjava.buffer.POPBuffer.putChar (char *value*) [pure virtual]

Insert a char into the buffer.

Parameters

value	char value to insert

Implemented in popjava.buffer.BufferRaw, and popjava.buffer.BufferPlugin.

Here is the caller graph for this function:

5.48.3.34 abstract void popjava.buffer.POPBuffer.putCharArray (char[] *value*) [pure virtual]

Insert a char array into the buffer.

Parameters

value	char array to insert

Implemented in popjava.buffer.BufferRaw, and popjava.buffer.BufferPlugin.

Here is the caller graph for this function:

5.48.3.35 abstract void popjava.buffer.POPBuffer.putDouble (double *value*) [pure virtual]

Insert a double value into the buffer.

Parameters

value	double value to insert

Implemented in popjava.buffer.BufferRaw, and popjava.buffer.BufferPlugin.

Here is the caller graph for this function:

5.48.3.36 abstract void popjava.buffer.POPBuffer.putDoubleArray (double[] *value*) [pure virtual]

Insert a double array into the buffer.

Parameters

value	double array to insert

Implemented in popjava.buffer.BufferRaw, and popjava.buffer.BufferPlugin.

Here is the caller graph for this function:

5.48.3.37 abstract void popjava.buffer.POPBuffer.putFloat (float *value*) [pure virtual]

Insert a float value into the buffer.

Parameters

value	float value to insert

Implemented in popjava.buffer.BufferRaw, and popjava.buffer.BufferPlugin.

Here is the caller graph for this function:

5.48.3.38 abstract void popjava.buffer.POPBuffer.putFloatArray (float[ ] *value*) [pure virtual]

Insert a float array into the buffer.

Parameters

value	float array to insert

Implemented in popjava.buffer.BufferRaw, and popjava.buffer.BufferPlugin.

Here is the caller graph for this function:

5.48.3.39 abstract void popjava.buffer.POPBuffer.putInt (int *value*) [pure virtual]

Insert a int into the buffer.

Parameters

value	int value to insert

Implemented in popjava.buffer.BufferRaw, and popjava.buffer.BufferPlugin.

Here is the caller graph for this function:

5.48.3.40 abstract void popjava.buffer.POPBuffer.putIntArray (int[ ] *value*) [pure virtual]

Insert a int array into the buffer.

Parameters

value	int array to insert

Implemented in popjava.buffer.BufferRaw, and popjava.buffer.BufferPlugin.

Here is the caller graph for this function:

5.48.3.41 abstract void popjava.buffer.POPBuffer.putLong (long *value*) [pure virtual]

Insert a long into the buffer.

Parameters

value	long value to insert

Implemented in popjava.buffer.BufferRaw, and popjava.buffer.BufferPlugin.

Here is the caller graph for this function:

5.48.3.42 abstract void popjava.buffer.POPBuffer.putLongArray (long[] *value*) [pure virtual]

Insert a long array into the buffer.

Parameters

value	long array to insert

Implemented in popjava.buffer.BufferRaw, and popjava.buffer.BufferPlugin.

Here is the caller graph for this function:

5.48.3.43 abstract void popjava.buffer.POPBuffer.putShort (short *value*) [pure virtual]

Insert a short into the buffer.

Parameters

value	short value to insert

Implemented in popjava.buffer.BufferRaw, and popjava.buffer.BufferPlugin.

Here is the caller graph for this function:

5.48.3.44 abstract void popjava.buffer.POPBuffer.putShortArray (short[] *value*) [pure virtual]

Insert a short array into the buffer.

Parameters

value	short array to insert

Implemented in popjava.buffer.BufferRaw, and popjava.buffer.BufferPlugin.

Here is the caller graph for this function:

5.48.3.45 abstract void popjava.buffer.POPBuffer.putString (String *value*) [pure virtual]

Insert a string into the buffer.

Parameters

value	string value to insert

Implemented in popjava.buffer.BufferRaw, and popjava.buffer.BufferPlugin.

Here is the caller graph for this function:

5.48.3.46 void popjava.buffer.POPBuffer.putValue (Object *o*, Class<?> *c*) throws POPException

Insert an object into the buffer.

Parameters

o	Object to be inserted
c	Class of the object to be inserted

Exceptions

POPException	thrown if the serialization process is not going well

Here is the call graph for this function:

Here is the caller graph for this function:

5.48.3.47 void popjava.buffer.POPBuffer.serializeReferenceObject (Class<?> *type*, Object *obj*) throws POPException

Insert an object reference into the buffer.

Parameters

type	Class of the object
obj	Object to be inserted

Exceptions

POPException	thrown if the serialization process is not going well

Here is the call graph for this function:

5.48.3.48 void popjava.buffer.POPBuffer.setHeader (MessageHeader *messageHeader*)

Associate a message header with this buffer.

Parameters

messageHeader	Message header to be associated with this buffer

Here is the call graph for this function:

Here is the caller graph for this function:

5.48.3.49 int popjava.buffer.POPBuffer.size ( )

Get the current size of the buffer.

Returns

current size of the buffer as a int value

Here is the caller graph for this function:

5.48.3.50 String popjava.buffer.POPBuffer.toCharString ( )

Return an empty string.

Returns

empty string

5.48.3.51 String popjava.buffer.POPBuffer.toIntString ( )

Return an empty string.

Returns

empty string

5.49 popjava.annotation.POPClass Interface Reference

Collaboration diagram for popjava.annotation.POPClass:

Public Member Functions

- String **className** () default""
- int **classId** () default-1
- boolean **deconstructor** () default false
- int **maxRequestQueue** () default RequestQueue.DEFAULT_REQUEST_QUEUE_SIZE

5.50 popjava.annotation.processors.POPClassProcessor Class Reference

http://www.javaspecialists.eu/archive/Issue167.html

Inheritance diagram for popjava.annotation.processors.POPClassProcessor:

Collaboration diagram for popjava.annotation.processors.POPClassProcessor:

Public Member Functions

- void **init** (ProcessingEnvironment env)
- boolean **process** (Set<?extends TypeElement > annotations, RoundEnvironment env)

5.50.1 Detailed Description

http://www.javaspecialists.eu/archive/Issue167.html

5.51 popjava.serviceadapter.POPCodeManager Class Reference

Partial POP-Java class implementation to be used with the POP-C++ runtime This class declares the necessary methods to use the CodeMgr parallel object of POP-C++.

Inheritance diagram for popjava.serviceadapter.POPCodeManager:

Collaboration diagram for popjava.serviceadapter.POPCodeManager:

Public Member Functions

- POPCodeManager ()

  Default constructor of POPCodeManager.
- POPCodeManager (String challenge)

  Constructor of POPCodeManager with challenge string.
- void registerCode (String objname, String platform, String codefile)

  Register a executable code file in the CodeMgr service.
- int queryCode (String objname, String platform, POPString codefile)

  Query the CodeMgr to retrieve the code file for a specific object on a specific architecture.

- int getPlatform (String objname, POPString platform)

  Query the CodeMgr to know the platforms of a specific object.

Additional Inherited Members

5.51.1 Detailed Description

Partial POP-Java class implementation to be used with the POP-C++ runtime This class declares the necessary methods to use the CodeMgr parallel object of POP-C++.

5.51.2 Constructor & Destructor Documentation

5.51.2.1 popjava.serviceadapter.POPCodeManager.POPCodeManager ( )

Default constructor of POPCodeManager.

Create a POP-C++ object CodeMgr

Here is the call graph for this function:

5.51.2.2 popjava.serviceadapter.POPCodeManager.POPCodeManager (String *challenge*)

Constructor of POPCodeManager with challenge string.

Parameters

challenge	challenge string to stop the service

5.51.3 Member Function Documentation

5.51.3.1 int popjava.serviceadapter.POPCodeManager.getPlatform (String *objname*, POPString *platform*)

Query the CodeMgr to know the platforms of a specific object.

Parameters

objname	Name of the object
platform	Output argument - platform available for the object

Returns

number of platform available

5.51.3.2 int popjava.serviceadapter.POPCodeManager.queryCode (String *objname*, String *platform*, POPString *codefile*)

Query the CodeMgr to retrieve the code file for a specific object on a specific architecture.

Parameters

objname	Name of the object
platform	Platform desired
codefile	Output argument - code file for the specific object and the specific platform

Returns

0 if the code file is not available

5.51.3.3 void popjava.serviceadapter.POPCodeManager.registerCode (String *objname*, String *platform*, String *codefile*)

Register a executable code file in the CodeMgr service.

Parameters

objname	Name of the parallel object
platform	Platform of the executable
codefile	Path of the executable code file

5.52 popjava.annotation.POPConfig Interface Reference

Collaboration diagram for popjava.annotation.POPConfig:

Classes

- enum Type

Public Member Functions

- Type **value** ()

5.53 popjava.base.POPErrorCode Class Reference

This class regroup all POP error code.

Collaboration diagram for popjava.base.POPErrorCode:

Static Public Attributes

- static int **USER_DEFINE_ERROR** = 10000
- static int **OBJECT_NO_RESOURCE** = USER_DEFINE_ERROR + 1
- static int **OBJECT_BIND_FAIL** = USER_DEFINE_ERROR + 2
- static int **OBJECT_MISMATCH_METHOD** = USER_DEFINE_ERROR + 3
- static int **CODE_SERVICE_FAIL** = USER_DEFINE_ERROR + 4
- static int **ALLOCATION_EXCEPTION** = USER_DEFINE_ERROR + 5
- static int **OBJECT_EXECUTABLE_NOTFOUND** = USER_DEFINE_ERROR + 6
- static int **POP_BUFFER_FORMAT** = USER_DEFINE_ERROR + 7
- static int **POP_APPSERVICE_FAIL** = USER_DEFINE_ERROR + 8
- static final int **POP_JOBSERVICE_FAIL** = 10009
- static final int **POP_EXEC_FAIL** = 10010
- static int **POP_BIND_BAD_REPLY** = USER_DEFINE_ERROR + 11
- static int **POP_NO_PROTOCOL** = USER_DEFINE_ERROR + 12
- static int **POP_NO_ENCODING** = USER_DEFINE_ERROR + 13
- static int **REFLECT_INVOKE_EXCEPTION** = USER_DEFINE_ERROR + 14
- static int **REFLECT_SERIALIZE_EXCEPTION** = USER_DEFINE_ERROR + 15
- static int **REFLECT_METHOD_NOT_FOUND_EXCEPTION** = USER_DEFINE_ERROR + 16

- static int **POP_BUFFER_NOT_AVAILABLE** = USER_DEFINE_ERROR + 16
- static int **POP_COMBOX_NOT_AVAILABLE** = USER_DEFINE_ERROR + 17
- static int **POP_ACCESSPOINT_NOT_AVAILABLE** = USER_DEFINE_ERROR + 18
- static int **NOT_ALLOW_PUT_NULL_OBJECT_TP_BUFFER** = USER_DEFINE_ERROR + 19
- static int **UNKNOWN_EXCEPTION** = USER_DEFINE_ERROR + 20
- static int **USER_DEFINE_LASTERROR** = USER_DEFINE_ERROR + 20

5.53.1 Detailed Description

This class regroup all POP error code.

They are the same as the ones defined in the POP-C++ implementation.

5.54 popjava.base.POPException Class Reference

This class is the base implementation for all POP exception.

Inheritance diagram for popjava.base.POPException:

Collaboration diagram for popjava.base.POPException:

Public Member Functions

- POPException (int errorCode, String errorMessage)

  Create a new POPException with the given value.
- POPException ()

  Create a new empty POPException.
- boolean deserialize (POPBuffer buffer)

  Deserialize an exception from the buffer.
- boolean serialize (POPBuffer buffer)

  Serialize an exception into the buffer.

Static Public Member Functions

- static void throwObjectNoResource () throws POPException

  Method to throw a new exception : No resource found.
- static void throwObjectBindException (POPAccessPoint accessPoint) throws POPException

  Throw an exception when the object binding is not a success.
- static void throwBufferFormatException (Class<?> c) throws POPException

  Throw an exception when the buffer format is not correct.
- static void throwReflectException (String methodName, String errorMessage) throws POPException

  Throw an exception when invoke a serialize method.
- static POPException createReflectException (String methodName, String errorMessage)

  Create an exception when invoke a serialize method.
- static void throwReflectSerializeException (String className, String errorMessage) throws POPException

  Throw an exception when invoke a serialize method.
- static void throwReflectMethodNotFoundException (String className, int methodId, String errorMessage) throws POPException

  Throw an exception when method is not found.
- static POPException createReflectMethodNotFoundException (String className, int methodId, String errorMessage)

  Create an exception when method is not found.

- static POPException throwBufferNotAvailableException () throws POPException

  Throw an exception when the buffer is not available.
- static POPException throwComboxNotAvailableException () throws POPException

  Throw an exception when the combox is not available.
- static POPException throwAccessPointNotAvailableException () throws POPException

  Throw an exception when the access point of an object is not available.
- static POPException throwNullObjectNotAllowException () throws POPException

  Throw an exception when trying to create a null object.

Public Attributes

- int errorCode

  Code of the error in the exception.
- String errorMessage

  Message associated with the exception.

5.54.1 Detailed Description

This class is the base implementation for all POP exception.

5.54.2 Constructor & Destructor Documentation

5.54.2.1 popjava.base.POPException.POPException (int *errorCode*, String *errorMessage*)

Create a new POPException with the given value.

Parameters

errorCode	Code of the error
errorMessage	Assiociated message

5.54.3 Member Function Documentation

5.54.3.1 static **POPException** popjava.base.POPException.createReflectException (String *methodName*, String *errorMessage*) `[static]`

Create an exception when invoke a serialize method.

Parameters

methodName	Name of the method
errorMessage	Message

Returns

the exception

Here is the call graph for this function:

Here is the caller graph for this function:

5.54.3.2 static **POPException** popjava.base.POPException.createReflectMethodNotFoundException (String *className,* int *methodId,* String *errorMessage*) [static]

Create an exception when method is not found.

Returns

the exception

Here is the call graph for this function:

Here is the caller graph for this function:

5.54.3.3 boolean popjava.base.POPException.deserialize (**POPBuffer** *buffer*)

Deserialize an exception from the buffer.

Parameters

buffer	The buffer to deserialize from

Implements popjava.dataswaper.IPOPBase.

Here is the call graph for this function:

Here is the caller graph for this function:

5.54.3.4 boolean popjava.base.POPException.serialize (**POPBuffer** *buffer*)

Serialize an exception into the buffer.

