
OZ Query Designer

User Help



Table of Contents

Part I. Welcome to 오즈 쿼리 디자이너	1
Part II. 스토어	2
1. 데이터베이스 스토어	3
스토어 추가	3
데이터베이스 연결 정보 설정	3
연결 정보 직접 설정	3
앨리어스로 설정	5
사용자 지정 패러미터로 설정	5
데이터베이스 초기화/종료 SQL 설정	6
데이터 셋 추가	6
2. NoSQL 스토어	7
스토어 추가	7
데이터 셋 추가	7
3. 사용자 데이터 스토어	8
스토어 추가	9
데이터셋 추가	9
사용자 필드 추가	9
4. 그룹 데이터 스토어	11
스토어 추가	11
데이터 셋 추가	11
데이터 셋 분할	12
수직 분할	12
수평 분할	14
5. 파일 스토어	16
데이터 셋 추가	16
XML 데이터 셋 추가	16
CSV 데이터 셋 추가	17
다이나믹 XML 데이터 셋 추가	18
다이나믹 CSV 데이터 셋 추가	20
파일 스토어에서 사용되는 파일 규격	22
XML 파일 규격	22
CSV 파일 규격	23
6. HTTP 스토어	24
데이터 셋 추가	24

XML 데이터 셋 추가	24
CSV 데이터 셋 추가	25
다이나믹 XML 데이터 셋 추가	27
다이나믹 CSV 데이터 셋 추가	29
7. SOAP 스토어	32
스토어 추가	32
데이터 셋 추가	33
디자인 뷰 모드	33
매뉴얼 모드	34
8. ADO DataSet 웹서비스 스토어	36
스토어 추가	36
데이터 셋 추가	36
9. XML 스토어	38
스토어 추가	38
데이터 셋 추가	40
디자인 뷰 모드	40
매뉴얼 모드	41
10. SAP 스토어	43
스토어 추가	43
데이터 셋 추가	44
11. CLEAR QUEST 스토어	45
스토어 추가	45
데이터 셋 추가	46
Part III. 데이터셋 디자인	48
1. 매뉴얼 모드로 쿼리 디자인	49
2. 디자인 모드로 쿼리 디자인	50
테이블 선택	50
조인 생성	50
필드 설정	51
정렬 선택	51
쿼리문 확인	52
쿼리문 실행	52
3. 마스터-디테일 데이터 셋 디자인	53
4. 다이나믹 데이터 셋 디자인	54
5. 멀티 데이터 셋 디자인	57
6. 프로시저 데이터 셋 디자인	58
7. 사용자 지정 패러미터 추가	59

8. 수식 필드 추가	60
9. INCLUDE ODI로 데이터 셋 디자인	61
10. 컴파일된 질의문 디자인	62
Part IV. 오즈 쿼리 컴포넌트	64
1. 오즈 데이터 트리	65
널 데이터 무시	65
사용자 지정 패러미터 구분자	65
Concurrent 전송 크기	66
서버 연결 위치	66
데이터서비스 허용	68
소수점 앞에 0 표시	68
스크립트 에러 무시	69
2. 오즈 데이터 스토어	70
데이터베이스 스토어	70
이름	71
사용자 이름	71
JDBC 드라이버	71
앨리어스 정보파일 사용	71
앨리어스 이름	71
앨리어스 정보파일 이름	72
자동 커밋	72
벤더	72
데이터 액션 클래스	72
데이터 액션 초기화 패러미터	73
사용자 데이터 스토어	73
이름	73
클래스 이름	73
필드 정의 사용	73
그룹 데이터 스토어	74
이름	74
원본셋	74
SOAP 스토어	74
이름	75
ADO DataSet 웹서비스 스토어	75
이름	75
XML 스토어	75
이름	75

SAP 스토어	75
클라이언트 이름	76
언어	76
호스트 이름(주소)	76
시스템 번호	76
데이터 액션 클래스	77
데이터 액션 초기화 패러미터	77
SID	77
CLEAR QUEST 스토어	77
이름	77
3. 오즈 데이터 셋	78
데이터베이스 데이터 셋	78
이름	79
질의문	79
마스터셋	79
최대행 수	80
컴파일된 질의문 사용	80
JDBC 전송 ROW	80
스크립트 사용	81
숨김	81
Concurrent 전송 크기	81
Concurrent 첫 전송 ROW	82
바인딩 전 데이터 보내기	82
바인딩 후 데이터 보내기	83
쿼리 타임아웃	83
쿼리 에러 무시	83
설명	84
다이나믹 데이터 셋	84
이름	85
마스터셋	85
최대행 수	85
시작 행 번호	85
숨김	85
Concurrent 전송 크기	86
Concurrent 첫 전송 ROW	86
컴파일된 SQL문 사용 (원본 셋)	87
스크립트 사용 (원본 셋)	87

바인딩 전 데이터 보내기	87
바인딩 후 데이터 보내기	88
쿼리 타임아웃	88
쿼리 에러 무시	88
설명	89
멀티 데이터 셋	89
이름	89
질의문	89
마스터셋	89
컴파일된 질의문 사용	90
스크립트 사용	90
쿼리 타임아웃	91
프로시저 데이터 셋	91
이름	91
프로시저 이름	92
프로시저 타입	92
타겟 패러미터	92
마스터셋	93
쿼리 타임아웃	93
사용자 데이터 셋	93
이름	94
실행문	94
마스터셋	94
최대행 수	95
스크립트 사용	95
숨김	95
Concurrent 전송 크기	96
Concurrent 첫 전송 ROW	96
바인딩 전 데이터 보내기	96
바인딩 후 데이터 보내기	97
설명	97
그룹 데이터 셋	98
이름	98
실행문	98
마스터셋	99
Consume Row	99
최대행 수	99

숨김	100
Concurrent 전송 크기	100
Concurrent 첫 전송 ROW	101
바인딩 전 데이터 보내기	101
바인딩 후 데이터 보내기	101
설명	102
파일 스토어 XML 데이터 셋	102
이름	103
마스터셋	103
최대행 수	103
데이터 셋 노드 이름	103
레코드셋 노드 이름	104
숨김	104
Concurrent 전송 크기	104
Concurrent 첫 전송 ROW	105
바인딩 전 데이터 보내기	105
바인딩 후 데이터 보내기	106
설명	106
파일 스토어 CSV 데이터 셋	106
이름	107
마스터셋	107
최대행 수	107
필드 구분자	108
라인 구분자	108
문자 셋	108
널 마크	109
숨김	109
Concurrent 전송 크기	109
Concurrent 첫 전송 ROW	110
바인딩 전 데이터 보내기	110
바인딩 후 데이터 보내기	111
설명	111
HTTP 스토어 XML 데이터 셋	111
이름	112
마스터셋	112
최대행 수	112
데이터 셋 노드 이름	112

레코드셋 노드 이름	113
숨김	113
Concurrent 전송 크기	113
Concurrent 첫 전송 ROW	114
바인딩 전 데이터 보내기	114
바인딩 후 데이터 보내기	115
설명	115
HTTP 스토어 CSV 데이터 셋	115
이름	116
마스터셋	116
최대행 수	116
필드 구분자	117
라인 구분자	117
문자 셋	117
널 마크	117
숨김	118
Concurrent 전송 크기	118
Concurrent 첫 전송 ROW	118
바인딩 전 데이터 보내기	119
바인딩 후 데이터 보내기	119
설명	120
SOAP 데이터 셋	120
이름	120
마스터셋	121
숨김	121
Concurrent 전송 크기	121
Concurrent 첫 전송 ROW	121
바인딩 전 데이터 보내기	122
바인딩 후 데이터 보내기	122
기준 XML 경로	122
설명	123
ADO DataSet 웹서비스 데이터 셋	123
이름	123
마스터셋	123
EndPoint	123
SOAPAction	123
XML 데이터 셋	124

이름	124
마스터셋	124
숨김	125
Concurrent 전송 크기	125
Concurrent 첫 전송 ROW	125
바인딩 전 데이터 보내기	126
바인딩 후 데이터 보내기	126
설명	127
SAP 데이터 셋	127
이름	127
데이터셋	128
숨김	128
Concurrent 전송 크기	128
Concurrent 첫 전송 ROW	129
바인딩 전 데이터 보내기	129
바인딩 후 데이터 보내기	129
설명	130
CLEAR QUEST 데이터 셋	130
이름	131
타입	131
마스터셋	131
최대행 수	131
스크립트 사용	131
숨김	132
Concurrent 전송 크기	132
Concurrent 첫 전송 ROW	132
바인딩 전 데이터 보내기	133
바인딩 후 데이터 보내기	133
설명	133
4. 오즈 데이터 필드	134
데이터 셋 필드	134
이름	134
타입	134
복호화	134
설명	135
SOAP 데이터 셋 필드	135
이름	135

타입	135
XPath	136
복호화	136
설명	136
XML 데이터 셋 필드	136
이름	137
타입	137
XPath	137
복호화	137
설명	138
5. 사용자 지정 패러미터	139
이름	139
타입	139
값	139
암호화	139
세션 키	140
설명	140
6. Include 필드	141
이름	141
OZURL	141
Part V. 자바 스크립트	142
1. 스크립트 입력, 비교	143
스크립트 입력	143
스크립트 비교	144
비교할 파일 및 버전 선택	144
스크립트 비교	145
스크립트 팝업 헬프 창	146
2. 자바 스크립트 문법	147
3. 자바 스크립트 API	149
OZObject	149
DataModule	149
Property	149
ConcurrentFetchSize	149
OZParamMark	150
ShowZeroDecimalPoint	150
Method	151
GetClientDataModuleType	151

GetDataSet	152
GetDataSetNames	152
GetDataStore	153
GetDataStoreNames	153
GetFetchType	153
GetFetchUnit	154
GetFileName	154
GetOZParam	154
GetOZParamNames	155
GetServerDataModuleType	155
GetServerType	156
IsCachedDataRequest	156
IsCompressedDataRequest	156
IsDesignTime	157
IsEnableDataSet	157
Event	158
Functions	158
OnPostBind	159
OnPostDataAction	159
OnPreBind	159
OnPreDataAction	159
DataStore	159
OZDBStore	160
Property	160
AutoCommit	160
ConnectionPoolAlias	160
DecodeCharaterSet	161
EncodeCharacterSet	161
UseConnectionPoolAlias	161
VendorName	162
Method	162
GetConnection	163
GetConnectionKey	163
GetConnectionKeyItemCount	163
GetConnectionKeyItemNames	164
GetConnectionKeyItemValue	164
GetDataModule	164

GetName	165
SetConnectionKeyItem	165
Event	166
OnFinalize	166
OnInitialize	166
OnPostInitialize	166
OnTransactionCommit	167
OnTransactionRollback	167
OnTransactionStart	167
OZGroupDataStore	167
Property	168
SourceDataSetName	168
Method	168
GetDataModule	168
GetName	169
Event	169
OnFinalize	169
OnInitialize	169
OnPostInitialize	170
OZSAPStore	170
Property	170
ClientName	171
ConnectionPoolAlias	171
FunctionList	171
HostName	172
Language	172
SID	172
SystemNumber	173
UseConnectionPoolAlias	173
UserID	174
UserPassword	174
Method	174
GetDataModule	174
GetName	175
Event	175
OnFinalize	175
OnInitialize	175

OnPostInitialize	176
OZSOAPStore	176
Property	176
EndPoint	176
SOAPAction	177
TargetNamespace	177
WSDLURL	177
Method	177
GetDataModule	178
GetName	178
GetOperationName	178
GetPortName	179
GetServiceName	179
Event	179
OnFinalize	180
OnInitialize	180
OnPostInitialize	180
OZUserDataStore	180
Property	180
ClassName	181
Method	181
GetDataModule	181
GetName	181
Event	182
OnFinalize	182
OnInitialize	182
OnPostInitialize	183
OnTransactionCommit	183
OnTransactionRollback	183
OnTransactionStart	183
OZXPathStore	184
Property	184
DoPost	184
EncodingURL	184
SchemaURL	185
TestXMLURL	185
XMLParser	185

XMLURL	186
Method	186
GetDataModule	186
GetName	187
Event	187
OnFinalize	187
OnInitialize	187
OnPostInitialize	188
DataSet	188
OZCSVDataSet	188
Property	188
CharacterSet	189
ColumnDelimiter	190
ConcurrentFetchSize	190
ConcurrentFirstRow	191
ConnectionRetryCount	191
DoPost	191
EncodingURL	192
FlushDataAfterBinding	192
FlushDataBeforeBinding	193
HasFieldType	193
Hidden	193
HTTPLogFileName	194
MaxRowCount	194
NullMark	194
ParseErrorCode	195
RowDelimiter	195
URL	195
Method	196
GetCalculatedFieldNames	196
GetDataFieldNames	197
GetDataModule	197
GetDataStore	197
GetDetailDataSets	198
GetField	198
GetMasterDataSetName	198
GetName	199