Parameters

buffer	The buffer to serialize in

Implements popjava.dataswaper.IPOPBase.

Here is the call graph for this function:

Here is the caller graph for this function:

5.54.3.5 static **POPException** popjava.base.POPException.throwAccessPointNotAvailableException ( ) throws **POPException** [static]

Throw an exception when the access point of an object is not available.

Returns

the exception

Exceptions

POPException	exception thrown by this method

Here is the call graph for this function:

Here is the caller graph for this function:

5.54.3.6 static void popjava.base.POPException.throwBufferFormatException (Class< ? > *c*) throws POPException [static]

Throw an exception when the buffer format is not correct.

Parameters

c	Class

Exceptions

POPException	exception thrown by this method

Here is the call graph for this function:

5.54.3.7 static POPException popjava.base.POPException.throwBufferNotAvailableException ( ) throws POPException [static]

Throw an exception when the buffer is not available.

Returns

the exception

Exceptions

POPException	exception thrown by this method

Here is the call graph for this function:

5.54.3.8 static POPException popjava.base.POPException.throwComboxNotAvailableException ( ) throws POPException [static]

Throw an exception when the combox is not available.

Returns

the exception

Exceptions

POPException	exception thrown by this method

Here is the call graph for this function:

5.54.3.9 static POPException popjava.base.POPException.throwNullObjectNotAllowException ( ) throws POPException [static]

Throw an exception when trying to create a null object.

Returns

the exception

Exceptions

POPException	exception thrown by this method

Here is the call graph for this function:

Here is the caller graph for this function:

5.54.3.10 static void popjava.base.POPException.throwObjectBindException (POPAccessPoint *accessPoint*) throws POPException [static]

Throw an exception when the object binding is not a success.

Parameters

accessPoint	Access point of the object

Exceptions

POPException	exception thrown by this method

Here is the call graph for this function:

Here is the caller graph for this function:

5.54.3.11 static void popjava.base.POPException.throwObjectNoResource ( ) throws POPException [static]

Method to throw a new exception : No resource found.

Exceptions

POPException	exception thrown by this method

Here is the call graph for this function:

5.54.3.12 static void popjava.base.POPException.throwReflectException (String *methodName*, String *errorMessage*) throws POPException [static]

Throw an exception when invoke a serialize method.

Exceptions

POPException	exception thrown by this method

Here is the call graph for this function:

5.54.3.13 static void popjava.base.POPException.throwReflectMethodNotFoundException (String *className*, int *methodId*, String *errorMessage*) throws POPException [static]

Throw an exception when method is not found.

Exceptions

POPException	exception thrown by this method

Here is the call graph for this function:

5.54.3.14 static void popjava.base.POPException.throwReflectSerializeException (String *className,* String *errorMessage*) throws POPException [static]

Throw an exception when invoke a serialize method.

Exceptions

POPException	exception thrown by this method

Here is the call graph for this function:

Here is the caller graph for this function:

5.55 popjava.PopJava Class Reference

This class is used to create parallel object.

Collaboration diagram for popjava.PopJava:

Public Member Functions

- PopJava ()

    Creates a new instance of PopJava.

Static Public Member Functions

- static< T > T newActive (Class< T > targetClass, ObjectDescription objectDescription, Object...argvs) throws POPException

    Static method used to create a new parallel object by passing an object description.
- static< T > T newActive (Class< T > targetClass, Object...argvs) throws POPException

    Static method used to create a new parallel object.
- static< T > T newActive (Class< T > targetClass, POPAccessPoint accessPoint) throws POPException

    Static method used to create a parallel object from an existing access point.
- static< T > T newActiveFromBuffer (Class< T > targetClass, POPBuffer buffer) throws POPException

    Static method used to create a parallel object from the buffer.

5.55.1 Detailed Description

This class is used to create parallel object.

All the methods from this class are static so no instantiation is needed.

5.55.2 Member Function Documentation

5.55.2.1 static <T> T popjava.PopJava.newActive (Class< T > *targetClass,* ObjectDescription *objectDescription,* Object... *argvs*) throws POPException [static]

Static method used to create a new parallel object by passing an object description.

Parameters

targetClass	the parallel class to be created
object-Description	the object description for the resource requirements
argvs	arguments of the constructor (may be empty)

Returns

references to the parallel object created

Exceptions

POPException	

Here is the call graph for this function:

Here is the caller graph for this function:

5.55.2.2 static <T> T popjava.PopJava.newActive (Class< T > *targetClass,* Object... *argvs*) throws POPException [static]

Static method used to create a new parallel object.

Parameters

targetClass	the parallel class to be created
argvs	arguments of the constructor (may be empty)

Returns

references to the parallel object created

Exceptions

POPException	

Here is the call graph for this function:

5.55.2.3 static <T> T popjava.PopJava.newActive (Class< T > *targetClass,* POPAccessPoint *accessPoint*) throws POPException [static]

Static method used to create a parallel object from an existing access point.

Parameters

targetClass	the parallel class to be created
accessPoint	access point of the living object

Returns

references to the parallel object

Exceptions

POPException	

Here is the call graph for this function:

5.55.2.4 static <T> T popjava.PopJava.newActiveFromBuffer (Class< T > *targetClass*, POPBuffer *buffer*) throws POPException `[static]`

Static method used to create a parallel object from the buffer.

Parameters

targetClass	the parallel class to be recovered
buffer	buffer from which the object must be recovered

Returns

references to the parallel object

Exceptions

POPException	

Here is the call graph for this function:

Here is the caller graph for this function:

5.56 popjava.codemanager.POPJavaAppService Class Reference

Inheritance diagram for popjava.codemanager.POPJavaAppService:

Collaboration diagram for popjava.codemanager.POPJavaAppService:

Public Member Functions

- void registerCode (String objname, String platform, String codefile)
  Register a executable code file in the CodeMgr service.
- int queryCode (String objname, String platform, POPString codefile)
  Query the CodeMgr to retrieve the code file for a specific object on a specific architecture.
- String **getLocalJavaFileLocation** (String objname)
- int getPlatform (String objname, POPString platform)
  Query the CodeMgr to know the platforms of a specific object.
- String **getPOPCAppID** ()

Additional Inherited Members

5.56.1 Member Function Documentation

5.56.1.1 int popjava.codemanager.POPJavaAppService.getPlatform (String *objname*, POPString *platform*)

Query the CodeMgr to know the platforms of a specific object.

Parameters

objname	Name of the object
platform	Output argument - platform available for the object

Returns

number of platform available

Implements popjava.codemanager.AppService.

5.56.1.2 int popjava.codemanager.POPJavaAppService.queryCode (String *objname*, String *platform*, POPString *codefile*)

Query the CodeMgr to retrieve the code file for a specific object on a specific architecture.

Parameters

objname	Name of the object
platform	Platform desired
codefile	Output argument - code file for the specific object and the specific platform

Returns

0 if the code file is not available

Implements popjava.codemanager.AppService.

Here is the call graph for this function:

5.56.1.3 void popjava.codemanager.POPJavaAppService.registerCode (String *objname*, String *platform*, String *codefile*)

Register a executable code file in the CodeMgr service.

Parameters

objname	Name of the parallel object
platform	Platform of the executable
codefile	Path of the executable code file

Implements popjava.codemanager.AppService.

5.57 popjava.scripts.Popjavac Class Reference

Collaboration diagram for popjava.scripts.Popjavac:

Static Public Member Functions

- static void **main** (String[] args)

5.58 popjava.system.POPJavaConfiguration Class Reference

Collaboration diagram for popjava.system.POPJavaConfiguration:

Static Public Member Functions

- static String **getBrokerCommand** ()
- static String getPopAppCoreService ()

Retrieve the POP-C++ AppCoreService executable location.
- static String getPopJavaLocation ()

  Retrieve the POP-Java installation location.
- static String getPopPluginLocation ()

  Retrieve the POP-Java plugin location.
- static String **getPOPJavaCodePath** ()
- static String **getPopJavaJar** ()

5.58.1 Member Function Documentation

5.58.1.1 static String popjava.system.POPJavaConfiguration.getPopAppCoreService ( ) [static]

Retrieve the POP-C++ AppCoreService executable location.

Returns

string value of the POP-C++ AppCoreService executable location

Here is the caller graph for this function:

5.58.1.2 static String popjava.system.POPJavaConfiguration.getPopJavaLocation ( ) [static]

Retrieve the POP-Java installation location.

Returns

string value of the POP-java location

Here is the caller graph for this function:

5.58.1.3 static String popjava.system.POPJavaConfiguration.getPopPluginLocation ( ) [static]

Retrieve the POP-Java plugin location.

Returns

string value of the POP-Java plugin location

Here is the caller graph for this function:

5.59 popjava.serviceadapter.POPJobManager Class Reference

Partial POP-Java class implementation to be used with the POP-C++ runtime This class declares the necessary methods to use the JobMgr parallel object of POP-C++.

Inheritance diagram for popjava.serviceadapter.POPJobManager:

Collaboration diagram for popjava.serviceadapter.POPJobManager:

Public Member Functions

- POPJobManager ()

  Default constructor of POPJobManager.
- POPJobManager (boolean daemon, String challenge, String url)

Constructor of POPJobManager with challenge string.

- POPJobManager (boolean daemon, String config, String challenge, String url)

  Constructor of POPCodeManager with challenge string.
- void registerNode (String url)

  Register a other JobMgr as a neighbor.
- int query (POPString type, POPString value)

  Query configuration informations.
- int createObject (POPAccessPoint localservice, POPString objname, ObjectDescriptionInput od, int howmany, POPAccessPoint[] objcontacts, int howmany2, POPAccessPoint[] remotejobcontacts)

  Ask the JobMgr service to create a new parallel object.
- boolean allocResource (String localservice, String objname, ObjectDescriptionInput od, int howmany, float[] fitness, POPAccessPoint[] jobcontacts, int[] reserveIDs, int[] requestInfo, int[] trace, int tracesize)

  Ask the JobMgr service to allocate resources for a new objects.
- void cancelReservation (int[] req, int howmany)

  Ask the JobMgr service to cancel some reservation for parallel object.
- int execObj (POPString objname, int howmany, int[] reserveIDs, String localservice, POPAccessPoint[] objcontacts)

  Ask the JobMgr service to execute a specific object.
- void **dump** ()
- void start ()

  Start the JobMgr service.
- void selfRegister ()

  Register the local JobMgr service to its known neighbors.

Static Public Attributes

- static final int DEFAULT_PORT = 2711

  Default running port of the JobMgr service.

Additional Inherited Members

5.59.1 Detailed Description

Partial POP-Java class implementation to be used with the POP-C++ runtime This class declares the necessary methods to use the JobMgr parallel object of POP-C++.

5.59.2 Constructor & Destructor Documentation

5.59.2.1 popjava.serviceadapter.POPJobManager.POPJobManager ( )

Default constructor of POPJobManager.

Create a POP-C++ object JobMgr

Here is the call graph for this function:

5.59.2.2 popjava.serviceadapter.POPJobManager.POPJobManager (boolean *daemon*, String *challenge*, String *url*)

Constructor of POPJobManager with challenge string.

Parameters

daemon	Set the service in deamon mode
challenge	Challenge string needed for the service stop
url	URL of the JobMgr service

5.59.2.3 popjava.serviceadapter.POPJobManager.POPJobManager (boolean *daemon,* String *config,* String *challenge,* String *url*)

Constructor of POPCodeManager with challenge string.

Parameters

daemon	Set the service in deamon mode
config	Configuration information
challenge	Challenge string needed for the service stop
url	URL of the JobMgr service

5.59.3 Member Function Documentation

5.59.3.1 boolean popjava.serviceadapter.POPJobManager.allocResource (String *localservice,* String *objname,* **ObjectDescriptionInput** *od,* int *howmany,* float[ ] *fitness,* **POPAccessPoint**[ ] *jobcontacts,* int[ ] *reserveIDs,* int[ ] *requestInfo,* int[ ] *trace,* int *tracesize*)

Ask the JobMgr service to allocate resources for a new objects.

Parameters

localservice	Access to the local application scope services
objname	Name of the object to create
od	Object description for the resource requirements of this object
howmany	Number of objects to create
fitness	Fitness of the resource
jobcontacts	Output arguments - contacts to the JobMgr to create objects
reserveIDs	Output arguments - reservation identifier for each objects
requestInfo	
trace	
tracesize	

Returns

true if the runtime has allocated some resources for the parallel objects

5.59.3.2 void popjava.serviceadapter.POPJobManager.cancelReservation (int[ ] *req,* int *howmany*)

Ask the JobMgr service to cancel some reservation for parallel object.

Parameters

req	Reservation identifiers of the reservations to cancel
howmany	Number of reservations to cancel

5.59.3.3 int popjava.serviceadapter.POPJobManager.createObject (**POPAccessPoint** *localservice,* **POPString** *objname,* **ObjectDescriptionInput** *od,* int *howmany,* **POPAccessPoint**[ ] *objcontacts,* int *howmany2,* **POPAccessPoint**[ ] *remotejobcontacts*)

Ask the JobMgr service to create a new parallel object.

Parameters

localservice	Access to the local application scope services
objname	Name of the object to create
od	Object description for the resource requirements of this object
howmany	Number of objects to create
jobcontacts	Output arguments - contacts to the objects created

Returns

0 if the object is created correctly

5.59.3.4 int popjava.serviceadapter.POPJobManager.execObj (POPString *objname*, int *howmany*, int[ ] *reserveIDs*, String *localservice*, POPAccessPoint[ ] *objcontacts*)

Ask the JobMgr service to execute a specific object.

Parameters

objname	Name of the object
howmany	Number of object to execute
reserveIDs	Reservations identifiers for these objects
localservice	Access to the local application scope services
objcontacts	Output arguments - contacts to the objects created

Returns

0 if the execution hasn't failed

5.59.3.5 int popjava.serviceadapter.POPJobManager.query (POPString *type*, POPString *value*)

Query configuration informations.

Parameters

type	Name of the configuration element
value	Output argument - Value of the configuration element

Returns

0 if the configuration element is not found

5.59.3.6 void popjava.serviceadapter.POPJobManager.registerNode (String *url*)

Register a other JobMgr as a neighbor.

Parameters

url	URL of the node to register

5.60 popjava.serviceadapter.POPJobService Class Reference

Partial POP-Java class implementation to be used with the POP-C++ runtime This class declares the necessary methods to use the JobMgr parallel object of POP-C++.

Inheritance diagram for popjava.serviceadapter.POPJobService:

Collaboration diagram for popjava.serviceadapter.POPJobService:

Public Member Functions

- POPJobService ()

  Default constructor of POPJobService.
- POPJobService (String challenge)

  Constructor of POPAppService with parameters.
- int createObject (POPAccessPoint localservice, String objname, ObjectDescriptionInput od, int howmany, POPAccessPoint[ ] objcontacts, int howmany2, POPAccessPoint[ ] remotejobcontacts)

  Ask the JobCoreService service to create a new parallel object.

Additional Inherited Members

5.60.1 Detailed Description

Partial POP-Java class implementation to be used with the POP-C++ runtime This class declares the necessary methods to use the JobMgr parallel object of POP-C++.

5.60.2 Constructor & Destructor Documentation

5.60.2.1 popjava.serviceadapter.POPJobService.POPJobService ( )

Default constructor of POPJobService.

Create a POP-C++ object JobCoreService

Here is the call graph for this function:

5.60.2.2 popjava.serviceadapter.POPJobService.POPJobService (String *challenge*)

Constructor of POPAppService with parameters.

Parameters

challenge	challenge string to stop the parallel object

5.60.3 Member Function Documentation

5.60.3.1 int popjava.serviceadapter.POPJobService.createObject (POPAccessPoint *localservice,* String *objname,* ObjectDescriptionInput *od,* int *howmany,* POPAccessPoint[ ] *objcontacts,* int *howmany2,* POPAccessPoint[ ] *remotejobcontacts*)

Ask the JobCoreService service to create a new parallel object.

Parameters

localservice	Access to the local application scope services
objname	Name of the object to create
od	Object description for the resource requirements of this object
howmany	Number of objects to create
objcontacts	Output arguments - contacts to the objects created

Returns

0 if the object is created correctly

Here is the caller graph for this function:

5.61 popjava.scripts.Popjrun Class Reference

Collaboration diagram for popjava.scripts.Popjrun:

Static Public Member Functions

- static void **main** (String[] args)

5.62 popjava.base.POPObject Class Reference

This class is the base class of all POP-Java parallel classes.

Inheritance diagram for popjava.base.POPObject:

Collaboration diagram for popjava.base.POPObject:

Public Member Functions

- POPObject ()

  Creates a new instance of POPObject.
- void **loadPOPAnnotations** (Constructor<?> constructor, Object...argvs)
- boolean isDaemon ()

  Specify if the parallel object is running like a deamon.
- final boolean canKill ()

  Ask if the object can be killed.
- final ObjectDescription getOd ()

  Get the object description of the POPObject.
- final void setOd (ObjectDescription od)

  Set a new object description to the POPObject.
- POPAccessPoint getAccessPoint ()

  Retrieve the access point of the parallel object.
- final String getClassName ()

  Retrieve the class name of the parallel object.
- final int getClassId ()

  Get the class unique identifier.
- Method getMethodByInfo (MethodInfo info) throws NoSuchMethodException

  Retrieve a specific method in the parallel class with some information.
- Constructor<?> getConstructorByInfo (MethodInfo info) throws NoSuchMethodException

Retrieve a constructor by its informations.

- MethodInfo getMethodInfo (Method method)

  Retrieve a method by its informations.
- MethodInfo getMethodInfo (Constructor<?> constructor)

  Retrieve a specific method by its constructor informations.
- int getSemantic (MethodInfo methodInfo)

  Retrieve the invocation semantic of a specific method.
- int getSemantic (Method method)

  Retrieve the invocation semantic of a specific method.
- final void addSemantic (Class<?> c, String methodName, int semantic)

  Set an invocation semantic to a specific method.
- final void addSemantic (Class<?> c, String methodName, int semantic, Class<?>...parameterTypes) throws java.lang.NoSuchMethodException

  Set an invocation semantic to a specific method that is overloaded.
- boolean deserialize (POPBuffer buffer)

  Deserialize the object from the buffer.
- boolean serialize (POPBuffer buffer)

  Serialize the object into the buffer.
- void exit ()

  Exit method.
- void printMethodInfo ()

  Print object information on the standard output.
- String getPOPCReference ()

  Return the reference of this object with a POP-C++ format.
- boolean **isTemporary** ()
- void **makeTemporary** ()

Protected Member Functions

- final void initializePOPObject ()

  Initialize the method identifiers of a POPObject.
- final void setClassName (String className)

  Set the class name.
- final boolean hasDestructor ()

  Return the value of the hasDestrcutor variable.
- final void hasDestructor (boolean hasDestructor)

  Set the destructor value.
- final void setClassId (int classId)

  Set the class unique identifier.
- void initializeMethodInfo (Class<?> c, int startIndex)

  Initialize the method identifier for all the methods in a class.
- int initializeConstructorInfo (Class<?> c, int startIndex)

  Initialize the constructor identifier and the semantic.
- void defineMethod (Class<?>c, String methodName, int methodId, int semanticId, Class<?>...paramTypes)

  Define informations about a method.
- void defineConstructor (Class<?>c, int constructorId, Class<?>...paramTypes)

  Define information about a constructor.
- void finalize ()

  Method called before the object destruction.

Protected Attributes

- int **refCount**
- boolean **generateClassId** = true
- boolean **definedMethodId** =false
- ObjectDescription **od** = new ObjectDescription()

5.62.1 Detailed Description

This class is the base class of all POP-Java parallel classes.

Every POP-Java parallel classes must inherit from this one.

5.62.2 Member Function Documentation

5.62.2.1 final void popjava.base.POPObject.addSemantic (Class<?> *c*, String *methodName*, int *semantic*)

Set an invocation semantic to a specific method.

Parameters

c	class of the method
methodName	method to modify
semantic	semantic to set on the method

Here is the call graph for this function:

5.62.2.2 final void popjava.base.POPObject.addSemantic (Class<?> *c*, String *methodName*, int *semantic*, Class<?>... *parameterTypes*) throws java.lang.NoSuchMethodException

Set an invocation semantic to a specific method that is overloaded.

Parameters

c	class of the method
methodName	method to modify
semantic	semantic to set on the method
parameterTypes	parameters types of the method

Exceptions

java.lang.NoSuchMethod-Exception	

Here is the call graph for this function:

5.62.2.3 final boolean popjava.base.POPObject.canKill ( )

Ask if the object can be killed.

Returns

true if the object can be killed

Here is the caller graph for this function:

5.62.2.4 void popjava.base.POPObject.defineConstructor (Class<?> *c,* int *constructorId,* Class<?>... *paramTypes*) [protected]

Define information about a constructor.

Parameters

c	Class of the constructor
constructorId	Unique identifier of the constructor
paramTypes	Parameters of the constructor

Here is the call graph for this function:

Here is the caller graph for this function:

5.62.2.5 void popjava.base.POPObject.defineMethod (Class<?> *c,* String *methodName,* int *methodId,* int *semanticId,* Class<?>... *paramTypes*) [protected]

Define informations about a method.

Parameters

c	Class of the method
methodName	Name of the method
methodId	Unique identifier of the method
semanticId	Semantic applied to the method
paramTypes	Parameters of the method

Here is the call graph for this function:

Here is the caller graph for this function:

5.62.2.6 boolean popjava.base.POPObject.deserialize (POPBuffer *buffer*)

Deserialize the object from the buffer.

Parameters

buffer	The buffer to deserialize from

Implements popjava.dataswaper.IPOPBase.

Here is the caller graph for this function:

5.62.2.7 POPAccessPoint popjava.base.POPObject.getAccessPoint ( )

Retrieve the access point of the parallel object.

Returns

POPAccessPoint object containing all access points to the parallel object

Here is the call graph for this function:

Here is the caller graph for this function:

5.62.2.8 final int popjava.base.POPObject.getClassId ( )

Get the class unique identifier.

Returns

the class unique identifier

Here is the caller graph for this function:

5.62.2.9 final String popjava.base.POPObject.getClassName ( )

Retrieve the class name of the parallel object.

Returns

class name as a String value

Here is the caller graph for this function:

5.62.2.10 Constructor<?> popjava.base.POPObject.getConstructorByInfo (MethodInfo *info*) throws NoSuchMethodException

Retrieve a constructor by its informations.

Parameters

info	Informations about the constructor to retrieve

Returns

The constructor found

Exceptions

NoSuchMethodException	thrown if no constrcutor is found

Here is the call graph for this function:

5.62.2.11 Method popjava.base.POPObject.getMethodByInfo (MethodInfo *info*) throws NoSuchMethodException

Retrieve a specific method in the parallel class with some information.

Parameters

info	informations about the method to retrieve

Returns

A method object that represent the method found in the parallel class

Exceptions

NoSuchMethodException	thrown is the method is not found

Here is the call graph for this function:

5.62.2.12 **MethodInfo** popjava.base.POPObject.getMethodInfo (Method *method*)

Retrieve a method by its informations.

Parameters

method	Informations about the method to retrieve

Returns

The method found

Here is the call graph for this function:

Here is the caller graph for this function:

5.62.2.13 **MethodInfo** popjava.base.POPObject.getMethodInfo (Constructor< ?> *constructor*)

Retrieve a specific method by its constructor informations.

Parameters

constructor	Informations about the constrcutor

Returns

The method found

5.62.2.14 final **ObjectDescription** popjava.base.POPObject.getOd ( )

Get the object description of the POPObject.

Returns

the object description of the POPObject

Here is the caller graph for this function:

5.62.2.15 String popjava.base.POPObject.getPOPCReference ( )

Return the reference of this object with a POP-C++ format.

Returns

access point of the object as a formatted string

Here is the call graph for this function:

5.62.2.16 int popjava.base.POPObject.getSemantic (**MethodInfo** *methodInfo*)

Retrieve the invocation semantic of a specific method.

Parameters

methodInfo	informations about the specific method

Returns

int value representing the semantics of the method

Here is the caller graph for this function:

5.62.2.17 int popjava.base.POPObject.getSemantic (Method *method*)

Retrieve the invocation semantic of a specific method.

Parameters

method	method to look at

Returns

int value representing the semantics of the method

Here is the call graph for this function:

5.62.2.18 final boolean popjava.base.POPObject.hasDestructor ( ) [protected]

Return the value of the hasDestrcutor variable.

Returns

true if the parclass has a destrcutor

Here is the caller graph for this function:

5.62.2.19 final void popjava.base.POPObject.hasDestructor (boolean *hasDestructor*) [protected]

Set the destructor value.

Must be set to true if the parclass has a destructor

Parameters

hasDestructor	set to true if the parclass has a destructor

Here is the call graph for this function:

5.62.2.20 int popjava.base.POPObject.initializeConstructorInfo (Class<?> *c*, int *startIndex*) [protected]

Initialize the constructor identifier and the semantic.

Parameters

c	class to initialize
startIndex	index of the first constructor

Returns

next index to be used for the methods

Here is the call graph for this function:

Here is the caller graph for this function:

5.62.2.21 void popjava.base.POPObject.initializeMethodInfo (Class<?> *c*, int *startIndex*) [protected]

Initialize the method identifier for all the methods in a class.

Parameters

c	class to initialize
startIndex	index of the first method

Here is the call graph for this function:

Here is the caller graph for this function:

5.62.2.22 final void popjava.base.POPObject.initializePOPObject ( ) [protected]

Initialize the method identifiers of a POPObject.

Parameters

c	the class to initialize

Here is the call graph for this function:

Here is the caller graph for this function:

5.62.2.23 boolean popjava.base.POPObject.isDaemon ( )

Specify if the parallel object is running like a deamon.

Returns

true if it's a deamon

Here is the caller graph for this function:

5.62.2.24 boolean popjava.base.POPObject.serialize (POPBuffer *buffer*)

Serialize the object into the buffer.

Parameters

buffer	The buffer to serialize in

Implements popjava.dataswaper.IPOPBase.

5.62.2.25 final void popjava.base.POPObject.setClassId (int *classId*) [protected]

Set the class unique identifier.

Parameters

classId	the class unique identifier

Here is the caller graph for this function:

5.62.2.26 final void popjava.base.POPObject.setClassName (String *className*) [protected]

Set the class name.

Parameters

className	the class name

Here is the caller graph for this function:

5.62.2.27 final void popjava.base.POPObject.setOd (ObjectDescription *od*)

Set a new object description to the POPObject.

Parameters

od	the new object description

5.63 popjava.annotation.POPObjectDescription Interface Reference

Collaboration diagram for popjava.annotation.POPObjectDescription:

Public Member Functions

- String **url** () default""
- String jvmParameters () default""

    JVM parameters to be used when creating this object.

5.63.1 Member Function Documentation

5.63.1.1 String popjava.annotation.POPObjectDescription.jvmParameters ( )

JVM parameters to be used when creating this object.

Returns

5.64 popjava.serviceadapter.POPObjectMonitor Class Reference

Partial POP-Java class implementation to be used with the POP-C++ runtime This class declares the necessary methods to use the ObjectMonitor parallel object of POP-C++.

Inheritance diagram for popjava.serviceadapter.POPObjectMonitor:

Collaboration diagram for popjava.serviceadapter.POPObjectMonitor:

Public Member Functions

- POPObjectMonitor ()

    Default constructor of POPJobManager.
- POPObjectMonitor (String challenge)

    Constructor of POPAppService with parameters.

- void killAll ()

  Ask the ObjectMonitor service to kill all parallel object.
- void manageObject (String p)

  Ask the ObjectMinotr service to manage a new object.
- void unManageObject (String p)

  Ask the ObjectMinotr service to stop the management of an object.
- int checkObjects ()

  Check how many parallel objects are currently alive.

Additional Inherited Members

5.64.1 Detailed Description

Partial POP-Java class implementation to be used with the POP-C++ runtime This class declares the necessary methods to use the ObjectMonitor parallel object of POP-C++.

5.64.2 Constructor & Destructor Documentation

5.64.2.1 popjava.serviceadapter.POPObjectMonitor.POPObjectMonitor ( )

Default constructor of POPJobManager.

Create a POP-C++ object JobMgr

Here is the call graph for this function:

5.64.2.2 popjava.serviceadapter.POPObjectMonitor.POPObjectMonitor (String *challenge*)

Constructor of POPAppService with parameters.

Parameters

challenge	challenge string to stop the parallel object

5.64.3 Member Function Documentation

5.64.3.1 int popjava.serviceadapter.POPObjectMonitor.checkObjects ( )

Check how many parallel objects are currently alive.