IsFileStore	199
IsHTTPStore	200
IsMasterDataSet	200
Event	201
OnFinalize	201
OnInitialize	201
OZDBDataSet	201
Property	201
ConcurrentFetchSize	202
ConcurrentFirstRow	202
FlushDataAfterBinding	203
FlushDataBeforeBinding	203
Hidden	204
IgnoreQueryError	204
JDBCFetchRow	205
MaxRowCount	205
QueryString	205
QueryTimeout	206
TargetParameter	206
UseCompiledSQL	207
UseOZScript	207
Method	208
GetCalculatedFieldNames	209
GetDataFieldNames	209
GetDataModule	209
GetDestinationFields	210
GetDestinationValue	210
GetDestinationValues	212
GetDataStore	212
GetDetailDataSets	213
GetDrivenDataSetIndex	213
GetField	213
GetName	214
GetMultiDataSet	214
GetMasterDataSetName	215
GetProcedureDataSet	215
GetSourceFields	215

GetSourceValue	216
GetSourceValues	218
IsMasterDataSet	218
Event	219
OnFinalize	219
OnInitialize	219
OnPostBind	219
OnPreBind	219
OZDBMultiDataSet	220
Property	220
QueryString	220
QueryTimeout	221
UseCompiledSQL	221
UseOZScript	222
Method	222
GetDataModule	222
GetDataStore	223
GetDrivenDataSets	223
GetMasterDataSetName	223
GetName	224
Event	224
OnFinalize	224
OnInitialize	224
OnPostBind	225
OnPreBind	225
OZDBProcedureDataSet	225
Property	225
ProcedureName	226
ProcedureType	226
QueryTimeout	226
Method	227
AddProcedureParameter	228
GetDataModule	228
GetDataStore	228
GetDrivenDataSets	229
GetMasterDataSetName	229
GetName	229

GetProcedureParameter	230
GetProcedureParameterNames	230
SetProcedureParameter	231
RemoveProcedureParameter	231
RemoveAllProcedureParameter	231
Event	232
OnFinalize	232
OnInitialize	232
OnPostBind	232
OnPreBind	233
OZDynamicDataSet	233
Property	233
ConcurrentFetchSize	234
ConcurrentFirstRow	234
FlushDataAfterBinding	235
FlushDataBeforeBinding	235
Hidden	235
IgnoreQueryError	235
MaxRowCount	236
QueryTimeout	236
SourceDataSetName	237
StartRowIndex	237
Method	237
GetCalculatedFieldNames	238
GetDataFieldNames	238
GetDataModule	238
GetDataStore	239
GetDetailDataSets	239
GetField	239
GetMasterDataSetName	240
GetName	240
IsMasterDataSet	241
Event	241
OnFinalize	241
OnInitialize	241
OZGroupDataSet	242
Property	242

Command	242
ConcurrentFetchSize	243
ConcurrentFirstRow	243
FlushDataAfterBinding	244
FlushDataBeforeBinding	244
Hidden	244
MaxRowCount	245
Method	245
GetDataFieldNames	246
GetDataModule	246
GetDataStore	246
GetDetailDataSets	247
GetField	247
GetMasterDataSetName	247
GetName	248
IsMasterDataSet	248
Event	249
OnFinalize	249
OnInitialize	249
OZSAPDataSet	249
Property	250
ConcurrentFetchSize	250
ConcurrentFirstRow	250
FlushDataAfterBinding	251
FlushDataBeforeBinding	251
FunctionName	252
FunctionType	252
Hidden	252
MappingKey	252
Method	253
GetCalculatedFieldNames	253
GetDataFieldNames	254
GetDataModule	254
GetDataStore	254
GetDetailDataSets	255
GetField	255
GetInputParam	255

GetInputParamNames	256
GetName	256
IsMasterDataSet	256
SetInputParam	257
Event	257
OnFinalize	257
OnInitialize	258
OZUserDataSet	258
Property	258
Command	259
ConcurrentFetchSize	259
ConcurrentFirstRow	259
FlushDataAfterBinding	260
FlushDataBeforeBinding	260
Hidden	261
UseOZScript	261
Method	261
GetCalculatedFieldNames	262
GetDataFieldNames	262
GetDataModule	263
GetDestinationFields	263
GetDestinationValue	263
GetDestinationValues	265
GetDataStore	266
GetDetailDataSets	266
GetField	266
GetMasterDataSetName	267
GetName	267
GetSourceFields	268
GetSourceValue	268
GetSourceValues	270
IsMasterDataSet	271
Event	271
OnFinalize	271
OnInitialize	271
OnPostBind	272
OnPreBind	272

OZXMLDataSet	272
Property	272
ConcurrentFetchSize	273
ConcurrentFirstRow	274
ConnectionRetryCount	274
DataSetNodeName	274
DoPost	275
EncodingURL	275
FlushDataAfterBinding	276
FlushDataBeforeBinding	276
Hidden	276
HTTPLogFileName	277
MaxRowCount	277
RecordNodeName	277
URL	278
Method	278
GetCalculatedFieldNames	279
GetDataFieldNames	279
GetDataModule	279
GetDataStore	280
GetDetailDataSets	280
GetField	280
GetMasterDataSetName	281
GetName	281
IsFileStore	282
IsHTTPStore	282
IsMasterDataSet	283
Event	283
OnFinalize	283
OnInitialize	283
OZXPathDataSet	284
Property	284
BaseXPath	284
BaseXPathCondition	285
ConcurrentFetchSize	285
ConcurrentFirstRow	285
FlushDataAfterBinding	286

FlushDataBeforeBinding	286
Hidden	287
Method	287
GetDataFieldNames	287
GetDataModule	288
GetDataStore	288
GetDetailDataSets	288
GetField	289
GetMasterDataSetName	289
GetName	290
IsMasterDataSet	290
Event	291
OnFinalize	291
OnInitialize	291
DataField	291
OZCalculatedField	291
Method	292
GetDataModule	292
GetDataSetName	292
GetDataSetRowIndex	293
GetDataSetValue	293
GetFieldName	294
GetFieldType	295
IsCalculatedField	295
IsDataSetValueNull	295
IsDecrypt	296
IsNull	296
SetFieldData	297
Event	297
GetFieldData	297
OZDataField	298
Method	298
GetFieldName	298
GetFieldType	299
GetValue	299
IsCalculatedField	299
IsDecrypt	300

IsNull	300
OZXPathDataField	300
Property	301
BaseXPathCondition	301
UseOZScript	301
Method	301
GetFieldName	302
GetFieldType	302
GetXPath	302
IsCalculatedField	303
IsDecrypt	303
IsNull	303
GeneralScript	304
Global	304
Method	304
_DEBUG	304
_ERROR	305
_INFO	305
OZBinary	306
Method	306
Base64decode	306
Base64encode	307
GetName	307
ReadFromFile	307
ReadFromHTTP	308
SaveToFile	308
SetName	309
ToString	309
OZCreator	309
Method	309
GetBuiltInObject	310
OZDBConnection	310
Method	310
AddItemValue	311
Close	312
Commit	313
Connect	314

CreateStatement	315
GetAutoCommit	318
IsClosed	319
PrepareStatement	319
Rollback	322
SetAlias	323
SetAutoCommit	324
SetVendor	325
OZJSPreparedStatement	326
Method	326
ClearParameters	327
Close	327
ExecuteQuery	328
ExecuteUpdate	329
SetFetchRow	329
SetMaxRows	330
SetParameter	331
SetQueryTimeout	333
OZJSResultSet	334
Method	334
Close	334
GetObject	335
GetString	339
GetMetaData	340
Next	341
OZJSResultSetMetaData	342
Method	342
GetColumnCount	342
GetColumnName	343
GetColumnType	344
OZJSStatement	344
Method	344
Close	345
ExecuteQuery	346
ExecuteUpdate	347
SetFetchRow	348
SetMaxRows	349

SetQueryTimeout	350
OZMail	350
Method	351
AddAttachmentPart	351
EnableSSL	352
PrefixSubjectMessage	353
Send	354
SendRetryCount	355
SendRetryPeriodTime	355
SendSync	356
SetAlias	357
SetContext	358
SetFrom	359
SetRecipients	360
SetSmtpID	361
SetSmtpPort	361
SetSmtpPW	362
SetSmtpServer	363
SetSubject	364
OZRepository	365
Method	365
AddCategory	367
AddGroupAdmin	368
CheckIn	369
CheckOut	373
CheckUserPassword	377
ClearHistoryItem	377
CreateGroup	378
CreateUser	380
DeleteCategory	381
DeleteGroup	382
DeleteItem	383
DeleteItems	384
DeleteUser	385
DownloadFile	386
DownloadFileByVersion	389
GetAccessibleTopNodePath	390

GetCategoryListInCategory	391
GetCheckOutFolder	392
GetCheckOutUserName	393
GetErrorMessageByCode	394
GetGroupAdminListInGroup	394
GetGroupAuthToCategory	395
GetGroupAuthToItem	396
GetGroupListAuthToCategory	397
GetGroupListAuthToItem	398
GetGroupListInGroup	399
GetGroupPathOfUser	400
GetHistoryItemList	401
GetItemListInCategory	402
GetLastDetailErrorMessage	403
GetLastErrorCode	403
GetUserAuthToCategory	404
GetUserAuthToItem	405
GetUserListAuthToCategory	406
GetUserListAuthToItem	407
GetUserListInGroup	408
IsFileExist	409
Login	410
LoginToServer	410
Logout	411
RemoveGroupAdmin	412
RenameCategory	413
RenamelItem	413
RollbackItem	415
SearchItems	416
UndoCheckOut	417
UpdateCategoryGroupAuth	420
UpdateCategoryUserAuth	421
UpdateGroupName	422
UpdateItemGroupAuth	422
UpdateItemUserAuth	423
UpdateUserDescription	424
UpdateUserName	425

UpdateUserPassword	426
UploadFile	427
OZProcedureParameter	431
Property	431
Name	431
DataType	431
DataTypeName	432
Direction	432
IsNull	433
Selected	433
Value	433
Constant	433
DataFieldTypeConst	434
ODIClientDMTypeConst	435
ODIFetchTypeConst	435
ODIFetchUnitConst	435
ODIServerDMTypeConst	436
OZJSResultSetConst	436
OZServerTypeConst	436
ProcedureDirectionConst	436
ProcedureTypeConst	437
RecipientTypeConst	437
SearchOptionConst	437
ShowZeroDecimalPointConst	438

Part VI. Reference 439

1. 디자이너 환경 설정	440
작업 폴더 설정	440
디자이너 기본 옵션 설정	441
데이터베이스 옵션 설정	442
편집기 옵션 설정	443
2. 리파지토리 서버 사용	445
리파지토리 서버 설정	445
리파지토리 서버 연결, 연결 해제	445
리파지토리 서버로부터 파일 열기	445
카테고리, 아이템 관리	446
체크인, 체크아웃	447
히스토리 보기	448

아이템 찾기	449
3. 파일 경로 설정	450
4. 문자셋과 유효한 앨리어스	451

Welcome to 오즈 쿼리 디자이너

최강의 쿼리 디자인 툴을 사용하게 되신 것을 환영합니다.

오즈 쿼리 디자이너는 Oracle, MSSQL 등 상용 데이터베이스뿐만 아니라 TXT, XML 파일 등 다양한 데이터 소스에 연결하여 쿼리문을 작성하고 데이터셋을 생성하기 위한 쿼리 디자인 툴입니다.

오즈 쿼리 디자이너는 오즈 리포트에서 연동할 데이터에 대한 정보를 추출하는 용도로 사용되며, 복잡한 쿼리 작성을 쉽고 빠르게 구현함으로써 개발 생산성을 높이고 고품질의 서비스를 제공할 수 있으며, 효율적인 관리로 유지 보수 및 관리 효율성을 증대할 수 있습니다.

스토어

본 장에서는 스토어와 데이터셋을 추가하는 방법에 대해 설명합니다.

데이터베이스 스토어

데이터베이스 스토어는 JDBC 또는 ODBC를 지원하는 데이터베이스로부터 SQL 쿼리를 이용해 데이터 셋을 생성하는 기능을 제공합니다.

스토어 추가

1. 데이터베이스 스토어를 추가하려면 데이터 창 의 **[데이터베이스]**를 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 **[스토어 추가]** 메뉴를 클릭합니다.
2. 데이터베이스 연결 정보 추가 다이얼로그의 각 옵션을 설정한 후 **[확인]** 버튼을 클릭하면 추가한 데이터베이스 정보가 표시됩니다.