Returns

Number of currently alive parallel objects

5.64.3.2 void popjava.serviceadapter.POPObjectMonitor.manageObject (String *p*)

Ask the ObjectMinotr service to manage a new object.

Parameters

p	acces point to this object

5.64.3.3 void popjava.serviceadapter.POPObjectMonitor.unManageObject (String *p*)

Ask the ObjectMinotr service to stop the management of an object.

Parameters

p	acces point to this object

5.65 popjava.annotation.POPParameter Interface Reference

Collaboration diagram for popjava.annotation.POPParameter:

Classes

- enum Direction

Public Member Functions

- Direction **value** ()

5.66 popjava.baseobject.POPReference Class Reference

This class defined a POPReference.

Collaboration diagram for popjava.baseobject.POPReference:

Public Member Functions

- POPReference ()

  Create a new POPReference.
- void setAccessPoint (POPAccessPoint ap)

  Set the access points.
- void **getReferenceForPOPCInteraction** ()
- void **setAp** (POPAccessPoint ap)
- POPAccessPoint **getAp** ()

5.66.1 Detailed Description

This class defined a POPReference.

As parallel object are not executed on the same machine, the reference of a parallel object is its access points.

5.66.2 Member Function Documentation

5.66.2.1 void popjava.baseobject.POPReference.setAccessPoint (POPAccessPoint *ap*)

Set the access points.

Parameters

ap	Access points to be set

5.67 popjava.serviceadapter.POPRemoteLog Class Reference

Partial POP-Java class implementation to be used with the POP-C++ runtime This class declares the necessary methods to use the RemoteLog parallel object of POP-C++.

Inheritance diagram for popjava.serviceadapter.POPRemoteLog:

Collaboration diagram for popjava.serviceadapter.POPRemoteLog:

Public Member Functions

- POPRemoteLog ()
  Default constructor of POPRemoteLog.
- POPRemoteLog (String challange)
  Constructor of POPAppService with parameters.
- void log (String info)
  Write a remote log.
- void **logPJ** (String appID, String info)

Additional Inherited Members

5.67.1 Detailed Description

Partial POP-Java class implementation to be used with the POP-C++ runtime This class declares the necessary methods to use the RemoteLog parallel object of POP-C++.

5.67.2 Constructor & Destructor Documentation

5.67.2.1 popjava.serviceadapter.POPRemoteLog.POPRemoteLog ( )

Default constructor of POPRemoteLog.

Create a POP-C++ object RomoteLog

Here is the call graph for this function:

5.67.2.2 popjava.serviceadapter.POPRemoteLog.POPRemoteLog (String *challange*)

Constructor of POPAppService with parameters.

Parameters

challange	Challenge string to stop the service

5.67.3 Member Function Documentation

5.67.3.1 void popjava.serviceadapter.POPRemoteLog.log (String *info*)

Write a remote log.

Parameters

info	Information to be written into the remote log file

5.68 popjava.system.POPRemoteLogThread Class Reference

Inheritance diagram for popjava.system.POPRemoteLogThread:

Collaboration diagram for popjava.system.POPRemoteLogThread:

Public Member Functions

- POPRemoteLogThread (String appID)

  POPRemoteLogThread constructor.
- String getFilename ()

  Get the file name used for the remote logging.
- void setRunning (boolean value)

  Set the boolean value used to run or stop the thread.
- void run ()

  Running method of the thread.

5.68.1 Detailed Description

Author

Valentin Clement This thread is responsible to handle the remote log service provided by POP-C++

5.68.2 Constructor & Destructor Documentation

5.68.2.1 popjava.system.POPRemoteLogThread.POPRemoteLogThread (String *appID*)

POPRemoteLogThread constructor.

Parameters

appID	POP Application ID

5.68.3 Member Function Documentation

5.68.3.1 String popjava.system.POPRemoteLogThread.getFilename ( )

Get the file name used for the remote logging.

Returns

File name as a string

5.68.3.2 void popjava.system.POPRemoteLogThread.run ( )

Running method of the thread.

The thread will work in this method until it is stopped

5.68.3.3 void popjava.system.POPRemoteLogThread.setRunning (boolean *value*)

Set the boolean value used to run or stop the thread.

Parameters

value	Boolean value (false will stop the thread)

5.69 popjava.serviceadapter.POPServiceBase Class Reference

Partial POP-Java class implementation to be used with the POP-C++ runtime This class declares the necessary methods to use the paroc_service_base parallel object of POP-C++.

Inheritance diagram for popjava.serviceadapter.POPServiceBase:

Collaboration diagram for popjava.serviceadapter.POPServiceBase:

Public Member Functions

- POPServiceBase ()

  Default constructor of POPCodeManager.
- POPServiceBase (String challenge)

  Constructor of POPServiceBase with parameters.
- void start ()

  Start the service.
- void start (String challenge)

  Start the service with a challenge string for the stop.
- void stop (String challenge)

  Stop the service by giving a challenge string.

Additional Inherited Members

5.69.1 Detailed Description

Partial POP-Java class implementation to be used with the POP-C++ runtime This class declares the necessary methods to use the paroc_service_base parallel object of POP-C++.

5.69.2 Constructor & Destructor Documentation

5.69.2.1 popjava.serviceadapter.POPServiceBase.POPServiceBase ( )

Default constructor of POPCodeManager.

Create a POP-C++ object CodeMgr

Here is the call graph for this function:

5.69.2.2 popjava.serviceadapter.POPServiceBase.POPServiceBase (String *challenge*)

Constructor of POPServiceBase with parameters.

Parameters

challenge	challenge string to stop the parallel object

5.69.3 Member Function Documentation

5.69.3.1 void popjava.serviceadapter.POPServiceBase.start (String *challenge*)

Start the service with a challenge string for the stop.

Parameters

challenge	Challenge string needed for the service stop

5.69.3.2 void popjava.serviceadapter.POPServiceBase.stop (String *challenge*)

Stop the service by giving a challenge string.

Parameters

challenge	Challenge string needed for the service stop

5.70 popjava.dataswaper.POPString Class Reference

Compatible with the POP-C++ paroc_string implementation.

Inheritance diagram for popjava.dataswaper.POPString:

Collaboration diagram for popjava.dataswaper.POPString:

Public Member Functions

- POPString ()
  Default constructor.
- POPString (String value)
  Constructor with given value.
- void setValue (String value)
  Set the string value of this object.
- String getValue ()
  Get the current value of this object.
- boolean serialize (POPBuffer buffer)
  Serialize the POPString into the buffer.
- boolean deserialize (POPBuffer buffer)
  Deserilize the POPString from the buffer.
- String **toString** ()

5.70.1 Detailed Description

Compatible with the POP-C++ paroc_string implementation.

5.70.2 Constructor & Destructor Documentation

5.70.2.1 popjava.dataswaper.POPString.POPString (String *value*)

Constructor with given value.

Parameters

value	String value to be stored in this object

5.70.3 Member Function Documentation

5.70.3.1 String popjava.dataswaper.POPString.getValue ( )

Get the current value of this object.

Returns

current string value

Here is the caller graph for this function:

5.70.3.2 void popjava.dataswaper.POPString.setValue (String *value*)

Set the string value of this object.

Parameters

value	new string value

Here is the caller graph for this function:

5.71 popjava.annotation.POPSyncConc Interface Reference

Collaboration diagram for popjava.annotation.POPSyncConc:

5.72 popjava.annotation.POPSyncMutex Interface Reference

Collaboration diagram for popjava.annotation.POPSyncMutex:

5.73 popjava.annotation.POPSyncSeq Interface Reference

Collaboration diagram for popjava.annotation.POPSyncSeq:

5.74 popjava.system.POPSystem Class Reference

This class is responsible for the initialization of a POP-Java application.

Collaboration diagram for popjava.system.POPSystem:

Public Member Functions

- POPSystem ()

  Creates a new instance of POPSystem.

Static Public Member Functions

- static void **writeLog** (String log)
- static int getIPAsInt ()

Retrieve the local IP address and format it as an int.

- static String getHostIP ()

    Get the host of the local node.
- static POPAccessPoint getDefaultAccessPoint ()

    Get the default local access point.
- static ObjectDescription getDefaultOD ()

    Get the default object description.
- static String getEnviroment (String name)

    Get the local environment variable.
- static String getPlatform ()

    Get the system platform.
- static String[] initialize (String...args)

    Entry point for the application scope initialization.
- static boolean initialize (ArrayList< String > argvList)

    Initialize the application scope services.
- static boolean initCodeService (String fileconf, String POPJavaObjectExecuteCommand, AppService app-CoreService) throws POPException

    Initialize the CodeMgr by reading the object map and register all code location.
- static AppService createAppCoreService (String codelocation) throws POPException

    Start the application scope services.
- static void **end** ()
- static boolean **isInitialized** ()

Static Public Attributes

- static final String PopLocationEnviromentName = "POP_LOCATION"

    POP-Java location environement variable name.
- static POPAccessPoint JobService = new POPAccessPoint()

    POP-Java Job service access point.
- static POPAccessPoint AppServiceAccessPoint = new POPAccessPoint()

    POP-Java application service access point.

5.74.1 Detailed Description

This class is responsible for the initialization of a POP-Java application.

It has also the responsibility to retrieve the configuration parameters.

5.74.2 Member Function Documentation

5.74.2.1 static **AppService** popjava.system.POPSystem.createAppCoreService (String *codelocation*) throws **POPException** [static]

Start the application scope services.

This services is a POP-C++ parallel object.

Parameters

codelocation	location of the POP-C++ AppCoreService executable file

Returns

Interface of AppCoreService

Exceptions

POPException	

Here is the call graph for this function:

5.74.2.2 static POPAccessPoint popjava.system.POPSystem.getDefaultAccessPoint ( ) [static]

Get the default local access point.

Returns

the default local access point

Here is the call graph for this function:

5.74.2.3 static ObjectDescription popjava.system.POPSystem.getDefaultOD ( ) [static]

Get the default object description.

Returns

a new empty object description

Here is the caller graph for this function:

5.74.2.4 static String popjava.system.POPSystem.getEnviroment (String *name*) [static]

Get the local environment variable.

Parameters

name	Name of the variable

Returns

Variable value or empty string

5.74.2.5 static String popjava.system.POPSystem.getHostIP ( ) [static]

Get the host of the local node.

Returns

Host name as a string value

Here is the caller graph for this function:

5.74.2.6 static int popjava.system.POPSystem.getIPAsInt ( ) [static]

Retrieve the local IP address and format it as an int.

Returns

int value of the local IP address

5.74.2.7 static String popjava.system.POPSystem.getPlatform ( ) [static]

Get the system platform.

Returns

platform as a string value

Here is the caller graph for this function:

5.74.2.8 static boolean popjava.system.POPSystem.initCodeService (String *fileconf*, String *POPJavaObjectExecuteCommand*, AppService *appCoreService*) throws POPException [static]

Initialize the CodeMgr by reading the object map and register all code location.

Parameters

fileconf	Object map file location
appCoreService	Reference to the AppCoreService

Returns

true if the initialization is well done

Exceptions

POPException	

Here is the call graph for this function:

Here is the caller graph for this function:

5.74.2.9 static String [ ] popjava.system.POPSystem.initialize (String... *args*) [static]

Entry point for the application scope initialization.

Parameters

argvs	Any arguments to pass to the initialization

Returns

true if the initialization is succeed

Exceptions

POPException	thrown is any problems occurred during the initialization

5.74.2.10 static boolean popjava.system.POPSystem.initialize (ArrayList< String > *argvList*) [static]

Initialize the application scope services.

Parameters

argvList	Any arguments to pass to the initialization

Returns

true if the initialization is succeed

Exceptions

POPException	thrown is any problems occurred during the initialization

Here is the call graph for this function:

5.75 popjava.base.POPSystemErrorCode Class Reference

This class regroup all exception code.

Collaboration diagram for popjava.base.POPSystemErrorCode:

Static Public Attributes

- static final int **EXCEPTION_INT** = 1
- static final int **EXCEPTION_UINT** = 2
- static final int **EXCEPTION_LONG** = 3
- static final int **EXCEPTION_ULONG** = 4
- static final int **EXCEPTION_SHORT** = 5
- static final int **EXCEPTION_USHORT** = 6
- static final int **EXCEPTION_BOOL** = 7
- static final int **EXCEPTION_CHAR** = 8
- static final int **EXCEPTION_UCHAR** = 9
- static final int **EXCEPTION_STRING** = 10
- static final int **EXCEPTION_FLOAT** = 11
- static final int **EXCEPTION_DOUBLE** = 12
- static final int **EXCEPTION_OBJECT** = 13
- static final int **EXCEPTION_PAROC_STD** = 14

5.75.1 Detailed Description

This class regroup all exception code.

5.76 popjava.broker.POPThread Class Reference

Base class of POPThread.

Inheritance diagram for popjava.broker.POPThread:

Collaboration diagram for popjava.broker.POPThread:

Public Member Functions

- POPThread (Request request)

    Creates a new instance of POPThread with a request.
- Request getRequest ()

    Return the request handled in the current POPThread.
- void setRequest (Request request)

    Set the request to be handled in this POPThread.
- void run ()

    Launch the execution of the current POPThread.

5.76.1 Detailed Description

Base class of POPThread.

Used to handle broker-side semantics

5.76.2 Member Function Documentation

5.76.2.1 Request popjava.broker.POPThread.getRequest ( )

Return the request handled in the current POPThread.

Returns

Request currently handled

5.76.2.2 void popjava.broker.POPThread.setRequest (Request *request*)

Set the request to be handled in this POPThread.

Parameters

request	Request to be handled

5.77 popjava.broker.Request Class Reference

This class symbolize a request between the interface-side and the broker-side.

Collaboration diagram for popjava.broker.Request:

Public Member Functions

- Request ()

    Creating a new pending request.
- Request (int classId, int methodId, int semantics, Broker broker, Combox combox)

    Creating a new specific request.
- void init (int classId, int methodId, int semantics, Broker broker, Combox combox)

    Initializes an empty request.
- int getClassId ()

    Get the class identifier of the current request.
- void setClassId (int classId)

Set the class identifier of the current request.
- int getMethodId ()

  Get the method identifier of this request.
- void setMethodId (int methodId)

  Set the method identifier of the current request.
- int getSenmatics ()

  Get the semantic of the current request.
- void setSenmatics (int semantics)

  Set the semantic of the current request.
- Broker getBroker ()

  Get the associated borker.
- void setBroker (Broker broker)

  Set an associated broker.
- POPBuffer getBuffer ()

  Get the associated buffer.
- void setBuffer (POPBuffer buffer)

  Set an associated buffer.
- int getStatus ()

  Get the request current status.
- void setStatus (int status)

  Set the current status of the request.
- ComboxReceiveRequestSocket getReceiveCombox ()

  Get the combox which received the request.
- void setReceiveCombox (ComboxReceiveRequestSocket combox)

  Get the combox which received the request.
- Combox getCombox ()

  Get the associated combox.
- void setCombox (Combox combox)

  Set the associated combox.
- void setBuffer (String bufferType)

  Set associated buffer.
- boolean isSynchronous ()

  Returns true if this request is a synchronous request, false if asynchronous.
- boolean isConcurrent ()

  Returns true if this request is a concurrent request, false otherwise.
- boolean isMutex ()

  Returns true if this request is a mutex request, false otherwise.
- boolean isSequential ()

  Returns true if this request is a sequential request, false otherwise.

Static Public Attributes

- static final int **Pending** = 0
- static final int **Serving** = 1
- static final int **Served** = 2

Protected Attributes

- int **classId**
- int **methodId**
- int **semantics**
- Broker **broker**
- POPBuffer **buffer**
- ComboxReceiveRequestSocket **receivedCombox**
- Combox **combox**
- int **status**

5.77.1 Detailed Description

This class symbolize a request between the interface-side and the broker-side.

5.77.2 Constructor & Destructor Documentation

5.77.2.1 popjava.broker.Request.Request (int *classId,* int *methodId,* int *semantics,* Broker *broker,* Combox *combox*)

Creating a new specific request.

Parameters

classId	Class identifier for this request
methodId	Method identifier for this request
semantics	Semantics used for this methods
broker	Broker associated with this request
combox	Combox associated with this request

5.77.3 Member Function Documentation

5.77.3.1 Broker popjava.broker.Request.getBroker ( )

Get the associated borker.

Returns

reference to the associated broker

Here is the caller graph for this function:

5.77.3.2 POPBuffer popjava.broker.Request.getBuffer ( )

Get the associated buffer.

Returns

reference to the associated buffer

Here is the caller graph for this function:

5.77.3.3 int popjava.broker.Request.getClassId ( )

Get the class identifier of the current request.

Returns

class identifier

5.77.3.4 Combox popjava.broker.Request.getCombox ( )

Get the associated combox.

Returns

associated combox

Here is the caller graph for this function:

5.77.3.5 int popjava.broker.Request.getMethodId ( )

Get the method identifier of this request.

Returns

method identifier

Here is the caller graph for this function:

5.77.3.6 ComboxReceiveRequestSocket popjava.broker.Request.getReceiveCombox ( )

Get the combox which received the request.

Returns

combox which received the request

5.77.3.7 int popjava.broker.Request.getSenmatics ( )

Get the semantic of the current request.

Returns

Semantic of the current request as an int value

Here is the caller graph for this function:

5.77.3.8 int popjava.broker.Request.getStatus ( )

Get the request current status.

Returns

status of the current request

5.77.3.9 void popjava.broker.Request.init (int *classId,* int *methodId,* int *semantics,* Broker *broker,* Combox *combox*)

Initializes an empty request.

Parameters

classId	Class identifier for this request
methodId	Method identifier for this request
semantics	Semantics used for this methods
broker	Broker associated with this request
combox	Combox associated with this request

5.77.3.10 boolean popjava.broker.Request.isConcurrent ( )

Returns true if this request is a concurrent request, false otherwise.

Returns

Here is the call graph for this function:

Here is the caller graph for this function:

5.77.3.11 boolean popjava.broker.Request.isMutex ( )

Returns true if this request is a mutex request, false otherwise.

Returns

Here is the call graph for this function:

Here is the caller graph for this function:

5.77.3.12 boolean popjava.broker.Request.isSequential ( )

Returns true if this request is a sequential request, false otherwise.

Returns

Here is the call graph for this function:

Here is the caller graph for this function:

5.77.3.13 boolean popjava.broker.Request.isSynchronous ( )

Returns true if this request is a synchronous request, false if asynchronous.

Returns

Here is the call graph for this function:

5.77.3.14 void popjava.broker.Request.setBroker (Broker *broker*)

Set an associated broker.

Parameters

broker	Reference to the associated broker to be set

Here is the caller graph for this function:

5.77.3.15 void popjava.broker.Request.setBuffer (POPBuffer *buffer*)

Set an associated buffer.

Parameters

buffer	Reference to the associated buffer

Here is the caller graph for this function:

5.77.3.16 void popjava.broker.Request.setBuffer (String *bufferType*)

Set associated buffer.

Parameters

bufferType	BufferType to be associate

5.77.3.17 void popjava.broker.Request.setClassId (int *classId*)

Set the class identifier of the current request.

Parameters

classId	Class ID to be set

Here is the caller graph for this function:

5.77.3.18 void popjava.broker.Request.setCombox (Combox *combox*)

Set the associated combox.

Parameters

combox	Combox to be associate

Here is the caller graph for this function:

5.77.3.19 void popjava.broker.Request.setMethodId (int *methodId*)

Set the method identifier of the current request.

Parameters

methodId	Method ID to be set

Here is the caller graph for this function:

5.77.3.20 void popjava.broker.Request.setReceiveCombox (ComboxReceiveRequestSocket *combox*)

Get the combox which received the request.

Parameters

combox	Combox which received the request

Here is the caller graph for this function:

5.77.3.21 void popjava.broker.Request.setSenmatics (int *semantics*)

Set the semantic of the current request.

Parameters

semantics	Semantic to be set on the current request as an int value

Here is the caller graph for this function:

5.77.3.22 void popjava.broker.Request.setStatus (int *status*)

Set the current status of the request.

Parameters

status	Status to be set

Here is the caller graph for this function:

5.78 popjava.broker.RequestQueue Class Reference

This class represents the request queue used in the broker-side Every requests are put into this request queue and are served in FIFO order.

Collaboration diagram for popjava.broker.RequestQueue:

Public Member Functions

- RequestQueue ()

  Creates a new instance of POPRequestQueue.
- synchronized int size ()

  Give the actual number of requests in the queue.
- synchronized int getMaxQueue ()

  Return the maximum number of requests in the queue.
- synchronized void setMaxQueue (int maxQueue)

  Set the maximum number of requests in the queue.
- boolean add (Request request)

  Put a new request in the queue.
- Request peek (int time, TimeUnit timeUnit)

  Peek a request in the queue.

- boolean remove (Request request)

  Remove a specific request from the queue.
- synchronized boolean clear ()

  Clear the queue.
- boolean canPeek ()

  Check if there is request to peek.

Static Public Attributes

- static final int **DEFAULT_REQUEST_QUEUE_SIZE** = 250

5.78.1 Detailed Description

This class represents the request queue used in the broker-side Every requests are put into this request queue and are served in FIFO order.

5.78.2 Member Function Documentation

5.78.2.1 boolean popjava.broker.RequestQueue.add (Request *request*)

Put a new request in the queue.

Parameters

request	Request to add

Returns

true if the request is added correctly

Here is the call graph for this function:

Here is the caller graph for this function:

5.78.2.2 boolean popjava.broker.RequestQueue.canPeek ( )

Check if there is request to peek.

Returns

true if a request can be peeked

Here is the caller graph for this function:

5.78.2.3 synchronized boolean popjava.broker.RequestQueue.clear ( )

Clear the queue.

Returns

true if the queue if correctly cleared

5.78.2.4 synchronized int popjava.broker.RequestQueue.getMaxQueue ( )

Return the maximum number of requests in the queue.

Returns

max requests number in the queue

5.78.2.5 Request popjava.broker.RequestQueue.peek (int *time*, TimeUnit *timeUnit*)

Peek a request in the queue.

If there is no request to peek, this method waits the time passed in parameters

Parameters

time	Time to wait
timeUnit	Unit of time

Returns

Request peeked in the queue

Here is the call graph for this function:

5.78.2.6 boolean popjava.broker.RequestQueue.remove (Request *request*)

Remove a specific request from the queue.

Parameters

request	Request to be removed

Returns

true if the request is correctly removed

Here is the call graph for this function:

5.78.2.7 synchronized void popjava.broker.RequestQueue.setMaxQueue (int *maxQueue*)

Set the maximum number of requests in the queue.

Parameters

maxQueue	Maximum number of requests

5.78.2.8 synchronized int popjava.broker.RequestQueue.size ( )

Give the actual number of requests in the queue.

Returns

number of requests

5.79 popjava.scripts.ScriptUtils Class Reference

Collaboration diagram for popjava.scripts.ScriptUtils:

Static Public Member Functions

- static String **getNewline** ()
- static boolean **isWindows** ()
- static boolean **containsOption** (String[] args, String option)
- static boolean **removeOption** (List< String > parameters, String...options)
- static String **getOption** (List< String > parameters, String defaultValue, String...options)
- static List< String > **arrayToList** (String...args)
- static String[] **listToArray** (List< String > list)
- static void **runNativeApplication** (String[] arguments, String notFoundError, BufferedWriter out, boolean verbose)

5.80 popjava.base.Semantic Class Reference

This class class is used to store the different semantics used in the POP model.

Collaboration diagram for popjava.base.Semantic:

Static Public Attributes

- static final int **Synchronous** = 1
- static final int **Asynchronous** = 0
- static final int **Constructor** = 4
- static final int **Concurrent** = 8
- static final int **Mutex** = 16
- static final int **Sequence** = 0

5.80.1 Detailed Description

This class class is used to store the different semantics used in the POP model.

The different semantics from this class can be combined with the | operator. Synchronous and Asynchronous must not be combined together. Concurrent, Sequence and mutex must not be combined together.

5.81 popjava.util.SystemUtil Class Reference

This glass gives some static method to deal with the system.

Collaboration diagram for popjava.util.SystemUtil:

Static Public Member Functions

- static void **endAllChildren** ()
- static int runCmd (List< String > argvs)

  Run a new command.
- static boolean **commandExists** (String command)
- static int **runRemoteCmdSSHJ** (String url, List< String > command)
- static int **runRemoteCmd** (String url, List< String > command)

5.81.1 Detailed Description

This glass gives some static method to deal with the system.

5.81.2 Member Function Documentation

5.81.2.1 static int popjava.util.SystemUtil.runCmd (List< String > *argvs*) `[static]`

Run a new command.

Parameters

argvs	arguments to pass to the new process

Returns

0 if the command launch is a success

Here is the call graph for this function:

5.82 popjava.annotation.POPConfig.Type Enum Reference

Collaboration diagram for popjava.annotation.POPConfig.Type:

Public Attributes

- **URL**

5.83 popjava.util.Util Class Reference

This class gives some static utility methods.

Collaboration diagram for popjava.util.Util:

Static Public Member Functions

- static boolean sameContact (String source, String dest)

  Check if the two contact string are the same.
- static boolean isLocal (String hostname)

  Check if the contact string is the local host.
- static String removeStringFromArrayList (ArrayList< String > list, String prefix)

  Remove a string in an array list.
- static boolean isStringEqual (String s1, String s2)

  Compare two no null Strings.
- static boolean isNoCaseStringEqual (String s1, String s2)

  Compare two not null string.
- static String generateRandomString (int length)

  Generate a random string of the given length.
- static ArrayList< String > splitTheCommand (String command)

  Split a command formatted as a string value into an array list.
- static boolean matchPlatform (String parent, String child)

Match a parent platform string with a child platform string.

- static int byteArrayToInt (byte[] value)

  Transform a byte array into an int value.
- static boolean isParameterNotOfDirection (Annotation[] annotations, POPParameter.Direction direction)

  Returns true of one of the annotations defines a IN only parameter.
- static boolean **isParameterOfAnyDirection** (Annotation[] annotations)

5.83.1 Detailed Description

This class gives some static utility methods.

5.83.2 Member Function Documentation

5.83.2.1 static int popjava.util.Util.byteArrayToInt (byte[] *value*) `[static]`

Transform a byte array into an int value.

Parameters

value	The byte array to transform

Returns

The int value

5.83.2.2 static String popjava.util.Util.generateRandomString (int *length*) `[static]`

Generate a random string of the given length.

Parameters

length	Length of the generated string

Returns

The generated string

5.83.2.3 static boolean popjava.util.Util.isLocal (String *hostname*) `[static]`

Check if the contact string is the local host.

Parameters

hostname	Contact string

Returns

true if the contact string is the local host

Here is the call graph for this function:

5.83.2.4 static boolean popjava.util.Util.isNoCaseStringEqual (String *s1*, String *s2*) [static]

Compare two not null string.

Case insensitive

Parameters

s1	First string
s2	Second String

Returns

true if the strings are equal

5.83.2.5 static boolean popjava.util.Util.isParameterNotOfDirection (Annotation[] *annotations*, POPParameter.Direction *direction*) [static]

Returns true of one of the annotations defines a IN only parameter.

Parameters

annotations	

Returns

Here is the caller graph for this function:

5.83.2.6 static boolean popjava.util.Util.isStringEqual (String *s1*, String *s2*) [static]

Compare two no null Strings.

Parameters

s1	First string
s2	Second string

Returns

true if the strings are equal

5.83.2.7 static boolean popjava.util.Util.matchPlatform (String *parent*, String *child*) [static]

Match a parent platform string with a child platform string.

Parameters

parent	The parent platform string
child	The child platform string

Returns

true if the string match

5.83.2.8 static String popjava.util.Util.removeStringFromArrayList (ArrayList< String > *list*, String *prefix*) [static]

Remove a string in an array list.

Parameters

list	The array list to work with
prefix	The prefix of the string to remove

Returns

The entire string removed

Here is the caller graph for this function:

5.83.2.9 static boolean popjava.util.Util.sameContact (String *source*, String *dest*) [static]

Check if the two contact string are the same.

Parameters

source	First contact string
dest	Second contact string

Returns

true if the contact strings are the same

5.83.2.10 static ArrayList<String> popjava.util.Util.splitTheCommand (String *command*) [static]

Split a command formatted as a string value into an array list.

Parameters

command	The command formatted as a string value

Returns

The split command as an array list

5.84 popjava.system.XMLWorker Class Reference

Base class to handle XML validation.

Inheritance diagram for popjava.system.XMLWorker:

Collaboration diagram for popjava.system.XMLWorker:

Public Member Functions

- XMLWorker ()
  Default XMLWorker empty constructor.
- boolean isValid (String xmlFile, String xmlSchema)
  Validate an XML file with an XML schema.