데이터베이스 연결 정보 설정

데이터베이스 연결 정보는 데이터베이스 연결 정보 추가 다이얼로그에서 설정할 수 있으며, 데이터베이스 서버의 IP 주소 또는 포트 번호 등을 직접 설정하는 방법과 앨리어스 이름과 앨리어스 파일을 이용하여 설정하는 방법, 사용자 지정 패러미터를 이용하여 설정하는 방법 등이 있습니다.

연결 정보 직접 설정

데이터베이스의 팝업 메뉴 중 [스토어 추가] 메뉴를 선택하여 데이터베이스 연결 정보 추가 다이얼로그를 실행합니다.

데이터베이스 연결 정보 추가 다이얼로그에서 벤더 이름, 데이터베이스 서버 IP 주소, 포트 번호, 사용자 이름, 패스워드 등을 직접 설정합니다.

ODBC 연결

ODBC를 이용한 연결은 ODBC 드라이버를 지원하는 MS Access 등 데이터베이스와 연결할 때 사용할 수 있습니다. ODBC를 사용하기 위해서는 해당 드라이버가 ODBC 데이터 원본 관리자 다이얼로그의 시스템 데이터 원본으로 등록되어있어야 합니다.

데이터베이스 연결 정보 추가 다이얼로그에서 "데이터 정보 이름"은 오즈 쿼리 디자이너가 데이터베이스 연결 정보를 구분하는 고유 이름으로 임의로 설정하며, "데이터베이스 벤더 이름"은 odbc로 선택하고 "데이터베이스 드라이버 종류"에 ODBC 데이터 원본 관리자 다이얼로그에 등록된 ODBC 데이터 원본 이름을 입력합니다.

DBMS 연결

MS-SQL, Oracle, Sybase, Informix, Uni-SQL, DB2, tiber, tiber5, mariadb 등의 연결 정보를 설정합니다.

"데이터 정보 이름"은 오즈 쿼리 디자이너가 데이터베이스 연결 정보를 구분하는 고유 이름으로 개발자가 임의로 부여하며, "데이터베이스 벤더 이름"은 mssql, oracle, Sybase, Informix, unisql, db2, tiber, tiber5, mariadb 등으로 선택합니다.

"데이터베이스 서버 IP 주소", "데이터베이스 서버 포트 번호", "데이터베이스 이름", "데이터베이스 서버 이름" 등에 연결할 데이터베이스 서버의 IP, 포트 번호, 데이터베이스 이름 등의 기본 정보를 입력합니다.

"사용자 이름", "패스워드"에 연결할 데이터베이스의 사용자 이름과 패스워드를 입력합니다. "패러미터 사용" 옵션을 체크하면 사용자 패스워드를 사용자 지정 패러미터 값으로 설정합니다.

sybase 연결 시 "데이터베이스 서버 드라이버 종류"에는 sybase의 JDBC 드라이버인 jConnect를 사용하기 위해 "Tds"를 입력합니다. Tds(Tabular Data Stream)는 Adative Server를 위한 Sybase의 네이티브 통신 프로토콜입니다.

OLEDDB 연결

OLEDDB 연결은 연결되는 서버가 닷넷일 경우에만 지원됩니다.

MSSQL-OLEDDB, ORACLE-OLEDDB, SYBASE-OLEDDB, DB2-OLEDDB, INFORMIX-OLEDDB 등의 연결 정보를 설정합니다.

"데이터 정보 이름"은 오즈 쿼리 디자이너가 데이터베이스 연결 정보를 구분하는 고유 이름으로 개발자가 임의로 부여하며, "데이터베이스 벤더 이름"은 mssql-oledb, oracle-oledb, sybase-oledb, db2-oledb, informix-oledb 등으로 선택합니다.

"데이터베이스 서버 IP 주소", "데이터베이스 서버 포트 번호", "데이터베이스 이름", "데이터베이스 서버 이름" 등에 연결할 데이터베이스 서버의 IP, 포트 번호, 데이터베이스 이름 등의 기본 정보를 입력합니다.

"사용자 이름", "패스워드"에 연결할 데이터베이스의 사용자 이름과 패스워드를 입력합니다. "패러미터 사용" 옵션을 체크하면 사용자 패스워드를 사용자 지정 패러미터 값으로 설정합니다.

User-Definded 연결

User-Definded 연결은 오즈에서 기본으로 제공되지 않는 데이터베이스인 경우 각각의 JDBC드라이버 정보를 사용자가 임의로 설정하여 연결하는 것으로 JDBC의 드라이버 정보와 Connection URL을 직접 입력합니다.

"데이터 정보 이름"은 오즈 쿼리 디자이너가 데이터베이스 연결 정보를 구분하는 고유 이름으로 개발자가 임의로 부여하며, "데이터베이스 벤더 이름"은 user로 선택하고 "드라이버 클래스 이름", "연결 URL"에 연결할 데이터베이스의 JDBC의 드라이버 정보와 Connection URL 정보를 입력합니다.

"사용자 이름", "패스워드"에 연결할 데이터베이스의 사용자 이름과 패스워드를 입력합니다. "패러미터 사용" 옵션을 체크하면 사용자 패스워드를 사용자 지정 패러미터 값으로 설정합니다.

앨리어스로 설정

데이터베이스 연결 정보를 설정할 때 데이터베이스 서버의 IP 주소 또는 포트 번호 등을 데이터베이스 연결 정보 다이얼로그에 바로 입력하지 않고 db.properties 파일에 데이터베이스 서버 접속 정보를 설정한 후 데이터베이스 연결 정보 다이얼로그에서는 앨리어스 이름과 앨리어스 정보 파일만 설정할 수 있습니다.

데이터베이스 연결 정보를 앨리어스로 설정하면 데이터베이스 서버 정보가 변경되었을 때 ODI 파일을 일일이 열어서 데이터베이스 서버 정보를 변경해야 하는 번거로움을 줄일 수 있습니다.

데이터베이스 연결 정보를 앨리어스로 설정하기 위해서는 db.properties 파일에 데이터베이스 벤더 별로 서버 정보를 입력한 후 데이터베이스 연결 정보 다이얼로그에서 "앨리어스" 옵션을 체크한 후 앨리어스 이름을 입력하고 앨리어스 정보 파일의 경로를 입력합니다.

예를 들어 MS-SQL 데이터베이스 연결 정보를 다음과 같이 설정하여 db.properties 파일로 저장한 경우

```
Sample.vendor=mssql
Sample.serverAddress=127.0.0.1
Sample.portNo=1433
Sample.dbName=Foodmart
Sample.user=user1
Sample.password=user1password
Sample.maxconns=20
Sample.initconns=5
Sample.timeout=5
```

데이터베이스 연결 정보 추가 다이얼로그에서 앨리어스를 체크한 후 db.properties에 설정한 앨리어스 이름을 입력하고 앨리어스 정보 파일에 db.properties 파일의 경로를 설정합니다.

참고 사항

db.properties 파일에 설정해야 할 데이터베이스 정보는 ODBC, MS-SQL, DB2, Oracle 등 데이터베이스 벤더 별로 다릅니다.

각 벤더 별로 설정해야 할 옵션 및 db.properties 샘플 파일은 오즈 서버가 설치된 경로의 /conf/sample 폴더에 있으며, 각 옵션에 대한 설명은 OZ Server Administrator's Guide를 참고하시기 바랍니다.

사용자 지정 패러미터로 설정

데이터베이스 서버 접속 정보 또는 앨리어스 이름을 고정된 값이 아니라 동적으로 설정할 경우에는 사용자 지정 패러미터를 이용하여 설정합니다.

접속 정보 설정

오즈 데이터 트리에서 사용자 지정 패러미터의 팝업 메뉴 중 **[패러미터 설정]** 메뉴를 클릭합니다.

사용자 지정 패러미터 다이얼로그에서 패러미터 이름과 필드 타입, 패러미터 값 등을 설정한 후 **[확인]** 버튼을 클릭하여 패러미터를 추가합니다.

데이터베이스 연결 정보 추가 다이얼로그에서 서버 IP 주소, 서버 포트 번호, 사용자 이름 등 서버 접속 정보를 #OZParam.사용자지정패러미터#와 같은 형식의 사용자 지정 패러미터를 이용하여 설정합니다.

참고 사항

각 패러미터 값은 운영 시점에 사용자 애플리케이션으로부터 전달되며, 전달된 정보에 따라 동적으로 데이터베이스에 연결됩니다.

패러미터를 이용한 동적 데이터베이스 연결의 경우 중요한 데이터베이스 연결 정보가 유출될 우려가 있으므로 db.properties 파일을 이용하여 앨리어스 이름을 동적으로 설정하는 방법을 권장합니다.

데이터베이스 초기화/종료 SQL 설정

데이터베이스 연결 후 바로 실행할 SQL문과 데이터베이스 연결 해제 전에 실행할 SQL문을 설정합니다.

데이터베이스 연결 정보 추가 다이얼로그에서 SQL문 이름 지정 옵션을 체크한 후 이름을 입력하고, SQL문을 입력한 뒤 **[추가]** 버튼을 클릭합니다.

스크립트 사용 시 스크립트 사용 옵션을 체크하시기 바랍니다.

참고 사항

SQL문은 JDBC에서 실행 가능한 표준 SQL문으로 설정하여야 하며, 결과셋에 대한 처리를 하지 않으므로 결과셋이 지원되지 않는 SQL(INSERT, UPDATE, DELETE)을 사용하여야 합니다. SQL문은 SQL 목록에 표시된 순서대로 실행됩니다.

데이터 셋 추가

1. 데이터 셋을 추가하려면 추가된 스토어를 마우스 오른쪽 버튼으로 클릭하면 나타나는 팝업 메뉴에서 **[데이터 셋 추가]** 메뉴를 클릭합니다.
2. 데이터 셋 추가 다이얼로그에서 데이터 셋 이름을 입력한 후 **[확인]** 버튼을 클릭하면 스토어 하위에 데이터 셋이 추가된 것을 확인할 수 있습니다.

NoSQL 스토어

NoSQL은 MongoDB 등 JSON 형태의 동적 스키마형 구조인 비관계형 데이터베이스를 이용하여 데이터 셋을 생성하는 기능을 제공합니다.

스토어 추가

1. NoSQL 스토어를 추가하려면 데이터 창 의 **[NoSQL]**을 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 **[스토어 추가]** 메뉴를 클릭합니다.
2. NoSQL 데이터베이스 연결 정보 추가 다이얼로그의 각 옵션을 설정한 후 **[다음]** 버튼을 클릭합니다.
3. Collection을 조회할 수 있는 화면에서 **[조회]** 버튼을 클릭하여 리스트에서 Collection을 클릭하고 **[다음]** 버튼을 클릭합니다.
4. find, aggregate Method에 따라 설정할 패러미터를 입력하고 **[미리보기]** 또는 **[마침]** 버튼을 클릭합니다.

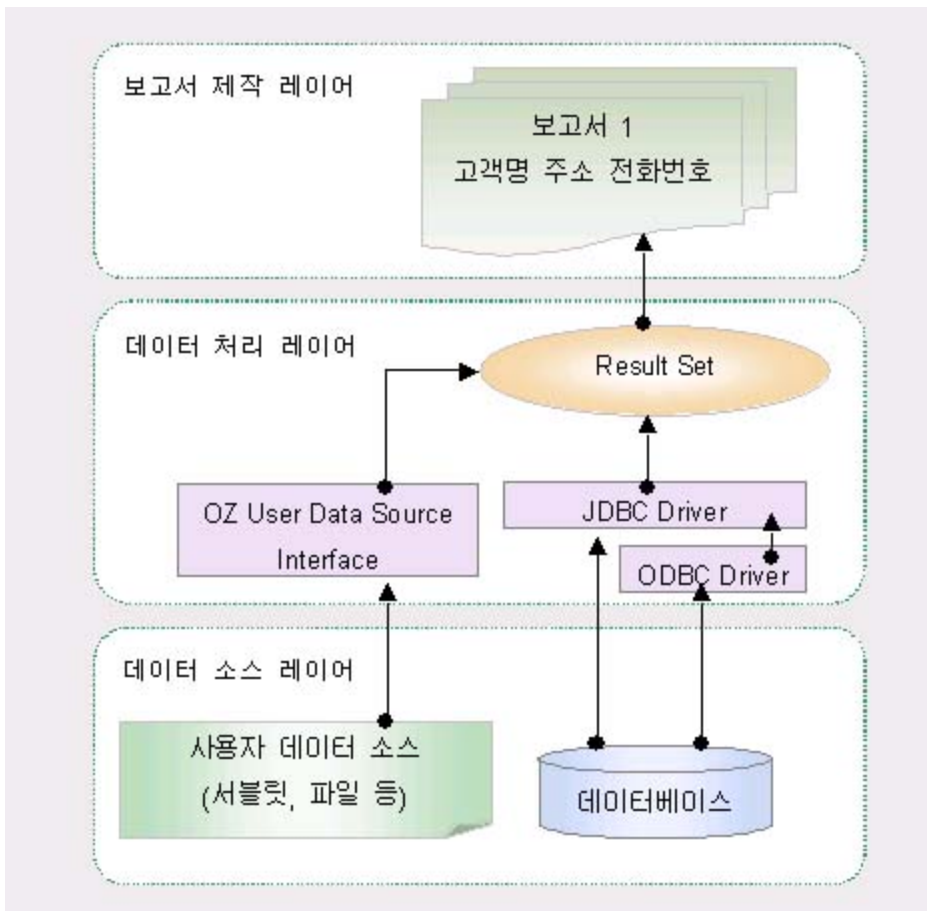
데이터 셋 추가

1. 데이터 셋을 추가하려면 추가된 스토어를 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 **[데이터 셋 추가]** 메뉴를 클릭합니다.
2. 데이터 셋 추가 다이얼로그에서 데이터 셋 이름을 입력하고 **[확인]** 버튼을 클릭합니다.
3. 트리에서 기준 JSON Path으로 설정할 필드를 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 **[기준 JSON Path 지정]** 메뉴를 클릭합니다.
4. 트리에서 추가할 필드를 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 **[필드 추가]** 메뉴를 클릭합니다.
5. 메인 툴바에서 **[쿼리문 실행]** 아이콘을 클릭하여 쿼리문을 실행합니다.

사용자 데이터 스토어

사용자 데이터 스토어 즉 UDS(User Data Store)는 일반 파일 데이터(텍스트 파일, CSV, XML) 및 사용자 프로그램 데이터(EJB, Servlet, ASP, JSP, Applet)와 같은 다양한 형태의 외부 데이터 소스로부터 데이터 셋을 생성하는 Java Interface 규격입니다.

따라서 UDS는 JDBC/ODBC를 통해 데이터베이스로부터 데이터를 가져오는 것과 마찬가지로 사용자 애플리케이션에서 파일이나 XML, 벡터, 스트림, EJB 객체 등의 형태로 데이터를 받아서 오즈의 데이터 소스로 변환해 주는 기능을 담당합니다.



UDS 개발자는 UDS Java Interface에 따라 다양한 형태의 사용자 데이터를 오즈가 인식할 수 있는 ResultSet 객체 형태로 가공하여 전달하는 방법을 구현해야 합니다. 이렇게 구현된 자바 사용자 프로그램은 클래스 파일 형태로 오즈 보고서와 연동되며, UDS를 통해 생성되는 UDS 데이터 셋은 일반 쿼리 데이터 셋과 동일하게 사용할 수 있습니다.

오즈는 사용자 클래스로 패러미터 스트링을 전달할 수 있으며 패러미터 값에 따라ダイナ믹하게 변화하는 데이터를 전달 받을 수도 있습니다.

스토어 추가

1. 오즈 데이터 트리에서 사용자 데이터(UDS)를 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴 중 **[스토어 추가]** 메뉴를 클릭합니다.
2. 사용자 데이터 스토어 추가 다이얼로그에서 사용자 데이터 스토어 이름을 설정하고 **[...]** 버튼을 클릭하여 클래스 파일을 추가하면 클래스(DLL) 파일 경로가 자동으로 설정됩니다.
3. **[확인]** 버튼을 클릭하면 사용자 데이터(UDS) 하위에 추가한 사용자 스토어가 표시됩니다.

다이얼로그 옵션

사용자 데이터 스토어 이름	사용자 데이터 스토어 이름을 설정합니다.
클래스(DLL) 파일 경로	UDS 클래스 파일의 경로를 설정합니다.
필드 정의 사용	해당 UDS 클래스에서 필드 정보를 가져올 수 없는 경우 사용자가 필드 정보를 직접 정의할지 여부를 설정합니다.

데이터셋 추가

1. 오즈 데이터 트리에서 사용자 데이터 스토어를 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴 중 **[데이터 셋 추가]** 메뉴를 클릭하면 데이터 셋 추가 다이얼로그가 나타납니다.
2. 데이터 셋 이름을 입력한 후 **[확인]** 버튼을 클릭하고 쿼리 창의 **[매뉴얼]** 탭을 선택하여 실행문을 입력합니다.
3. 메뉴바의 **[실행] > [쿼리문 실행]** 메뉴를 선택하거나 툴바에서 **[쿼리문 실행]** 아이콘을 클릭하여 쿼리문을 실행합니다. 실행문에 에러가 없을 경우 **[결과]** 탭이 선택되고 결과 데이터가 표시되며 해당 데이터셋 아래에 결과 필드가 생성됩니다.

참고 사항

사용자 데이터 스토어의 데이터셋 추가 시 디자인 모드는 매뉴얼 모드만 지원되며 디자인 뷰 모드는 지원되지 않습니다.

사용자 필드 추가

1. 오즈 데이터 트리에서 사용자 데이터(UDS)를 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴 중 **[스토어 추가]** 메뉴를 클릭합니다.
2. 사용자 데이터 스토어 추가 다이얼로그에서 "필드 정의 사용" 옵션을 체크하여 사용자 데이터 스토어에 사용자 필드를 추가할 수 있도록 설정합니다.
3. 추가된 사용자 데이터 스토어에 데이터셋을 추가한 후 데이터셋을 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 **[사용자 필드 설정]** 메뉴를 클릭합니다.
4. 사용자 필드 설정 다이얼로그에서 필드 이름을 입력하고 필드 타입을 설정한 후 **[확인]** 버튼을 클릭하면 데이터셋에 사용자 필드가 추가되는 것을 확인할 수 있습니다.

참고 사항

사용자 필드는 연결할 클래스 파일을 설정하지 않아도 데이터셋과 필드를 추가할 수 있습니다. 만일 클래스 파일이 설정된 사용자 데이터 스토어에 사용자 필드를 데이터 스토어의 데이터셋 추가 시 디자인 모드는 매뉴얼 모드만 지원되며 디자인 뷰 모드는 지원되지 않습니다.

그룹 데이터 스토어

그룹 데이터 스토어는 다른 데이터 소스(데이터베이스, UDS, 파일/HTTP 스토어, XML DTD 스토어)에서 생성된 데이터 셋을 다수의 데이터 셋으로 분할하는 2차 데이터 소스로서, 다수의 데이터 셋이 하나의 데이터 셋에서 파생 가능할 때 이용하는 데이터 소스입니다.

그룹 데이터 스토어는 하나의 데이터 셋을 이용해 다수의 데이터 셋을 파생시키기 때문에 다수의 독립 데이터 셋을 이용하는 것에 비해 쿼리와 통신 횟수가 감소하는 장점이 있습니다.

예를 들어 마스터-디테일 보고서를 작성하기 위해 세 개의 쿼리로 세 개의 데이터 셋을 생성하는 것보다 하나의 쿼리로 하나의 데이터 셋을 생성하여 GDS로 분할하는 것이 데이터베이스 성능을 향상시킬 수 있습니다.

원본 데이터 셋

다중 데이터 셋으로 분할되는 데이터 셋을 원본 데이터 셋이라 합니다. 마스터/디테일 데이터 셋을 분할할 경우 가장 상위 마스터셋으로 사용되는 데이터 셋만을 원본 데이터 셋으로 사용할 수 있습니다.

그룹 데이터 셋

원본 데이터 셋을 분할해서 만든 데이터 셋을 그룹 데이터 셋이라 합니다. 원본 데이터 셋으로부터 생성된 그룹 데이터 셋은 일반 데이터 셋과 동일하게 사용할 수 있습니다.

스토어 추가

1. 실제 데이터를 가져오기 위한 원본 데이터 셋을 생성한 후, 데이터 창의 **[그룹 데이터]**를 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 **[스토어 추가]** 메뉴를 클릭합니다.
2. 그룹 데이터 스토어 추가 다이얼로그에서 그룹 데이터 스토어 이름을 입력하고 원본셋 이름을 선택한 후 **[확인]** 버튼을 클릭하여 그룹 데이터 스토어를 추가합니다.

데이터 셋 추가

1. 데이터 셋을 추가하려면 추가된 스토어를 마우스 오른쪽 버튼으로 클릭하면 나타나는 팝업 메뉴에서 **[데이터 셋 추가]** 메뉴를 클릭합니다.

2. 그룹 데이터 셋 추가 다이얼로그에서 그룹 데이터 셋 이름을 입력하고 **[확인]** 버튼을 클릭한 후, 그룹 데이터 셋 추가 다이얼로그에서 원본 셋 분할을 위한 조건문을 설정하고 **[확인]** 버튼을 클릭하면 그룹 데이터 스토어 하위에 데이터 셋이 추가됩니다.

다이얼로그 옵션

데이터 셋 이름	설정된 데이터 셋 이름을 보여줍니다.
실행문	그룹 데이터를 생성하기 위한 조건문을 설정합니다.
Consume Row	원본 데이터 셋의 데이터를 여러 개의 데이터 셋으로 분할할 때 사용합니다. 한 행의 데이터를 모두 분할하고 다음 행을 가져오는 시점에 "True"를 설정하여야 그룹 데이터 스토어의 바인딩이 정상적으로 작동합니다. 일반적으로 하나의 그룹 데이터 스토어 내의 데이터 셋들 중 가장 하위의 데이터 셋만 "True"로 설정합니다.
패러미터 추가	패러미터 마법사를 이용하여 사용할 패러미터를 설정합니다.
필드 선택	필드 선택 다이얼로그를 이용하여 사용할 필드를 설정합니다.

데이터 셋 분할

원본 데이터 셋은 분할 조건문에 의해서 여러 개의 그룹 데이터 셋으로 나뉘지는데 오즈 쿼리에서는 수직 분할과 수평 분할을 지원합니다.

수직 분할

그룹 데이터 스토어에 추가된 그룹 데이터 셋을 수직 분할합니다.

일반적으로 가장 많이 사용되는 형태로 하나의 데이터 셋으로 마스터-디테일 관계의 보고서를 작성하기 위해 원본셋을 마스터 그룹 데이터 셋과 디테일셋과 디테일 그룹 데이터 셋으로 분할합니다.

부서별 사원 정보

부서 코드 : 10 부서 명 : ACCOUNTING
부서 위치 : NEW YORK

사원명	직급	급여
CLARK	MANAGER	2572.5
KING	PRESIDENT	5500
MILLER	CLERK	920

부서 코드 : 20 부서 명 : RESEARCH
부서 위치 : DALLAS

사원명	직급	급여
JONES	MANAGER	3123.75
CHAN	ANALYST	3450

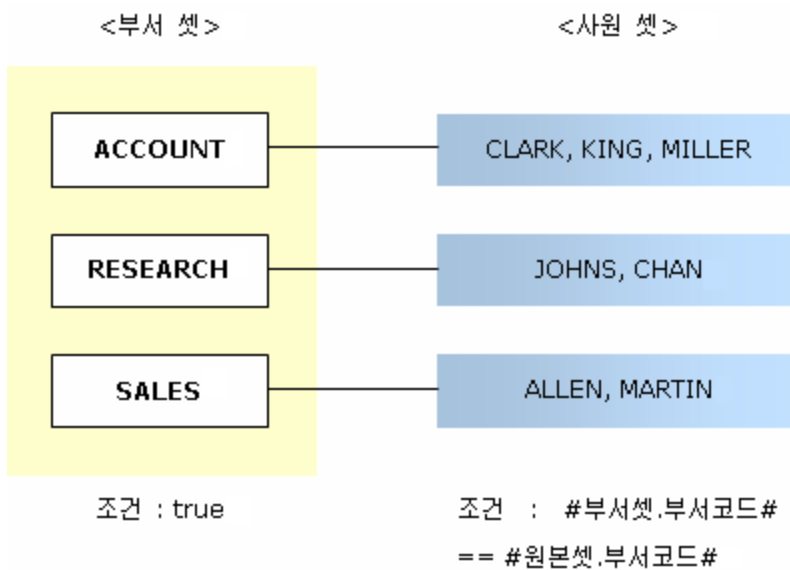
부서 코드 : 30 부서 명 : SALES
부서 위치 : CHICAGO

사원명	직급	급여
ALLEN	SALESMAN	1600
MARTIN	SALESMAN	1312.5

위와 같은 보고서를 제작하기 위해 부서 정보와 사원 정보를 아래의 표와 같이 하나의 데이터 셋으로 생성합니다.