Static Protected Attributes

- static final String **XML_FILE_EXTENSION** = ".xml"
- static final String **XSD_FILE_EXTENSION** = ".xsd"

5.84.1 Detailed Description

Base class to handle XML validation.

Author

clementval

5.84.2 Member Function Documentation

5.84.2.1 boolean popjava.system.XMLWorker.isValid (String *xmlFile,* String *xmlSchema*)

Validate an XML file with an XML schema.

Parameters

xmlFile	location of the XML file
xmlSchema	location of the XML schema

Returns

true if the XML file is valid

Here is the caller graph for this function:

Index

AccessPoint
  popjava::baseobject::AccessPoint, 14
add
  popjava::broker::RequestQueue, 176
addAccessPoint
  popjava::baseobject::POPAccessPoint, 114
addSemantic
  popjava::base::POPObject, 151
allocResource
  popjava::serviceadapter::POPJobManager, 146
allocate
  popjava::interfacebase::Interface, 77

bind
  popjava::interfacebase::Interface, 78
bindObject
  popjava::PJMethodHandler, 109
bindPOPObject
  popjava::PJProxyFactory, 111
BufferRaw
  popjava::buffer::BufferRaw, 39
BufferXDR
  popjava::buffer::BufferXDR, 50
byteArrayToInt
  popjava::util::Util, 180

canKill
  popjava::base::POPObject, 151
canPeek
  popjava::broker::RequestQueue, 176
cancelReservation
  popjava::serviceadapter::POPJobManager, 146
checkAndThrow
  popjava::buffer::POPBuffer, 120
checkObjects
  popjava::serviceadapter::POPObjectMonitor, 158
clear
  popjava::broker::RequestQueue, 176
clearResourceAfterInvoke
  popjava::broker::Broker, 20
Combox
  popjava::combox::Combox, 54
ComboxAcceptSocket
  popjava::combox::ComboxAcceptSocket, 56
ComboxReceiveRequestSocket
  popjava::combox::ComboxReceiveRequestSocket, 66
ComboxServer
  popjava::combox::ComboxServer, 67
ComboxServerPlugin
  popjava::combox::ComboxServerPlugin, 68
ComboxServerSocket
  popjava::combox::ComboxServerSocket, 69
ComboxSocket
  popjava::combox::ComboxSocket, 71
ConfigurationWorker
  popjava::system::ConfigurationWorker, 75
connect
  popjava::combox::Combox, 54
  popjava::combox::ComboxPlugin, 64
  popjava::combox::ComboxSocket, 71
create
  popjava::baseobject::AccessPoint, 14
createAppCoreService
  popjava::system::POPSystem, 165
createClientCombox
  popjava::combox::ComboxFactory, 58
  popjava::combox::ComboxFactoryPlugin, 62
  popjava::combox::ComboxSocketFactory, 72, 73
createObject
  popjava::serviceadapter::POPJobManager, 146
  popjava::serviceadapter::POPJobService, 148
createReflectException
  popjava::base::POPException, 136
createReflectMethodNotFoundException
  popjava::base::POPException, 136
createServerCombox
  popjava::combox::ComboxFactory, 59
  popjava::combox::ComboxFactoryPlugin, 63
  popjava::combox::ComboxSocketFactory, 73

defineConstructor
  popjava::base::POPObject, 151
defineMethod
  popjava::base::POPObject, 152
deleteLogDir
  popjava::util::LogWriter, 83
deserialize
  popjava::base::POPException, 137
  popjava::base::POPObject, 152
  popjava::baseobject::ODElement, 106
  popjava::dataswaper::IPOPBase, 80
  popjava::dataswaper::IPOPBaseInput, 82
  popjava::interfacebase::Interface, 78
deserializeReferenceObject
  popjava::buffer::POPBuffer, 120

equals
  popjava::base::MethodInfo, 88
execObj

popjava::serviceadapter::POPJobManager, 147
extractHeader
popjava::buffer::BufferPlugin, 28
popjava::buffer::BufferRaw, 39
popjava::buffer::POPBuffer, 121

findEndcoding
popjava::broker::Broker, 20
findFactory
popjava::buffer::BufferFactoryFinder, 25
popjava::combox::ComboxFactoryFinder, 60

generateRandomString
popjava::util::Util, 180
get
popjava::baseobject::POPAccessPoint, 114
popjava::buffer::BufferPlugin, 28
popjava::buffer::BufferRaw, 39
popjava::buffer::POPBuffer, 121
popjava::combox::ComboxFactoryFinder, 60
getAccessPoint
popjava::base::POPObject, 152
popjava::broker::Broker, 20
popjava::interfacebase::Interface, 78
getArray
popjava::buffer::POPBuffer, 121
getBatch
popjava::baseobject::ObjectDescription, 92
getBoolean
popjava::buffer::BufferPlugin, 28
popjava::buffer::BufferRaw, 39
popjava::buffer::POPBuffer, 121
getBooleanArray
popjava::buffer::BufferPlugin, 29
popjava::buffer::BufferRaw, 39
popjava::buffer::POPBuffer, 122
getBroker
popjava::broker::Request, 171
getBuffer
popjava::broker::Request, 171
getBufferFactory
popjava::combox::Combox, 54
getBufferName
popjava::buffer::BufferFactory, 24
popjava::buffer::BufferFactoryPlugin, 26
getByteArray
popjava::buffer::BufferPlugin, 29
popjava::buffer::BufferRaw, 40
popjava::buffer::POPBuffer, 122
getChar
popjava::buffer::BufferPlugin, 29
popjava::buffer::BufferRaw, 40
popjava::buffer::POPBuffer, 122
getCharArray
popjava::buffer::BufferPlugin, 29
popjava::buffer::BufferRaw, 40
popjava::buffer::POPBuffer, 122
getClassId
popjava::base::MessageHeader, 85
popjava::base::MethodInfo, 88
popjava::base::POPObject, 152
popjava::broker::Request, 171
getClassName
popjava::base::POPObject, 153
getCode
popjava::base::BindStatus, 17
getCodeFile
popjava::baseobject::ObjectDescription, 92
popjava::dataswaper::ObjectDescriptionInput, 101
getCombox
popjava::broker::Request, 172
getComboxName
popjava::combox::ComboxFactory, 59
popjava::combox::ComboxFactoryPlugin, 63
popjava::combox::ComboxSocketFactory, 73
getConstructor
popjava::util::ClassUtil, 51
getConstructorByInfo
popjava::base::POPObject, 153
getDefaultAccessPoint
popjava::system::POPSystem, 166
getDefaultOD
popjava::system::POPSystem, 166
getDefaultPrimitiveValue
popjava::util::ClassUtil, 51
getDouble
popjava::buffer::BufferPlugin, 30
popjava::buffer::BufferRaw, 40
popjava::buffer::POPBuffer, 123
getDoubleArray
popjava::buffer::BufferPlugin, 30
popjava::buffer::BufferRaw, 40
popjava::buffer::POPBuffer, 123
getEncoding
popjava::baseobject::ObjectDescription, 92
popjava::dataswaper::ObjectDescriptionInput, 101
getEnviroment
popjava::system::POPSystem, 166
getExceptionCode
popjava::base::MessageHeader, 86
getFactoryCount
popjava::combox::ComboxFactoryFinder, 61
getFilename
popjava::system::POPRemoteLogThread, 161
getFloat
popjava::buffer::BufferPlugin, 30
popjava::buffer::BufferRaw, 41
popjava::buffer::POPBuffer, 123
getFloatArray
popjava::buffer::BufferPlugin, 30
popjava::buffer::BufferRaw, 41
popjava::buffer::POPBuffer, 123
getHeader
popjava::buffer::POPBuffer, 124
getHost
popjava::baseobject::AccessPoint, 14
getHostIP

popjava::system::POPSystem, 166
getHostName
popjava::baseobject::ObjectDescription, 92
popjava::dataswaper::ObjectDescriptionInput, 101
getHostarch
popjava::baseobject::ObjectDescription, 92
getHostcore
popjava::baseobject::ObjectDescription, 92
getHostuser
popjava::baseobject::ObjectDescription, 92
getIPAsInt
popjava::system::POPSystem, 166
getInfo
popjava::base::BindStatus, 17
getInstance
popjava::buffer::BufferFactoryFinder, 25
popjava::combox::ComboxFactoryFinder, 61
getInt
popjava::buffer::BufferPlugin, 30
popjava::buffer::BufferRaw, 41
popjava::buffer::POPBuffer, 124
getIntArray
popjava::buffer::BufferPlugin, 31
popjava::buffer::BufferRaw, 42
popjava::buffer::POPBuffer, 124
getJVMParameters
popjava::baseobject::ObjectDescription, 93
getJobUrl
popjava::baseobject::ObjectDescription, 93
popjava::dataswaper::ObjectDescriptionInput, 101
getLogPrefix
popjava::broker::Broker, 20
getLong
popjava::buffer::BufferPlugin, 31
popjava::buffer::BufferRaw, 42
popjava::buffer::POPBuffer, 124
getLongArray
popjava::buffer::BufferPlugin, 31
popjava::buffer::BufferRaw, 42
popjava::buffer::POPBuffer, 124
getMaxQueue
popjava::broker::RequestQueue, 176
getMethod
popjava::util::ClassUtil, 52
getMethodByInfo
popjava::base::POPObject, 153
getMethodId
popjava::base::MessageHeader, 86
popjava::base::MethodInfo, 88
popjava::broker::Request, 172
getMethodInfo
popjava::base::POPObject, 153, 154
getMethodSign
popjava::util::ClassUtil, 52
getMinValue
popjava::baseobject::ODElement, 106
getOD
popjava::interfacebase::Interface, 78
getOd
popjava::base::POPObject, 154
getPOPCReference
popjava::base::POPObject, 154
getPOPObjectClass
popjava::broker::Broker, 20
getPlatform
popjava::base::BindStatus, 17
popjava::baseobject::ObjectDescription, 93
popjava::codemanager::POPJavaAppService, 142
popjava::dataswaper::ObjectDescriptionInput, 102
popjava::serviceadapter::POPCodeManager, 133
popjava::system::POPSystem, 167
getPopAppCoreService
popjava::system::POPJavaConfiguration, 144
getPopJavaLocation
popjava::system::POPJavaConfiguration, 144
getPopPluginLocation
popjava::system::POPJavaConfiguration, 144
getPort
popjava::baseobject::AccessPoint, 15
getPowerMin
popjava::baseobject::ObjectDescription, 93
getProtocol
popjava::baseobject::AccessPoint, 15
popjava::baseobject::ObjectDescription, 93
popjava::dataswaper::ObjectDescriptionInput, 102
getReceiveCombox
popjava::broker::Request, 172
getRequest
popjava::broker::POPThread, 169
getRequestQueue
popjava::combox::ComboxServer, 68
popjava::combox::ComboxServerPlugin, 68
getRequestType
popjava::base::MessageHeader, 86
getRequiredValue
popjava::baseobject::ODElement, 106
getSearchMaxDepth
popjava::baseobject::ObjectDescription, 93
getSearchMaxSize
popjava::baseobject::ObjectDescription, 94
getSearchWaitTime
popjava::baseobject::ObjectDescription, 94
getSemantic
popjava::base::POPObject, 154, 155
getSenmatics
popjava::base::MessageHeader, 86
popjava::broker::Request, 172
getShort
popjava::buffer::BufferPlugin, 31
popjava::buffer::BufferRaw, 42
popjava::buffer::POPBuffer, 125
getShortArray
popjava::buffer::BufferPlugin, 31
popjava::buffer::BufferRaw, 43
popjava::buffer::POPBuffer, 125
getState

popjava::broker::Broker, 21
getStatus
popjava::broker::Request, 172
popjava::combox::ComboxAcceptSocket, 56
popjava::combox::ComboxReceiveRequestSocket, 66
getString
popjava::buffer::BufferPlugin, 32
popjava::buffer::BufferRaw, 43
popjava::buffer::POPBuffer, 125
getSupportingBuffer
popjava::buffer::BufferFactoryFinder, 25
getTranslatedInteger
popjava::buffer::BufferPlugin, 32
popjava::buffer::BufferRaw, 43
popjava::buffer::POPBuffer, 125
GetUrl
popjava::combox::ComboxServerSocket, 69
getUrl
popjava::combox::ComboxAllocateSocket, 57
getValue
popjava::baseobject::ObjectDescription, 94
popjava::buffer::POPBuffer, 126
popjava::dataswaper::ObjectDescriptionInput, 102
popjava::dataswaper::POPString, 164
popjava::system::ConfigurationWorker, 75
getWallTime
popjava::baseobject::ObjectDescription, 94
popjava::dataswaper::ObjectDescriptionInput, 102

hasDestructor
popjava::base::POPObject, 155

init
popjava::broker::Request, 172
initCodeService
popjava::system::POPSystem, 167
initialize
popjava::broker::Broker, 21
popjava::system::POPSystem, 167
initializeConstructorInfo
popjava::base::POPObject, 155
initializeMethodInfo
popjava::base::POPObject, 155
initializePOPObject
popjava::base::POPObject, 156
Interface
popjava::interfacebase::Interface, 77
invoke
popjava::broker::Broker, 21
popjava::PJMethodHandler, 109
isAlive
popjava::interfacebase::Interface, 79
isConcurrent
popjava::broker::Request, 173
isDaemon
popjava::base::POPObject, 156
popjava::broker::Broker, 22
isEmpty
popjava::baseobject::AccessPoint, 15
popjava::baseobject::ObjectDescription, 94
popjava::baseobject::ODElement, 106
popjava::baseobject::POPAccessPoint, 114
popjava::dataswaper::ObjectDescriptionInput, 102
isHandled
popjava::PJMethodFilter, 108
isLocal
popjava::util::Util, 180
isMutex
popjava::broker::Request, 173
isNoCaseStringEqual
popjava::util::Util, 180
isParameterNotOfDirection
popjava::util::Util, 181
isSearchSet
popjava::baseobject::ObjectDescription, 95
isSequential
popjava::broker::Request, 173
isStringEqual
popjava::util::Util, 181
isSynchronous
popjava::broker::Request, 173
isValid
popjava::system::XMLWorker, 183

jvmParameters
popjava::annotation::POPObjectDescription, 157

limit
popjava::buffer::BufferRaw, 43
loadBufferMap
popjava::buffer::BufferFactoryFinder, 25
loadComboxMap
popjava::combox::ComboxFactoryFinder, 61
log
popjava::serviceadapter::POPRemoteLog, 160

main
popjava::broker::Broker, 22
manageObject
popjava::serviceadapter::POPObjectMonitor, 158
manual
popjava::baseobject::ObjectDescription, 95
matchPlatform
popjava::util::Util, 181
merge
popjava::baseobject::ObjectDescription, 95
popjava::dataswaper::ObjectDescriptionInput, 102
MessageHeader
popjava::base::MessageHeader, 85
MethodInfo
popjava::base::MethodInfo, 88

newActive
popjava::PopJava, 140, 141
newActiveFromBuffer
popjava::PJProxyFactory, 111
popjava::PopJava, 141

newPOPObject
popjava::PJProxyFactory, 112

ODElement
popjava::baseobject::ODElement, 106
ObjectDescriptionInput
popjava::dataswaper::ObjectDescriptionInput, 101