코드	부서명	부서위치	사원명	직급	급여
10	ACCOUNTING	NEW YORK	CLARK	MANAGER	2572.2
10	ACCOUNTING	NEW YORK	KING	PRESIDENT	5500
10	ACCOUNTING	NEW YORK	MILLER	CLERK	920
20	RESEARCH	DALLAS	JONES	MANAGER	3123.75
20	RESEARCH	DALLAS	CHAN	ANALYST	3450
30	SALES	CHICAGO	ALLEN	SALESMAN	1600
30	SALES	CHICAGO	MARTIN	SALESMAN	1312.5

원본 데이터 셋을 수직 분할하여 두 개의 데이터 그룹 데이터 셋 즉, 부서 데이터 셋과 사원 데이터 셋으로 나눌 수 있으며 각 셋의 데이터와 분할 조건은 다음과 같습니다.



조건이 "true"인 경우 원본 셋에서 해당 셋으로 무조건 데이터를 가져옵니다. 조건식이 있을 경우 해당 조건식이 "true"가 될 경우에만 해당 데이터 셋으로 데이터를 가져옵니다.

참고 사항

데이터 셋은 수직 분할 뿐만 아니라 수평 분할도 할 수 있으며 수직 분할과 수평 분할을 혼합하여 사용할 수 있습니다.

수평 분할

그룹 데이터 스토어에 추가된 그룹 데이터 셋을 수평 분할합니다.

데이터 셋을 수평 분할하면 하나의 데이터 셋이 서로 독립된 여러 개의 셋으로 분할됩니다.

아래와 같은 원본 데이터 셋이 있을 때

상품코드	상품명	단가
A	FLAT TV 19"	500,000
A	FLAT TV 21"	800,000
A	FLAT TV 25"	1,500,000
B	광폭 타이어	100,000
B	광폭 타이어 V	180,000
C	사과맛 음료수	500

C	포도맛 음료수	500
---	---------	-----

각 코드 별로 보고서 양식에서 표현하는 형태가 다를 경우 코드 별로 데이터 셋을 분할해서 보고서를 작성합니다.

• <A 셋> (조건) #원본셋.코드# == "A"

상품코드	상품명	단가
A	FLAT TV 19"	500,000
A	FLAT TV 21"	800,000
A	FLAT TV 25"	1,500,000

• <B 셋> (조건) #원본셋.코드# == "B"

상품코드	상품명	단가
B	광폭 타이어	100,000
B	광폭 타이어 V	180,000

• <C 셋> (조건) #원본셋.코드# == "C"

상품코드	상품명	단가
C	사과맛 음료수	500
C	포도맛 음료수	500

참고 사항

데이터 셋은 수평 분할 뿐만 아니라 수직 분할도 할 수 있으며 수직 분할과 수평 분할을 혼합하여 사용할 수 있습니다.

파일 스토어

파일 스토어는 일반 텍스트 파일을 데이터 소스로 활용하는 기능으로써, CSV 형식과 DTD나 XSD를 이용하지 않는 표준 XML 형식을 지원합니다.

데이터 셋 추가

파일 스토어에서는 XML 데이터 셋, CSV 데이터 셋, 다이내믹 XML 데이터 셋, 다이내믹 CSV 데이터 셋을 추가할 수 있습니다.

XML 데이터 셋 추가

1. 데이터 셋을 추가하려면 데이터 창 의 파일 스토어를 마우스 오른쪽 버튼으로 클릭하면 나타나는 팝업 메뉴에서 **[XML 데이터 셋 추가]** 메뉴를 클릭합니다.
2. XML 데이터 셋 추가 - 파일 스토어 다이얼로그의 각 옵션을 설정한 후 **[확인]** 버튼을 클릭하면 파일 스토어에 추가된 XML 데이터 셋 정보가 표시됩니다.

다이얼로그 옵션

XML 데이터 셋 이름	데이터 셋 이름을 설정합니다.
XML 파일 경로	실제 서버에서 사용할 XML 파일의 경로를 설정합니다. [패러미터] 버튼을 클릭하면 패러미터 마법사를 이용하여 XML 파일 경로를 설정할 수 있으며, [경로] 버튼을 클릭하면 XML 파일을 선택할 수 있습니다.
디자인시 사용할 임시 XML 파일 경로	오즈 서버와 디자이너의 XML 파일 경로가 다른 경우 디자인 시점에만 사용하는 임시 XML 파일의 경로를 설정합니다. [패러미터] 버튼을 클릭하면 패러미터 마법사를 이용하여 디자인 시 사용할 임시 XML 파일 경로를 설정할 수 있으며, [경로] 버튼을 클릭하면 XML 파일을 선택할 수 있습니다.
데이터 셋 노드 이름	해당 XML 파일에서 데이터 셋을 나타내는 노드 이름을 설정합니다. 기본 값은 DATASET입니다.

레코드 셋 노드 이름	해당 XML 파일에서 레코드 셋을 나타내는 노드 이름을 설정합니다. 기본 값은 RECORD입니다.
앨리어스	해당 데이터 셋 설정 정보를 별도의 연결 정보 파일에서 가져올지 여부를 설정하며 앨리어스를 사용할 경우 오즈 서버의 datasource.properties 파일에 정의한 앨리어스 이름을 설정합니다. 앨리어스 정보 파일: 앨리어스 정보가 있는 데이터 셋 설정 정보 파일을 설정합니다.
데이터 보기	설정된 XML 파일의 데이터를 미리보기합니다.
데이터 셋 정보 저장	데이터 셋의 정보(필드 이름, 필드 타입)을 XML 파일로 저장합니다. 해당 버튼은 데이터 셋을 수정할 경우에만 활성화되며, 데이터 셋 정보 저장 시 인코딩이 UTF-8로 설정됩니다

CSV 데이터 셋 추가

1. 데이터 셋을 추가하려면 데이터 창 의 파일 스토어를 마우스 오른쪽 버튼으로 클릭하면 나타나는 팝업 메뉴에서 **[CSV 데이터 셋 추가]** 메뉴를 클릭합니다.
2. CSV 데이터 셋 추가 - 파일 스토어 다이얼로그의 각 옵션을 설정한 후 **[확인]** 버튼을 클릭하면 파일 스토어에 추가된 CSV 데이터 셋 정보가 표시됩니다.

다이얼로그 옵션

CSV 데이터 셋 이름	데이터 셋 이름을 설정합니다.
CSV 파일 경로	실제 서버에서 사용할 CSV 파일의 경로를 설정합니다. [패러미터] 버튼을 클릭하면 패러미터 마법사를 이용하여 CSV 파일 경로를 설정할 수 있으며, [경로] 버튼을 클릭하면 CSV 파일을 선택할 수 있습니다.
디자인시 사용할 임시 CSV 파일 경로	오즈 서버와 디자이너의 CSV 파일 경로가 다른 경우 디자인 시점에만 사용하는 임시 CSV 파일의 경로를 설정합니다. [패러미터] 버튼을 클릭하면 패러미터 마법사를 이용하여 디자인 시 사용할 임시 CSV 파일 경로를 설정할 수 있으며, [경로] 버튼을 클릭하면 CSV 파일을 선택할 수 있습니다.
필드 구분자	필드 구분자를 설정합니다. 미리 정의된 필드 구분자인 TAB, SPACE, SEMICOLON, COLON, COMMA중에서 선택하거나 직

	접 입력하여 필드 구분자를 설정할 수 있습니다. [패러미터] 버튼을 클릭하면 패러미터 마법사를 이용하여 필드 구분자를 설정할 수 있습니다.
필드 타입 정보 포함	해당 CSV 파일에 필드에 대한 타입 정보가 포함되어 있는지 여부를 선택합니다. 체크하지 않으면 모든 필드의 타입은 "VARCHAR"로 간주됩니다.
라인 구분자	라인 구분자를 선택합니다. 기본 값은 LINEFEED이며 직접 입력하여 설정할 수 있습니다. [패러미터] 버튼을 클릭하면 패러미터 마법사를 이용하여 라인 구분자를 설정할 수 있습니다.
널 마크	NULL 데이터를 구분하기 위한 문자열을 설정합니다.
문자 셋	파일의 문자 셋을 정의합니다. 설정하지 않을 경우 시스템의 기본 로케일 값이 이용됩니다.
에러 표시 코드	데이터 에러를 표시하기 위한 코드를 설정합니다. 설정한 코드 값이 파일의 첫 줄에 있을 경우 보고서 미리보기를 수행하지 않고 에러 메시지 창을 실행합니다. 예를 들어 CSV 파일의 첫 줄에 "ERR"이라는 구분이 있고 에러 표시 코드를 "ERR"이라고 입력하면 "ERR"을 데이터로 인식하지 않고 해당 파일이 적당한 CSV 데이터 파일이 아니라는 의미의 에러 메시지 창이 실행됩니다.
쌍따옴표 감싸짐	쌍따옴표로 감싸진 데이터를 하나의 데이터 값으로 바인딩할지 여부를 설정합니다.
앨리어스	해당 데이터 셋 설정 정보를 별도의 연결 정보 파일에서 가져올지 여부를 설정하며 앨리어스를 사용할 경우 오즈 서버의 datasource.properties 파일에 정의한 앨리어스 이름을 설정합니다. 앨리어스 정보 파일: 앨리어스 정보가 있는 데이터 셋 설정 정보 파일을 설정합니다.
데이터 보기	설정된 CSV 파일의 데이터를 미리보기합니다.
데이터 셋 정보 저장	데이터 셋의 정보(필드 이름, 필드 타입)를 CSV 파일로 저장합니다. 해당 버튼은 데이터 셋을 수정할 경우에만 활성화되며, 필드 타입은 "필드 타입 정보 포함" 옵션을 체크하였을 경우에만 저장합니다.

다이나믹 XML 데이터 셋 추가

1. 데이터 셋을 추가하려면 데이터 창 의 파일 스토어를 마우스 오른쪽 버튼으로 클릭하면 나타나는 팝업 메뉴에서 **[다이나믹 데이터 셋 추가]** 메뉴를 클릭합니다.
2. 다이나믹 데이터 셋 추가 - 파일 스토어 다이얼로그에서 "XML 데이터 셋"을 선택합니다.
3. 다이나믹 데이터 셋 추가 - XML 다이얼로그의 각 옵션을 설정한 후 **[확인]** 버튼을 클릭하면 파일 스토어에 추가된 다이나믹 XML 데이터 셋 정보가 표시됩니다.

다이얼로그 옵션

XML 데이터 셋 이름	데이터 셋 이름을 설정합니다.
시작 행 번호	다이나믹 데이터 셋 생성 시 ROW INDEX에 처음 시작되는 번호입니다. 기본 값은 0이며 각 ROW INDEX는 0부터 차례로 번호가 부여되며, 0~2147483647 이외의 값은 시작 행 번호로 설정할 수 없습니다.
XML 파일 경로	실제 서버에서 사용할 XML 파일의 경로를 설정합니다. [패러미터] 버튼을 클릭하면 패러미터 마법사를 이용하여 XML 파일 경로를 설정할 수 있으며, [경로] 버튼을 클릭하면 XML 파일을 선택할 수 있습니다.
디자인시 사용할 임시 XML 파일 경로	오즈 서버와 디자이너의 XML 파일 경로가 다른 경우 디자인 시점에만 사용하는 임시 XML 파일의 경로를 설정합니다. [패러미터] 버튼을 클릭하면 패러미터 마법사를 이용하여 디자인 시 사용할 임시 XML 파일 경로를 설정할 수 있으며, [경로] 버튼을 클릭하면 XML 파일을 선택할 수 있습니다.
데이터 셋 노드 이름	해당 XML 파일에서 데이터 셋을 나타내는 노드 이름을 설정합니다. 기본 값은 DATASET입니다.
레코드 셋 노드 이름	해당 XML 파일에서 레코드 셋을 나타내는 노드 이름을 설정합니다. 기본 값은 RECORD입니다.
앨리어스	해당 데이터 셋 설정 정보를 별도의 연결 정보 파일에서 가져올지 여부를 설정하며 앨리어스를 사용할 경우 오즈 서버의 datasource.properties 파일에 정의한 앨리어스 이름을 설정합니다. 앨리어스 정보: 파일 앨리어스 정보가 있는 데이터 셋 설정 정보 파일을 설정합니다.
데이터 보기	설정한 XML 파일의 데이터를 미리보기합니다.

데이터 셋 정보 저장	데이터 셋의 정보(필드 이름, 필드 타입)을 XML 파일로 저장합니다. 해당 버튼은 데이터 셋을 수정할 경우에만 활성화되며, 데이터 셋 정보 저장 시 인코딩이 UTF-8로 설정됩니다.
-------------	--

다이나믹 CSV 데이터 셋 추가

1. 데이터 셋을 추가하려면 데이터 창 의 파일 스토어를 마우스 오른쪽 버튼으로 클릭하면 나타나는 팝업 메뉴에서 **[다이나믹 데이터 셋 추가]** 메뉴를 클릭합니다.
2. 다이나믹 데이터 셋 추가 - 파일 스토어 다이얼로그에서 "CSV 데이터 셋"을 선택합니다.
3. 다이나믹 데이터 셋 추가 - CSV 다이얼로그의 각 옵션을 설정한 후 **[확인]** 버튼을 클릭하면 파일 스토어에 추가된 다이나믹 CSV 데이터 셋 정보가 표시됩니다.

다이얼로그 옵션

CSV 데이터 셋 이름	데이터 셋 이름을 설정합니다.
시작 행 번호	다이나믹 데이터 셋 생성 시 ROW INDEX에 처음 시작되는 번호입니다. 기본 값은 0이며 각 ROW INDEX는 0부터 차례로 번호가 부여되며, 0~2147483647 이외의 값은 시작 행 번호로 설정할 수 없습니다.
CSV 파일 경로	실제 서버에서 사용할 CSV 파일의 경로를 설정합니다. [패러미터] 버튼을 클릭하면 패러미터 마법사를 이용하여 CSV 파일 경로를 설정할 수 있으며, [경로] 버튼을 클릭하면 CSV 파일을 선택할 수 있습니다.
디자인시 사용할 임시 CSV 파일 경로	오즈 서버와 디자이너의 CSV 파일 경로가 다른 경우 디자인 시점에만 사용하는 임시 CSV 파일의 경로를 설정합니다. [패러미터] 버튼을 클릭하면 패러미터 마법사를 이용하여 디자인 시 사용할 임시 CSV 파일 경로를 설정할 수 있으며, [경로] 버튼을 클릭하면 CSV 파일을 선택할 수 있습니다.
필드 구분자	필드 구분자를 설정합니다. 미리 정의된 필드 구분자인 TAB, SPACE, SEMICOLON, COLON, COMMA중에서 선택하거나 직접 입력하여 필드 구분자를 설정할 수 있습니다. [패러미터] 버튼을 클릭하면 패러미터 마법사를 이용하여 필드 구분자를 설정할 수 있습니다. 다이나믹 데이터 셋은 CSV 파일에 설정된 필드 이름을 사용하지 않고 ROWIDX, COLUMN, VALUE로 필드가 생성됩니다.

필드 타입 정보 포함	해당 CSV 파일의 필드에 대한 타입 정보가 포함되어 있는지 여부를 선택합니다. 다이나믹 데이터 셋은 CSV 파일에 설정된 필드 타입을 사용하지 않고, ROWIDX는 INTEGER로, COLUMN, VALUE는 VARCHAR로 설정됩니다.
라인 구분자	라인 구분자를 선택합니다. 미리 정의된 라인 구분자인 LINEFEED로 설정하거나 직접 입력하여 설정할 수 있습니다. [패러미터] 버튼을 클릭하면 패러미터 마법사를 이용하여 라인 구분자를 설정할 수 있습니다.
널 마크	NULL 데이터를 구분하기 위한 문자열을 설정합니다.
문자 셋	파일의 문자 셋을 정의합니다. 설정하지 않을 경우 시스템의 기본 로케일 값이 이용됩니다.
에러 표시 코드	데이터 에러를 표시하기 위한 코드를 설정합니다. 설정한 코드 값이 파일의 첫 줄에 있을 경우 보고서 미리보기를 수행하지 않고 에러 메시지 창을 실행합니다. 예를 들어 CSV 파일의 첫 줄에 "ERR"이라는 구분이 있고 에러 표시 코드를 "ERR"이라고 입력하면 "ERR"을 데이터로 인식하지 않고 해당 파일이 적당한 CSV 데이터 파일이 아니라는 의미의 에러 메시지 창이 실행됩니다.
쌍따옴표 감싸짐	쌍따옴표로 감싸진 데이터를 하나의 데이터 값으로 바인딩할지 여부를 설정합니다.
앨리어스	해당 데이터 셋 설정 정보를 별도의 연결 정보 파일에서 가져올지 여부를 설정하며 앨리어스를 사용할 경우 오즈 서버의 datasource.properties 파일에 정의한 앨리어스 이름을 설정합니다. 앨리어스 정보: 파일 앨리어스 정보가 있는 데이터 셋 설정 정보 파일을 설정합니다.
데이터 보기	설정한 CSV 파일의 데이터를 미리보기합니다.
데이터 셋 정보 저장	CSV 파일의 데이터 셋 정보(필드 이름, 필드 타입)를 CSV 파일로 저장합니다. 해당 버튼은 데이터 셋을 수정할 경우에만 활성화되며, 필드 타입은 "필드 타입 정보 포함" 옵션을 체크하였을 경우에만 저장합니다.

파일 스토어에서 사용되는 파일 규격

파일 스토어로 데이터 셋을 생성할 때 사용되는 XML, CSV 파일 규격에 대해 설명합니다.

XML 파일 규격

XML은 웹 상에서 서로 다른 업체의 응용 프로그램 간의 데이터 교환을 위한 데이터 표현 언어로써, The World Wide Web Consortium(W3C)이 정의한 데이터를 응용 프로그램으로부터 분리하는 웹 언어입니다. 예를 들어, HTML이 웹 상에서 표준 UI를 구현하는 웹 언어라면 XML은 웹 상에서의 데이터 교환을 위한 웹 언어라고 할 수 있습니다.

오즈는 XML을 데이터 소스로 이용하기 위해 파일 스토어와 XML 스토어를 제공합니다. DTD나 XSD 없이 정해진 형태의 단순한 XML을 데이터 소스로 이용하기 위해서는 파일 스토어를 이용하여야 하고, XML 스토어는 DTD나 XSD를 이용한 XML을 데이터 소스로 이용할 수 있습니다.

파일 스토어가 지원하는 XML 파일 규격은 다음과 같습니다.

- DTD가 없는 표준 XML 규격에 맞는 파일을 지원합니다.
- 데이터 셋 및 레코드 셋을 나타내는 노드를 포함하여야 합니다.
- 필드 정보를 나타내는 노드는 다음과 같은 형태로 기술되어야 하며 생략할 수 있습니다.

다음은 데이터 셋, 레코드 셋 노드의 이름이 각각 "DATASET", "RECORD"이고 필드 정보 노드를 포함한 XML 파일의 예제입니다.

```
<?xml version="1.0" encoding="KSC5601"?>
<XMLSET>
<DATASET>
<FIELDINFO>
<FIELD NAME="ID" TYPE="INT"/>
<FIELD NAME="NAME" TYPE="CHAR"/>
<FIELD NAME="ADDRESS" TYPE="VARCHAR"/>
</FIELDINFO>
<RECORD>
<ID>1</ID>
<NAME>Park Chan Ho</NAME>
<ADDRESS>Texas</ADDRESS>
</RECORD>
<RECORD>
<ID>2</ID>
<NAME>Shin Ki Sung</NAME>
```

```

<ADDRESS>Seoul</ADDRESS>
</RECORD>
<RECORD>
<ID>3</ID>
<NAME>Kim Byung Hyun</NAME>
<ADDRESS>Arizona</ADDRESS>
</RECORD>
</DATASET>
</XMLSET>

```

CSV 파일 규격

CSV 파일은 데이터베이스 간의 데이터 교환을 위해 널리 사용되는 파일 포맷으로, 필드와 레코드를 구분하기 위해 콤마를 이용하는 일반 텍스트 파일입니다. 표준 CSV 파일은 콤마만을 구분자로 이용하지만, 오즈 파일 스토어에서는 특정 문자를 구분자로 지정할 수 있습니다.

파일 스토어가 지원하는 CSV 파일 규격은 다음과 같습니다.

- DTD가 없는 표준 XML 규격에 맞는 파일을 지원합니다.
- 첫 줄과 두 번째 줄에 필드 이름과 필드 타입으로 구성되는 필드 정보가 오고, 필드 정보 다음 줄부터 필드 정보와 일치하는 데이터 레코드가 반복되어야 합니다.
- 필드 간의 구분자는 모두 동일한 문자를 이용하여야 하며, 필드 정보와 데이터 레코드는 새로운 줄에서 시작되어야 합니다.
- 필드 정보에서 필드 타입을 나타내는 두 번째 줄은 필요에 따라 생략될 수 있습니다.

다음은 ","를 구분자로 이용하는 CSV 파일의 예제입니다.

```

no,jumin_id,name,address
INT, VARCHAR, VARCHAR, VARCHAR
1,7411231932347,kim,seoul
2,6305212983717,lee,pusan
3,7006031837261,park,daegu
4,7271722848281,song,daejeon

```

HTTP 스토어

HTTP 스토어는 HTTP 출력을 데이터 소스로 활용하는 기능으로써, CSV 형식과 DTD나 XSD를 이용하지 않는 표준 XML 형식을 지원합니다.

데이터 셋 추가

HTTP 스토어에서는 XML 데이터 셋, CSV 데이터 셋, 다이내믹 XML 데이터 셋, 다이내믹 CSV 데이터 셋을 추가할 수 있습니다.

XML 데이터 셋 추가

1. 데이터 셋을 추가하려면 데이터 창의 HTTP 스토어를 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 **[XML 데이터 셋 추가]** 메뉴를 클릭합니다.
2. XML 데이터 셋 추가 - HTTP 스토어 다이얼로그의 각 옵션을 설정한 후 **[확인]** 버튼을 클릭하면 파일 스토어에 추가된 XML 데이터 셋 정보가 표시됩니다.

다이얼로그 옵션

XML 데이터 셋 이름	데이터 셋 이름을 설정합니다.
사용자 애플리케이션 주소(URL)	데이터를 가져올 파일 또는 사용자 애플리케이션의 HTTP 주소입니다. [패러미터] 버튼을 클릭하면 패러미터 마법사를 이용하여 사용자 애플리케이션 주소(URL)를 설정할 수 있습니다.
디자인시 사용할 임시 XML 파일 경로	디자인 시점에만 사용할 수 있는 로컬 시스템의 파일 경로를 설정합니다. [패러미터] 버튼을 클릭하면 패러미터 마법사를 이용하여 디자인 시 사용할 임시 XML 파일 경로를 설정할 수 있습니다. [경로] 버튼을 클릭하여 가져올 XML 파일을 선택합니다.
데이터 셋 노드 이름	해당 XML 파일에서 데이터 셋을 나타내는 노드 이름을 설정합니다. 기본 값은 DATASET입니다.
레코드 셋 노드 이름	해당 XML 파일에서 레코드 셋을 나타내는 노드 이름을 설정합니다. 기본 값은 RECORD입니다.

서버 로그 파일 이름	ODI가 서버에서 바인딩될 때 전달받은 데이터에 대한 로그를 서버에 남길 경우 서버 로그 파일명을 설정합니다. 파일 이름에는 경로를 설정하지 않고 파일명만 입력하여야 합니다.
재시도 횟수	HTTP 스토어의 데이터 셋 사용 시 에러가 발생할 경우 연결을 재시도하는 횟수입니다. 값을 설정하지 않거나 잘못된 값을 설정할 경우 기본 값으로 적용됩니다.
URL 인코드 설정	Server URL 인코딩 유무를 설정합니다. 데이터 요청시 URL의 패러미터 부분에 한국어나 기타 다국어 등이 포함되어 있을 경우 체크합니다.
포스트 타입 전송	URL의 패러미터 전송 시 GET 방식을 이용할 것인지 POST 방식을 이용할 것인지 선택합니다. 옵션을 체크하면 POST 방식으로 전송됩니다.
앨리어스	해당 데이터 셋 설정 정보를 별도의 연결 정보 파일에서 가져올지 여부를 설정하며 앨리어스를 사용할 경우 오즈 서버의 datasource.properties 파일에 정의한 앨리어스 이름을 설정합니다. 앨리어스 정보: 파일 앨리어스 정보가 있는 데이터 셋 설정 정보 파일을 설정합니다.
데이터 보기	설정된 XML 파일의 데이터를 미리보기합니다. 서버 로그 파일 이름이 설정되어 있으면 최대 행 수와 관계 없이 모든 행을 가져온 후 데이터 보기가 되므로 속도가 느려질 수 있습니다.
데이터 셋 정보 저장	데이터 셋의 정보(필드 이름, 필드 타입)를 XML 파일로 저장합니다. 해당 버튼은 데이터 셋을 수정할 경우에만 활성화되며, 데이터 셋 정보 저장 시 인코딩이 UTF-8로 설정됩니다.
원본 저장	데이터 셋의 원본 데이터를 파일로 저장합니다. [원본 저장] 버튼은 데이터 셋을 수정할 경우에만 활성화됩니다.