PJMethodHandler
popjava::PJMethodHandler, 109
PJProxyFactory
popjava::PJProxyFactory, 111
POPAccessPoint
popjava::baseobject::POPAccessPoint, 113, 114
POPAppService
popjava::serviceadapter::POPAppService, 115
POPBuffer
popjava::buffer::POPBuffer, 120
POPCodeManager
popjava::serviceadapter::POPCodeManager, 133
POPException
popjava::base::POPException, 136
POPJobManager
popjava::serviceadapter::POPJobManager, 145, 146
POPJobService
popjava::serviceadapter::POPJobService, 148
POPObjectMonitor
popjava::serviceadapter::POPObjectMonitor, 158
POPRemoteLog
popjava::serviceadapter::POPRemoteLog, 160
POPRemoteLogThread
popjava::system::POPRemoteLogThread, 161
POPServiceBase
popjava::serviceadapter::POPServiceBase, 162
POPString
popjava::dataswaper::POPString, 163
packMessageHeader
popjava::buffer::BufferPlugin, 32
popjava::buffer::BufferRaw, 43
popjava::buffer::POPBuffer, 126
peek
popjava::broker::RequestQueue, 177
popCall
popjava::broker::Broker, 22
popConstructor
popjava::PJMethodHandler, 110
popDispatch
popjava::interfacebase::Interface, 79
popResponse
popjava::interfacebase::Interface, 79
popjava.annotation.POPAsyncConc, 117
popjava.annotation.POPAsyncMutex, 117
popjava.annotation.POPAsyncSeq, 117
popjava.annotation.POPClass, 132
popjava.annotation.POPConfig, 134
popjava.annotation.POPConfig.Type, 179
popjava.annotation.POPObjectDescription, 157
popjava.annotation.POPParameter, 159
popjava.annotation.POPParameter.Direction, 76
popjava.annotation.POPSyncConc, 164
popjava.annotation.POPSyncMutex, 164
popjava.annotation.POPSyncSeq, 164
popjava.annotation.processors.POPClassProcessor, 132
popjava.base.BindStatus, 16
popjava.base.MessageHeader, 84
popjava.base.MethodInfo, 87
popjava.base.POPErrorCode, 134
popjava.base.POPException, 135
popjava.base.POPObject, 149
popjava.base.POPSystemErrorCode, 168
popjava.base.Semantic, 178
popjava.baseobject.AccessPoint, 13
popjava.baseobject.ODElement, 105
popjava.baseobject.ObjectDescription, 89
popjava.baseobject.POPAccessPoint, 113
popjava.baseobject.POPReference, 159
popjava.broker.Broker, 18
popjava.broker.POPThread, 168
popjava.broker.Request, 169
popjava.broker.RequestQueue, 175
popjava.buffer.BufferFactory, 24
popjava.buffer.BufferFactoryFinder, 24
popjava.buffer.BufferFactoryPlugin, 26
popjava.buffer.BufferPlugin, 26
popjava.buffer.BufferRaw, 36
popjava.buffer.BufferRawFactory, 49
popjava.buffer.BufferXDR, 49
popjava.buffer.BufferXDRFactory, 50
popjava.buffer.POPBuffer, 117
popjava.codemanager.AppService, 16
popjava.codemanager.POPJavaAppService, 142
popjava.combox.Combox, 53
popjava.combox.ComboxAcceptSocket, 55
popjava.combox.ComboxAllocateSocket, 56
popjava.combox.ComboxFactory, 58
popjava.combox.ComboxFactoryFinder, 60
popjava.combox.ComboxFactoryPlugin, 61
popjava.combox.ComboxPlugin, 63
popjava.combox.ComboxReceiveRequestSocket, 65
popjava.combox.ComboxServer, 67
popjava.combox.ComboxServerPlugin, 68
popjava.combox.ComboxServerSocket, 69
popjava.combox.ComboxSocket, 70
popjava.combox.ComboxSocketFactory, 72
popjava.dataswaper.IPOPBase, 80
popjava.dataswaper.IPOPBaseConst, 81
popjava.dataswaper.IPOPBaseInput, 81
popjava.dataswaper.ObjectDescriptionInput, 99
popjava.dataswaper.POPString, 163
popjava.interfacebase, 11
popjava.interfacebase.Interface, 76
popjava.PJMethodFilter, 108
popjava.PJMethodHandler, 108
popjava.PJProxyFactory, 110
popjava.PopJava, 140

popjava.scripts.Popjavac, 143
popjava.scripts.Popjrun, 149
popjava.scripts.ScriptUtils, 178
popjava.serviceadapter.POPAppService, 115
popjava.serviceadapter.POPCodeManager, 132
popjava.serviceadapter.POPJobManager, 144
popjava.serviceadapter.POPJobService, 148
popjava.serviceadapter.POPObjectMonitor, 157
popjava.serviceadapter.POPRemoteLog, 160
popjava.serviceadapter.POPServiceBase, 162
popjava.system.ConfigurationWorker, 74
popjava.system.POPJavaConfiguration, 143
popjava.system.POPRemoteLogThread, 161
popjava.system.POPSystem, 164
popjava.system.XMLWorker, 182
popjava.util.ClassUtil, 51
popjava.util.Configuration, 74
popjava.util.LogWriter, 82
popjava.util.SystemUtil, 178
popjava.util.Util, 179
popjava::PJMethodFilter
    isHandled, 108
popjava::PJMethodHandler
    bindObject, 109
    invoke, 109
    PJMethodHandler, 109
    popConstructor, 110
popjava::PJProxyFactory
    bindPOPObject, 111
    newActiveFromBuffer, 111
    newPOPObject, 112
    PJProxyFactory, 111
popjava::PopJava
    newActive, 140, 141
    newActiveFromBuffer, 141
popjava::annotation::POPObjectDescription
    jvmParameters, 157
popjava::base::BindStatus
    getCode, 17
    getInfo, 17
    getPlatform, 17
    setCode, 17
    setInfo, 18
    setPlatform, 18
popjava::base::MessageHeader
    getClassId, 85
    getExceptionCode, 86
    getMethodId, 86
    getRequestType, 86
    getSenmatics, 86
    MessageHeader, 85
    setClassId, 86
    setExceptionCode, 86
    setMethodId, 87
    setRequestType, 87
    setSenmatics, 87
popjava::base::MethodInfo
    equals, 88
    getClassId, 88
    getMethodId, 88
    MethodInfo, 88
popjava::base::POPException
    createReflectException, 136
    createReflectMethodNotFoundException, 136
    deserialize, 137
    POPException, 136
    serialize, 137
    throwAccessPointNotAvailableException, 137
    throwBufferFormatException, 137
    throwBufferNotAvailableException, 138
    throwComboxNotAvailableException, 138
    throwNullObjectNotAllowException, 138
    throwObjectBindException, 139
    throwObjectNoResource, 139
    throwReflectException, 139
    throwReflectMethodNotFoundException, 139
    throwReflectSerializeException, 139
popjava::base::POPObject
    addSemantic, 151
    canKill, 151
    defineConstructor, 151
    defineMethod, 152
    deserialize, 152
    getAccessPoint, 152
    getClassId, 152
    getClassName, 153
    getConstructorByInfo, 153
    getMethodByInfo, 153
    getMethodInfo, 153, 154
    getOd, 154
    getPOPCReference, 154
    getSemantic, 154, 155
    hasDestructor, 155
    initializeConstructorInfo, 155
    initializeMethodInfo, 155
    initializePOPObject, 156
    isDaemon, 156
    serialize, 156
    setClassId, 156
    setClassName, 156
    setOd, 157
popjava::baseobject::AccessPoint
    AccessPoint, 14
    create, 14
    getHost, 14
    getPort, 15
    getProtocol, 15
    isEmpty, 15
    setHost, 15
    setPort, 15
    setProtocol, 15
popjava::baseobject::ODElement
    deserialize, 106
    getMinValue, 106
    getRequiredValue, 106
    isEmpty, 106

ODElement, 106
serialize, 107
set, 107
setMinValue, 107
setRequiredValue, 107
popjava::baseobject::ObjectDescription
getBatch, 92
getCodeFile, 92
getEncoding, 92
getHostName, 92
getHostarch, 92
getHostcore, 92
getHostuser, 92
getJVMParameters, 93
getJobUrl, 93
getPlatform, 93
getPowerMin, 93
getProtocol, 93
getSearchMaxDepth, 93
getSearchMaxSize, 94
getSearchWaitTime, 94
getValue, 94
getWallTime, 94
isEmpty, 94
isSearchSet, 95
manual, 95
merge, 95
removeValue, 95
setBandwidth, 95
setBatch, 95
setCodeFile, 96
setDirectory, 96
setEncoding, 96
setHostarch, 96
setHostcore, 96
setHostname, 97
setHostuser, 97
setJVMParamters, 97
setJobUrl, 97
setMemory, 97
setPlatform, 97
setPower, 98
setProtocol, 98
setSearch, 98
setValue, 98
setWallTime, 98
popjava::baseobject::POPAccessPoint
addAccessPoint, 114
get, 114
isEmpty, 114
POPAccessPoint, 113, 114
setAccessString, 114
size, 115
popjava::baseobject::POPReference
setAccessPoint, 159
popjava::broker::Broker
clearResourceAfterInvoke, 20
findEndcoding, 20
getAccessPoint, 20
getLogPrefix, 20
getPOPObjectClass, 20
getState, 21
initialize, 21
invoke, 21
isDaemon, 22
main, 22
popCall, 22
sendException, 22
sendResponse, 23
serveRequest, 23
setState, 23
treatRequests, 23
popjava::broker::POPThread
getRequest, 169
setRequest, 169
popjava::broker::Request
getBroker, 171
getBuffer, 171
getClassId, 171
getCombox, 172
getMethodId, 172
getReceiveCombox, 172
getSenmatics, 172
getStatus, 172
init, 172
isConcurrent, 173
isMutex, 173
isSequential, 173
isSynchronous, 173
Request, 171
setBroker, 173
setBuffer, 174
setClassId, 174
setCombox, 174
setMethodId, 174
setReceiveCombox, 175
setSenmatics, 175
setStatus, 175
popjava::broker::RequestQueue
add, 176
canPeek, 176
clear, 176
getMaxQueue, 176
peek, 177
remove, 177
setMaxQueue, 177
size, 177
popjava::buffer::BufferFactory
getBufferName, 24
popjava::buffer::BufferFactoryFinder
findFactory, 25
getInstance, 25
getSupportingBuffer, 25
loadBufferMap, 25
popjava::buffer::BufferFactoryPlugin
getBufferName, 26

popjava::buffer::BufferPlugin
extractHeader, 28
get, 28
getBoolean, 28
getBooleanArray, 29
getByteArray, 29
getChar, 29
getCharArray, 29
getDouble, 30
getDoubleArray, 30
getFloat, 30
getFloatArray, 30
getInt, 30
getIntArray, 31
getLong, 31
getLongArray, 31
getShort, 31
getShortArray, 31
getString, 32
getTranslatedInteger, 32
packMessageHeader, 32
put, 32, 33
putBoolean, 33
putBooleanArray, 33
putByteArray, 33
putChar, 33
putCharArray, 34
putDouble, 34
putDoubleArray, 34
putFloat, 34
putFloatArray, 34
putInt, 35
putIntArray, 35
putLong, 35
putLongArray, 35
putShort, 35
putShortArray, 35
putString, 36
popjava::buffer::BufferRaw
BufferRaw, 39
extractHeader, 39
get, 39
getBoolean, 39
getBooleanArray, 39
getByteArray, 40
getChar, 40
getCharArray, 40
getDouble, 40
getDoubleArray, 40
getFloat, 41
getFloatArray, 41
getInt, 41
getIntArray, 42
getLong, 42
getLongArray, 42
getShort, 42
getShortArray, 43
getString, 43
getTranslatedInteger, 43
limit, 43
packMessageHeader, 43
position, 44
put, 44
putBoolean, 45
putBooleanArray, 45
putByteArray, 45
putChar, 45
putCharArray, 46
putDouble, 46
putDoubleArray, 46
putFloat, 46
putFloatArray, 46
putInt, 47
putIntArray, 47
putLong, 47
putLongArray, 47
putShort, 48
putShortArray, 48
putString, 48
resize, 48
popjava::buffer::BufferXDR
BufferXDR, 50
putBoolean, 50
popjava::buffer::POPBuffer
checkAndThrow, 120
deserializeReferenceObject, 120
extractHeader, 121
get, 121
getArray, 121
getBoolean, 121
getBooleanArray, 122
getByteArray, 122
getChar, 122
getCharArray, 122
getDouble, 123
getDoubleArray, 123
getFloat, 123
getFloatArray, 123
getHeader, 124
getInt, 124
getIntArray, 124
getLong, 124
getLongArray, 124
getShort, 125
getShortArray, 125
getString, 125
getTranslatedInteger, 125
getValue, 126
POPBuffer, 120
packMessageHeader, 126
put, 126, 127
putArray, 127
putBoolean, 127
putBooleanArray, 127
putByteArray, 127
putChar, 128