CSV 데이터 셋 추가

- 데이터 셋을 추가하려면 데이터 창이 HTTP 스토어를 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 **[CSV 데이터 셋 추가]** 메뉴를 클릭합니다.

2. CSV 데이터 셋 추가 - HTTP 스토어 다이얼로그의 각 옵션을 설정한 후 **[확인]** 버튼을 클릭하면 파일 스토어에 추가된 CSV 데이터 셋 정보가 표시됩니다.

다이얼로그 옵션

CSV 데이터 셋 이름	데이터 셋 이름을 설정합니다.
사용자 애플리케이션 주소(URL)	데이터를 가져올 파일 또는 사용자 애플리케이션의 HTTP 주소입니다. [패러미터] 버튼을 클릭하면 패러미터 마법사를 이용하여 사용자 애플리케이션 주소(URL)을 설정할 수 있습니다.
디자인시 사용할 임시 CSV 파일 경로	디자인 시점에만 사용할 수 있는 로컬 시스템의 파일 경로를 설정합니다. [패러미터] 버튼을 클릭하면 패러미터 마법사를 이용하여 디자인 시 사용할 임시 CSV 파일 경로를 설정할 수 있습니다. [경로] 버튼을 클릭하여 가져올 CSV 파일을 선택합니다.
필드 구분자	필드 구분자를 설정합니다. 미리 정의된 필드 구분자인 TAB, SPACE, SEMICOLON, COLON, COMMA중에서 선택하거나 직접 입력하여 필드 구분자를 설정할 수 있습니다. [패러미터] 버튼을 클릭하면 패러미터 마법사를 이용하여 필드 구분자를 설정할 수 있습니다.
필드 타입 정보 포함	해당 CSV 파일에 필드에 대한 타입 정보가 포함되어 있는지 여부를 선택합니다. 체크하지 않으면 모든 필드의 타입은 "VARCHAR"로 간주됩니다.
라인 구분자	라인 구분자를 선택합니다. 기본 값은 LINEFEED이며 직접 입력하여 설정할 수 있습니다. [패러미터] 버튼을 클릭하면 패러미터 마법사를 이용하여 라인 구분자를 설정할 수 있습니다.
널 마크	NULL 데이터를 구분하기 위한 문자열을 설정합니다.
문자 셋	파일의 문자 셋을 정의합니다. 설정하지 않을 경우 시스템의 기본 로케일 값이 이용됩니다.
에러 표시 코드	데이터 에러를 표시하기 위한 코드를 설정합니다. 설정한 코드 값이 파일의 첫 줄에 있을 경우 보고서 미리보기를 수행하지 않고 에러 메시지 창을 실행합니다. 예를 들어 CSV 파일의 첫 줄에 "ERR"이라는 구분이 있고 에러 표시 코드를 "ERR"이라고 입력하면 "ERR"을 데이터로 인식하지 않고 해당 파일이 적당한 CSV 데이터 파일이 아니라는 의미의 에러 메시지 창이 실행됩니다.

쌍따옴표 감싸짐	쌍따옴표로 감싸진 데이터를 하나의 데이터 값으로 바인딩할지 여부를 설정합니다.
서버 로그 파일 이름	ODI가 서버에서 바인딩될 때 전달받은 데이터에 대한 로그를 서버에 남길 경우 서버 로그 파일명을 설정합니다. 파일 이름에는 경로를 설정하지 않고 파일명만 입력하여야 합니다.
재시도 횟수	HTTP 스토어의 데이터 셋 사용 시 에러가 발생할 경우 연결을 재시도하는 횟수입니다. 값을 설정하지 않거나 잘못된 값을 설정할 경우 기본 값으로 적용됩니다.
URL 인코드 설정	Server URL 인코딩 유무를 설정합니다. 데이터 요청 시 URL의 패러미터 부분에 한국어나 기타 다국어 등이 포함되어 있을 경우 체크합니다.
포스트 타입 전송	URL의 패러미터 전송 시 GET 방식을 이용할 것인지 POST 방식을 이용할 것인지 선택합니다.
앨리어스	해당 데이터 셋 설정 정보를 별도의 연결 정보 파일에서 가져올지 여부를 설정하며 앨리어스를 사용할 경우 오즈 서버의 datasource.properties 파일에 정의한 앨리어스 이름을 설정합니다. 앨리어스 정보: 파일 앨리어스 정보가 있는 데이터 셋 설정 정보 파일을 설정합니다.
데이터 보기	설정된 CSV 파일의 데이터를 미리보기합니다. 서버 로그 파일 이름이 설정되어 있으면 최대 행 수와 관계 없이 모든 행을 가져온 후 데이터 보기가 되므로 속도가 느려질 수 있습니다.
데이터 셋 정보 저장	데이터 셋의 정보(필드 이름, 필드 타입)을 CSV 파일로 저장합니다. 해당 버튼은 데이터 셋을 수정할 경우에만 활성화됩니다.
원본 저장	데이터 셋의 원본 데이터를 파일로 저장합니다. [원본 저장] 버튼은 데이터 셋을 수정할 경우에만 활성화됩니다.

다이나믹 XML 데이터 셋 추가

- 데이터 셋을 추가하려면 데이터 창 의 HTTP 스토어를 마우스 오른쪽 버튼으로 클릭하면 나타나는 팝업 메뉴에서 **[다이나믹 데이터 셋 추가]** 메뉴를 클릭합니다.

2. 다이나믹 데이터 셋 추가 - HTTP 스토어 다이얼로그에서 "XML 데이터 셋"을 선택합니다.
3. 다이나믹 데이터 셋 추가 - XML 다이얼로그의 각 옵션을 설정한 후 **[확인]** 버튼을 클릭하면 파일 스토어에 추가된 다이나믹 XML 데이터 셋 정보가 표시됩니다.

다이얼로그 옵션

XML 데이터 셋 이름	데이터 셋 이름을 설정합니다.
시작 행 번호	다이나믹 데이터 셋 생성 시 ROW INDEX에 처음 시작되는 번호입니다. 기본 값은 0이며 각 ROW INDEX는 0부터 차례로 번호가 부여되며, 0~2147483647 이외의 값은 시작 행 번호로 설정할 수 없습니다.
사용자 애플리케이션 주소(URL)	데이터를 가져올 파일 또는 사용자 애플리케이션의 HTTP 주소입니다. [패러미터] 버튼을 클릭하면 패러미터 마법사를 이용하여 XML 파일 경로를 설정할 수 있습니다.
디자인시 사용할 임시 XML 파일 경로	디자인 시점에만 사용할 수 있는 로컬 시스템의 파일 경로를 설정합니다. [패러미터] 버튼을 클릭하면 패러미터 마법사를 이용하여 디자인 시 사용할 임시 XML 파일 경로를 설정할 수 있습니다. [경로] 버튼을 클릭하여 가져올 XML 파일을 선택합니다.
데이터 셋 노드 이름	해당 XML 파일에서 데이터 셋을 나타내는 노드 이름을 설정합니다. 기본 값은 DATASET입니다.
레코드 셋 노드 이름	해당 XML 파일에서 레코드 셋을 나타내는 노드 이름을 설정합니다. 기본 값은 RECORD입니다.
서버 로그 파일 이름	ODI가 서버에서 바인딩될 때 전달받은 데이터에 대한 로그를 서버에 남길 경우 서버 로그 파일명을 설정합니다. 파일 이름에는 경로를 설정하지 않고 파일명만 입력하여야 합니다.
재시도 횟수	HTTP 스토어의 데이터 셋 사용 시 에러가 발생할 경우 연결을 재시도하는 횟수입니다. 값을 설정하지 않거나 잘못된 값을 설정할 경우 기본 값으로 적용됩니다.
URL 인코드 설정	Server URL 인코딩 유무를 설정합니다. 데이터 요청 시 URL의 패러미터 부분에 한국어나 기타 다국어 등이 포함되어 있을 경우 체크합니다.

포스트 타입 전송	URL의 패러미터 전송 시 GET 방식을 이용할 것인지 POST 방식을 이용할 것인지 선택합니다. 옵션을 체크하면 POST 방식으로 전송됩니다.
앨리어스	해당 데이터 셋 설정 정보를 별도의 연결 정보 파일에서 가져올지 여부를 설정하며 앨리어스를 사용할 경우 오즈 서버의 datasource.properties 파일에 정의한 앨리어스 이름을 설정합니다. 앨리어스 정보: 파일 앨리어스 정보가 있는 데이터 셋 설정 정보 파일을 설정합니다.
데이터 보기	설정된 XML 파일의 데이터를 미리보기합니다. 서버 로그 파일 이름이 설정되어 있으면 최대 행 수와 관계 없이 모든 행을 가져온 후 데이터 보기가 되므로 속도가 느려질 수 있습니다.
데이터 셋 정보 저장	데이터 셋의 정보(필드 이름, 필드 타입)를 XML 파일로 저장합니다. 해당 버튼은 데이터 셋을 수정할 경우에만 활성화되며, 데이터 셋 정보 저장 시 인코딩이 UTF-8로 설정됩니다.
원본 저장	데이터 셋의 원본 데이터를 파일로 저장합니다. [원본 저장] 버튼은 데이터 셋을 수정할 경우에만 활성화됩니다.

다이나믹 CSV 데이터 셋 추가

1. 데이터 셋을 추가하려면 데이터 창 의 HTTP 스토어를 마우스 오른쪽 버튼으로 클릭하면 나타나는 팝업 메뉴에서 **[다이나믹 데이터 셋 추가]** 메뉴를 클릭합니다.
2. 다이나믹 데이터 셋 추가 - HTTP 스토어 다이얼로그에서 "CSV 데이터 셋"을 선택합니다.
3. 다이나믹 데이터 셋 추가 - CSV 다이얼로그의 각 옵션을 설정한 후 **[확인]** 버튼을 클릭하면 파일 스토어에 추가된 다이나믹 CSV 데이터 셋 정보가 표시됩니다.

다이얼로그 옵션

CSV 데이터 셋 이름	데이터 셋 이름을 설정합니다.
시작 행 번호	다이나믹 데이터 셋 생성 시 ROW INDEX에 처음 시작되는 번호입니다. 기본 값은 0이며 각 ROW INDEX는 0부터 차례로 번호가 부여되며, 0~2147483647 이외의 값은 시작 행 번호로 설정할 수 없습니다.

사용자 애플리케이션 주소(URL)	데이터를 가져올 파일 또는 사용자 애플리케이션의 HTTP 주소입니다. [패러미터] 버튼을 클릭하면 패러미터 마법사를 이용하여 사용자 애플리케이션 주소(URL)를 설정할 수 있습니다.
디자인시 사용할 임시 CSV 파일 경로	오즈 서버와 디자이너의 CSV 파일 경로가 다른 경우 디자인 시점에만 사용하는 임시 CSV 파일의 경로를 설정합니다. [패러미터] 버튼을 클릭하면 패러미터 마법사를 이용하여 디자인 시 사용할 임시 CSV 파일 경로를 설정할 수 있으며, [경로] 버튼을 클릭하면 CSV 파일을 선택할 수 있습니다.
필드 구분자	필드 구분자를 설정합니다. 미리 정의된 필드 구분자인 TAB, SPACE, SEMICOLON, COLON, COMMA중에서 선택하거나 직접 입력하여 필드 구분자를 설정할 수 있습니다. [패러미터] 버튼을 클릭하면 패러미터 마법사를 이용하여 필드 구분자를 설정할 수 있습니다. 다이나믹 데이터 셋은 CSV 파일에 설정된 필드 이름을 사용하지 않고 ROWIDX, COLUMN, VALUE로 필드가 생성됩니다.
필드 타입 정보 포함	해당 CSV 파일의 필드에 대한 타입 정보가 포함되어 있는지 여부를 선택합니다. 다이나믹 데이터 셋은 CSV 파일에 설정된 필드 타입을 사용하지 않고, ROWIDX는 INTEGER로, COLUMN, VALUE는 VARCHAR로 설정됩니다.
라인 구분자	라인 구분자를 선택합니다. 미리 정의된 라인 구분자인 LINEFEED로 설정하거나 직접 입력하여 설정할 수 있습니다. [패러미터] 버튼을 클릭하면 패러미터 마법사를 이용하여 라인 구분자를 설정할 수 있습니다.
널 마크	NULL 데이터를 구분하기 위한 문자열을 설정합니다.
문자 셋	파일의 문자 셋을 정의합니다. 설정하지 않을 경우 시스템의 기본 로케일 값이 이용됩니다.
에러 표시 코드	데이터 에러를 표시하기 위한 코드를 설정합니다. 설정한 코드 값이 파일의 첫 줄에 있을 경우 보고서 미리보기를 수행하지 않고 에러 메시지 창을 실행합니다. 예를 들어 CSV 파일의 첫 줄에 "ERR"이라는 구분이 있고 에러 표시 코드를 "ERR"이라고 입력하면 "ERR"을 데이터로 인식하지 않고 해당 파일이 적당한 CSV 데이터 파일이 아니라는 의미의 에러 메시지 창이 실행됩니다.