putCharArray, 128
putDouble, 128
putDoubleArray, 128
putFloat, 128
putFloatArray, 129
putInt, 129
putIntArray, 129
putLong, 129
putLongArray, 129
putShort, 130
putShortArray, 130
putString, 130
putValue, 130
serializeReferenceObject, 131
setHeader, 131
size, 131
toCharString, 131
toIntString, 131
popjava::codemanager::POPJavaAppService
getPlatform, 142
queryCode, 143
registerCode, 143
popjava::combox::Combox
Combox, 54
connect, 54
getBufferFactory, 54
receive, 54
send, 55
setBufferFactory, 55
popjava::combox::ComboxAcceptSocket
ComboxAcceptSocket, 56
getStatus, 56
setStatus, 56
popjava::combox::ComboxAllocateSocket
getUrl, 57
receive, 57
send, 57
popjava::combox::ComboxFactory
createClientCombox, 58
createServerCombox, 59
getComboxName, 59
popjava::combox::ComboxFactoryFinder
findFactory, 60
get, 60
getFactoryCount, 61
getInstance, 61
loadComboxMap, 61
popjava::combox::ComboxFactoryPlugin
createClientCombox, 62
createServerCombox, 63
getComboxName, 63
popjava::combox::ComboxPlugin
connect, 64
receive, 64
send, 64
popjava::combox::ComboxReceiveRequestSocket
ComboxReceiveRequestSocket, 66
getStatus, 66
receiveRequest, 66
setBuffer, 66
setStatus, 66
popjava::combox::ComboxServer
ComboxServer, 67
getRequestQueue, 68
popjava::combox::ComboxServerPlugin
ComboxServerPlugin, 68
getRequestQueue, 68
popjava::combox::ComboxServerSocket
ComboxServerSocket, 69
GetUrl, 69
popjava::combox::ComboxSocket
ComboxSocket, 71
connect, 71
receive, 71
send, 71
popjava::combox::ComboxSocketFactory
createClientCombox, 72, 73
createServerCombox, 73
getComboxName, 73
popjava::dataswaper::IPOPBase
deserialize, 80
serialize, 81
popjava::dataswaper::IPOPBaseInput
deserialize, 82
serialize, 82
popjava::dataswaper::ObjectDescriptionInput
getCodeFile, 101
getEncoding, 101
getHostName, 101
getJobUrl, 101
getPlatform, 102
getProtocol, 102
getValue, 102
getWallTime, 102
isEmpty, 102
merge, 102
ObjectDescriptionInput, 101
removeValue, 103
setBandwidth, 103
setCodeFile, 103
setEncoding, 103
setHostname, 103
setJobUrl, 103
setMemory, 104
setPlatform, 104
setPower, 104
setProtocol, 104
setSearch, 104
setValue, 105
setWallTime, 105
popjava::dataswaper::POPString
getValue, 164
POPString, 163
setValue, 164
popjava::interfacebase::Interface
allocate, 77

bind, 78
deserialize, 78
getAccessPoint, 78
getOD, 78
Interface, 77
isAlive, 79
popDispatch, 79
popResponse, 79
serialize, 79
setAccessPoint, 80
setOd, 80
popjava::serviceadapter::POPAppService
POPAppService, 115
queryService, 116
registerService, 116
unregisterService, 116
popjava::serviceadapter::POPCodeManager
getPlatform, 133
POPCodeManager, 133
queryCode, 133
registerCode, 134
popjava::serviceadapter::POPJobManager
allocResource, 146
cancelReservation, 146
createObject, 146
execObj, 147
POPJobManager, 145, 146
query, 147
registerNode, 147
popjava::serviceadapter::POPJobService
createObject, 148
POPJobService, 148
popjava::serviceadapter::POPObjectMonitor
checkObjects, 158
manageObject, 158
POPObjectMonitor, 158
unManageObject, 158
popjava::serviceadapter::POPRemoteLog
log, 160
POPRemoteLog, 160
popjava::serviceadapter::POPServiceBase
POPServiceBase, 162
start, 162
stop, 163
popjava::system::ConfigurationWorker
ConfigurationWorker, 75
getValue, 75
popjava::system::POPJavaConfiguration
getPopAppCoreService, 144
getPopJavaLocation, 144
getPopPluginLocation, 144
popjava::system::POPRemoteLogThread
getFilename, 161
POPRemoteLogThread, 161
run, 161
setRunning, 161
popjava::system::POPSystem
createAppCoreService, 165
getDefaultAccessPoint, 166
getDefaultOD, 166
getEnviroment, 166
getHostIP, 166
getIPAsInt, 166
getPlatform, 167
initCodeService, 167
initialize, 167
popjava::system::XMLWorker
isValid, 183
popjava::util::ClassUtil
getConstructor, 51
getDefaultPrimitiveValue, 51
getMethod, 52
getMethodSign, 52
popjava::util::LogWriter
deleteLogDir, 83
writeDebugInfo, 83
writeExceptionLog, 83
writeLogInfo, 84
writeLogfile, 83
popjava::util::SystemUtil
runCmd, 179
popjava::util::Util
byteArrayToInt, 180
generateRandomString, 180
isLocal, 180
isNoCaseStringEqual, 180
isParameterNotOfDirection, 181
isStringEqual, 181
matchPlatform, 181
removeStringFromArrayList, 181
sameContact, 182
splitTheCommand, 182
position
popjava::buffer::BufferRaw, 44
put
popjava::buffer::BufferPlugin, 32, 33
popjava::buffer::BufferRaw, 44
popjava::buffer::POPBuffer, 126, 127
putArray
popjava::buffer::POPBuffer, 127
putBoolean
popjava::buffer::BufferPlugin, 33
popjava::buffer::BufferRaw, 45
popjava::buffer::BufferXDR, 50
popjava::buffer::POPBuffer, 127
putBooleanArray
popjava::buffer::BufferPlugin, 33
popjava::buffer::BufferRaw, 45
popjava::buffer::POPBuffer, 127
putByteArray
popjava::buffer::BufferPlugin, 33
popjava::buffer::BufferRaw, 45
popjava::buffer::POPBuffer, 127
putChar
popjava::buffer::BufferPlugin, 33
popjava::buffer::BufferRaw, 45

popjava::buffer::POPBuffer, 128
putCharArray
popjava::buffer::BufferPlugin, 34
popjava::buffer::BufferRaw, 46
popjava::buffer::POPBuffer, 128
putDouble
popjava::buffer::BufferPlugin, 34
popjava::buffer::BufferRaw, 46
popjava::buffer::POPBuffer, 128
putDoubleArray
popjava::buffer::BufferPlugin, 34
popjava::buffer::BufferRaw, 46
popjava::buffer::POPBuffer, 128
putFloat
popjava::buffer::BufferPlugin, 34
popjava::buffer::BufferRaw, 46
popjava::buffer::POPBuffer, 128
putFloatArray
popjava::buffer::BufferPlugin, 34
popjava::buffer::BufferRaw, 46
popjava::buffer::POPBuffer, 129
putInt
popjava::buffer::BufferPlugin, 35
popjava::buffer::BufferRaw, 47
popjava::buffer::POPBuffer, 129
putIntArray
popjava::buffer::BufferPlugin, 35
popjava::buffer::BufferRaw, 47
popjava::buffer::POPBuffer, 129
putLong
popjava::buffer::BufferPlugin, 35
popjava::buffer::BufferRaw, 47
popjava::buffer::POPBuffer, 129
putLongArray
popjava::buffer::BufferPlugin, 35
popjava::buffer::BufferRaw, 47
popjava::buffer::POPBuffer, 129
putShort
popjava::buffer::BufferPlugin, 35
popjava::buffer::BufferRaw, 48
popjava::buffer::POPBuffer, 130
putShortArray
popjava::buffer::BufferPlugin, 35
popjava::buffer::BufferRaw, 48
popjava::buffer::POPBuffer, 130
putString
popjava::buffer::BufferPlugin, 36
popjava::buffer::BufferRaw, 48
popjava::buffer::POPBuffer, 130
putValue
popjava::buffer::POPBuffer, 130

query
popjava::serviceadapter::POPJobManager, 147
queryCode
popjava::codemanager::POPJavaAppService, 143
popjava::serviceadapter::POPCodeManager, 133
queryService
popjava::serviceadapter::POPAppService, 116

receive
popjava::combox::Combox, 54
popjava::combox::ComboxAllocateSocket, 57
popjava::combox::ComboxPlugin, 64
popjava::combox::ComboxSocket, 71
receiveRequest
popjava::combox::ComboxReceiveRequestSocket, 66
registerCode
popjava::codemanager::POPJavaAppService, 143
popjava::serviceadapter::POPCodeManager, 134
registerNode
popjava::serviceadapter::POPJobManager, 147
registerService
popjava::serviceadapter::POPAppService, 116
remove
popjava::broker::RequestQueue, 177
removeStringFromArrayList
popjava::util::Util, 181
removeValue
popjava::baseobject::ObjectDescription, 95
popjava::dataswaper::ObjectDescriptionInput, 103
Request
popjava::broker::Request, 171
resize
popjava::buffer::BufferRaw, 48
run
popjava::system::POPRemoteLogThread, 161
runCmd
popjava::util::SystemUtil, 179

sameContact
popjava::util::Util, 182
send
popjava::combox::Combox, 55
popjava::combox::ComboxAllocateSocket, 57
popjava::combox::ComboxPlugin, 64
popjava::combox::ComboxSocket, 71
sendException
popjava::broker::Broker, 22
sendResponse
popjava::broker::Broker, 23
serialize
popjava::base::POPException, 137
popjava::base::POPObject, 156
popjava::baseobject::ODElement, 107
popjava::dataswaper::IPOPBase, 81
popjava::dataswaper::IPOPBaseInput, 82
popjava::interfacebase::Interface, 79
serializeReferenceObject
popjava::buffer::POPBuffer, 131
serveRequest
popjava::broker::Broker, 23
set
popjava::baseobject::ODElement, 107
setAccessPoint
popjava::baseobject::POPReference, 159
popjava::interfacebase::Interface, 80
setAccessString

popjava::baseobject::POPAccessPoint, 114
setBandwidth
popjava::baseobject::ObjectDescription, 95
popjava::dataswaper::ObjectDescriptionInput, 103
setBatch
popjava::baseobject::ObjectDescription, 95
setBroker
popjava::broker::Request, 173
setBuffer
popjava::broker::Request, 174
popjava::combox::ComboxReceiveRequestSocket, 66
setBufferFactory
popjava::combox::Combox, 55
setClassId
popjava::base::MessageHeader, 86
popjava::base::POPObject, 156
popjava::broker::Request, 174
setClassName
popjava::base::POPObject, 156
setCode
popjava::base::BindStatus, 17
setCodeFile
popjava::baseobject::ObjectDescription, 96
popjava::dataswaper::ObjectDescriptionInput, 103
setCombox
popjava::broker::Request, 174
setDirectory
popjava::baseobject::ObjectDescription, 96
setEncoding
popjava::baseobject::ObjectDescription, 96
popjava::dataswaper::ObjectDescriptionInput, 103
setExceptionCode
popjava::base::MessageHeader, 86
setHeader
popjava::buffer::POPBuffer, 131
setHost
popjava::baseobject::AccessPoint, 15
setHostarch
popjava::baseobject::ObjectDescription, 96
setHostcore
popjava::baseobject::ObjectDescription, 96
setHostname
popjava::baseobject::ObjectDescription, 97
popjava::dataswaper::ObjectDescriptionInput, 103
setHostuser
popjava::baseobject::ObjectDescription, 97
setInfo
popjava::base::BindStatus, 18
setJVMParamters
popjava::baseobject::ObjectDescription, 97
setJobUrl
popjava::baseobject::ObjectDescription, 97
popjava::dataswaper::ObjectDescriptionInput, 103
setMaxQueue
popjava::broker::RequestQueue, 177
setMemory
popjava::baseobject::ObjectDescription, 97
popjava::dataswaper::ObjectDescriptionInput, 104
setMethodId
popjava::base::MessageHeader, 87
popjava::broker::Request, 174
setMinValue
popjava::baseobject::ODElement, 107
setOd
popjava::base::POPObject, 157
popjava::interfacebase::Interface, 80
setPlatform
popjava::base::BindStatus, 18
popjava::baseobject::ObjectDescription, 97
popjava::dataswaper::ObjectDescriptionInput, 104
setPort
popjava::baseobject::AccessPoint, 15
setPower
popjava::baseobject::ObjectDescription, 98
popjava::dataswaper::ObjectDescriptionInput, 104
setProtocol
popjava::baseobject::AccessPoint, 15
popjava::baseobject::ObjectDescription, 98
popjava::dataswaper::ObjectDescriptionInput, 104
setReceiveCombox
popjava::broker::Request, 175
setRequest
popjava::broker::POPThread, 169
setRequestType
popjava::base::MessageHeader, 87
setRequiredValue
popjava::baseobject::ODElement, 107
setRunning
popjava::system::POPRemoteLogThread, 161
setSearch
popjava::baseobject::ObjectDescription, 98
popjava::dataswaper::ObjectDescriptionInput, 104
setSenmatics
popjava::base::MessageHeader, 87
popjava::broker::Request, 175
setState
popjava::broker::Broker, 23
setStatus
popjava::broker::Request, 175
popjava::combox::ComboxAcceptSocket, 56
popjava::combox::ComboxReceiveRequestSocket, 66
setValue
popjava::baseobject::ObjectDescription, 98
popjava::dataswaper::ObjectDescriptionInput, 105
popjava::dataswaper::POPString, 164
setWallTime
popjava::baseobject::ObjectDescription, 98
popjava::dataswaper::ObjectDescriptionInput, 105
size
popjava::baseobject::POPAccessPoint, 115
popjava::broker::RequestQueue, 177
popjava::buffer::POPBuffer, 131
splitTheCommand
popjava::util::Util, 182

start
popjava::serviceadapter::POPServiceBase, 162
stop
popjava::serviceadapter::POPServiceBase, 163

throwAccessPointNotAvailableException
popjava::base::POPException, 137
throwBufferFormatException
popjava::base::POPException, 137
throwBufferNotAvailableException
popjava::base::POPException, 138
throwComboxNotAvailableException
popjava::base::POPException, 138
throwNullObjectNotAllowException
popjava::base::POPException, 138
throwObjectBindException
popjava::base::POPException, 139
throwObjectNoResource
popjava::base::POPException, 139
throwReflectException
popjava::base::POPException, 139
throwReflectMethodNotFoundException
popjava::base::POPException, 139
throwReflectSerializeException
popjava::base::POPException, 139
toCharString
popjava::buffer::POPBuffer, 131
toIntString
popjava::buffer::POPBuffer, 131
treatRequests
popjava::broker::Broker, 23

unManageObject
popjava::serviceadapter::POPObjectMonitor, 158
unregisterService
popjava::serviceadapter::POPAppService, 116

writeDebugInfo
popjava::util::LogWriter, 83
writeExceptionLog
popjava::util::LogWriter, 83
writeLogInfo
popjava::util::LogWriter, 84
writeLogfile
popjava::util::LogWriter, 83