쌍따옴표 감싸짐	쌍따옴표로 감싸진 데이터를 하나의 데이터 값으로 바인딩할지 여부를 설정합니다.
서버 로그 파일 이름	ODI가 서버에서 바인딩될 때 전달받은 데이터에 대한 로그를 서버에 남길 경우 서버 로그 파일명을 설정합니다. 파일 이름에는 경로를 설정하지 않고 파일명만 입력하여야 합니다.
재시도 횟수	HTTP 스토어의 데이터 셋 사용 시 에러가 발생할 경우 연결을 재시도하는 횟수입니다. 값을 설정하지 않거나 잘못된 값을 설정할 경우 기본 값으로 적용됩니다.
URL 인코드 설정	Server URL 인코딩 유무를 설정합니다. 데이터 요청 시 URL의 패러미터 부분에 한국어나 기타 다국어 등이 포함되어 있을 경우 체크합니다.
포스트 타입 전송	URL의 패러미터 전송 시 GET 방식을 이용할 것인지 POST 방식을 이용할 것인지 선택합니다. 옵션을 체크하면 POST 방식으로 전송됩니다.
앨리어스	해당 데이터 셋 설정 정보를 별도의 연결 정보 파일에서 가져올지 여부를 설정하며 앨리어스를 사용할 경우 오즈 서버의 <code>datasource.properties</code> 파일에 정의한 앨리어스 이름을 설정합니다. 앨리어스 정보: 파일 앨리어스 정보가 있는 데이터 셋 설정 정보 파일을 설정합니다.
데이터 보기	설정된 CSV 파일의 데이터를 미리보기합니다. 서버 로그 파일 이름이 설정되어 있으면 최대 행 수와 관계 없이 모든 행을 가져온 후 데이터 보기가 되므로 속도가 느려질 수 있습니다.
데이터 셋 정보 저장	CSV 파일의 데이터 셋 정보(필드 이름, 필드 타입)을 CSV 파일로 저장합니다. 해당 버튼은 데이터 셋을 수정할 경우에만 활성화되며, 데이터 셋 정보 저장 시 인코딩이 UTF-8로 설정됩니다.
원본 저장	데이터 셋의 원본 데이터를 파일로 저장합니다. [원본 저장] 버튼은 데이터 셋을 수정할 경우에만 활성화됩니다.

SOAP 스토어

SOAP(Simple Object Access Protocol)은 분산 환경에서 구조화된 정보를 교환하기 위한 경량 프로토콜 통신 규약입니다.

SOAP은 확장성 있는 메시지를 정의하기 위해 XML을 사용하며, 이 XML을 이용하여 객체(Object) 데이터끼리 손쉽게 교환할 수 있는 메시지 구조를 제공합니다.

OZ에서는 비즈니스 서비스를 기술하여 웹에서 비즈니스들끼리 접근하는 방법을 제공하기 위해 만든 WSDL(Web Service Description Language)을 통하여 웹 서비스를 사용할 수 있습니다.

스토어 추가

1. 스토어를 추가하려면 데이터 창 의 SOAP 스토어를 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 **[스토어 추가]** 메뉴를 클릭합니다.
2. SOAP 스토어 추가 다이얼로그에서 SOAP 스토어 이름, WSDL 위치를 설정한 후 **[서비스 얻어오기]** 버튼을 클릭하여 네임스페이스 목록, 서비스 목록을 표시한 후 **[확인]** 버튼을 클릭하면 SOAP 스토어에 추가된 데이터 셋 정보가 표시됩니다.

다이얼로그 옵션

SOAP 스토어 이름	SOAP 스토어 이름을 설정합니다.
WSDL 위치	WSDL 문서 위치를 설정합니다.
앨리어스	해당 데이터 셋 설정 정보를 별도의 연결 정보 파일에서 가져올지 여부를 설정하며 앨리어스를 사용할 경우 오즈 서버의 datasource.properties 파일에 정의한 앨리어스 이름을 설정합니다. 앨리어스 정보 파일: 앨리어스 정보가 있는 데이터 셋 설정 정보 파일을 설정합니다.
서비스 얻어오기	서비스에 사용된 네임스페이스 목록과 서비스 목록을 표시합니다. "WSDL 위치" 옵션 값을 입력하여야 해당 버튼이 활성화되며 입력한 WSDL 주소에 접속이 되어야만 네임스페이스 목록과 서비스 목록이 표시됩니다.
네임스페이스	WSDL의 타입에 정의된 Prefix와 NameSpace URI를 리스트 형태로 표시하며 사용하고자 하는 Prefix를 설정합니다.
서비스 목록	서비스 목록을 표시합니다.

	서비스: WSDL에서 제공하는 서비스 명이 콤보박스 리스트에 표시되며 사용하고자 하는 서비스를 설정합니다. 포트: 서비스에 속한 바인딩 목록이 콤보박스 리스트에 표시되며 사용하고자 하는 포트를 설정합니다. 오퍼레이션: 포트에 속한 바인딩 목록이 콤보박스 리스트에 표시되며 사용하고자 하는 Operation을 설정합니다. SOAP: 액션 선택된 Operation의 SOAP 액션 정보가 표시됩니다. 요청: Operation의 입력 정보인 이름, 타입, XPath, OZParam을 "#OZParam.패러미터 명#" 형태로 입력합니다. 요청 미리보기: 요청한 SOAP 메시지를 XML 형태로 표시하는 다이얼로그를 실행합니다. SOAP Header: SOAP Header를 입력할 수 있도록 다이얼로그를 실행합니다. 자동 매핑: 요청 옵션의 "OZParam" 값을 자동으로 설정하며 설정한 값은 "#OZParam.<operation name>_<name>#" 형태로 표시됩니다.
--	--

참고 사항

쿼리 디자이너에서 SOAP 스토어 기능을 사용하기 위해서는 SOAP Toolkit 3.0 라이브러리가 필요합니다. 또한 MSSOAP30.dll, WHSC30.dll 파일을 쿼리 디자이너가 설치된 경로의 bin/soap/register_soap.bat 파일을 실행하여 레지스트리에 등록하여야 합니다.

디자인 모드를 매뉴얼 모드에서 디자인 뷰 모드로는 변경할 수 없습니다.

자동으로 사용자 패러미터 필드 지정 시 XML 스키마의 "DateTime" 타입은 "VARCHAR" 타입으로 생성됩니다.

WSDL 정의의 메시지에 논리적 타입을 지정해 주기 위해서는 part와 type이 1:1 매핑이 되어야하나 n:1인 WSDL인 경우 첫 번째 part와 type이 1:1 매핑됩니다.

데이터 셋 추가

SOAP 스토어의 데이터 셋 추가 방법을 디자인 뷰 모드와 매뉴얼 모드로 나누어 설명합니다.

디자인 뷰 모드

1. 데이터 창에 SOAP 스토어를 마우스 오른쪽 버튼으로 클릭하면 나타나는 팝업 메뉴에서 **[데이터 셋 추가]** 메뉴를 클릭합니다. 데이터 셋 추가 다이얼로그에서 데이터 셋 이름을 입력한 후 **[확인]** 버튼을 클릭하여 데이터 셋을 추가하면 노드 트리 창에 SOAP 스토어의 노드 구조가 트리 모양으로 표시됩니다.

2. 노드 트리 창의 노드명 앞에는 노드 각각의 특성을 나타내는 아이콘이 있어 노드에 대한 식별을 용이하게 합니다. 노드의 아이콘에 대한 설명은 다음과 같습니다.

	애트리뷰트
	최소 발생 1, 최대 발생 1인 엘리먼트(기본 값)
	최소 발생 0, 최대 발생 제한 없는 엘리먼트
	최소 발생 0, 최대 발생 1인 엘리먼트
	최소 발생 1, 최대 발생 제한 없는 엘리먼트
	트리에서 확장된 엘리먼트
	필드로 선택 불가능한 엘리먼트
	기본 XPath로 지정된 엘리먼트

3. 기본 XPath를 설정합니다. 노드 트리 창에서 기본 XPath로 지정할 노드를 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 **[기본 XPath 지정]** 메뉴를 클릭합니다.
4. 기본 XPath로 지정된 노드는 아이콘이 체크된 상태로 표시되고 노드 이름이 하이라이트되어 표시되며, 기본 XPath에 지정한 기본 XPath가 표시됩니다.
5. 기본 XPath를 수정하려면 기본 XPath를 나타내는 텍스트 박스에서 다른 값으로 입력하거나 트리 창에서 기본 XPath로 변경할 노드 아이템을 마우스 오른쪽 버튼으로 선택하여 나타나는 **[기본 XPath 지정]** 메뉴를 클릭하여 수정할 수 있습니다.
6. 노드 트리 창에서 노드를 더블 클릭하여 데이터 셋의 필드를 추가합니다. 추가된 노드는 선택된 필드 창에서 확인할 수 있습니다.
7. 선택된 필드 창에서 필드의 이름을 변경하거나 [맨 위로], [위로], [아래로], [맨 아래로] 버튼을 클릭하여 순서를 변경하고, 잘못 추가된 필드는 [삭제] 버튼을 클릭하여 삭제합니다.
8. 데이터 셋에 필드를 모두 추가한 후 실행 아이콘을 클릭하면 추가된 필드가 표시됩니다.

매뉴얼 모드

1. 데이터 창의 SOAP 스토어를 마우스 오른쪽 버튼으로 클릭하면 나타나는 팝업 메뉴에서 **[데이터 셋 추가]** 메뉴를 클릭합니다. 데이터 셋 추가 다이얼로그에서 데이터 셋 이름을 입력한 후 **[확인]** 버튼을 클릭합니다.
2. **[쿼리] > [디자인 모드] > [매뉴얼]** 메뉴를 클릭하여 쿼리 창의 디자인 모드를 매뉴얼 모드로 변경합니다. 노드 트리 창에는 추가된 SOAP 스토어의 노드 구조가 트리 모양으로 표시됩니다.
3. 노드 트리 창의 노드명 앞에는 노드 각각의 특성을 나타내는 아이콘이 있어 노드에 대한 식별을 용이하게 합니다. 노드의 아이콘에 대한 설명은 다음과 같습니다.

	애트리뷰트
	최소 발생 1, 최대 발생 1인 엘리먼트(기본 값)
	최소 발생 0, 최대 발생 제한 없는 엘리먼트
	최소 발생 0, 최대 발생 1인 엘리먼트
	최소 발생 1, 최대 발생 제한 없는 엘리먼트
	트리에서 확장된 엘리먼트
	필드로 선택 불가능한 엘리먼트
	기준 XPath로 지정된 엘리먼트

4. 기준 XPath를 설정합니다. 노드 트리 창에서 기준 XPath로 지정할 노드를 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 **[기준 XPath 지정]** 메뉴를 클릭합니다.
5. 기준 XPath를 설정합니다. 기준 XPath에 기준 XPath를 직접 입력하거나 기준 XPath로 지정할 노드를 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 **[노드 경로 복사]** 메뉴를 클릭하여 노드 경로를 복사한 후 기준 XPath에 붙여넣기합니다.
6. 노드 트리 창에서 노드를 더블 클릭하여 데이터 셋의 필드를 추가합니다. 추가된 노드는 선택된 필드 창에서 확인할 수 있습니다.
7. 선택된 필드 창에서 필드의 이름을 변경하거나 [맨 위로], [위로], [아래로], [맨 아래로] 버튼을 클릭하여 순서를 변경하고, 잘못 추가된 필드는 [삭제] 버튼을 클릭하여 삭제합니다.
8. 데이터 셋에 필드를 모두 추가한 후 실행 아이콘을 클릭하면 추가된 필드가 표시됩니다.

ADO DataSet 웹서비스 스토어

ADO DataSet 웹서비스 스토어는 닷넷 웹 서비스에서 ADO DataSet 객체를 연동할 수 있는 스토어입니다.

스토어 추가

1. 스토어를 추가하려면 데이터 창에 ADO DataSet 웹서비스를 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 **[스토어 추가]** 메뉴를 클릭합니다.
2. ADO DataSet 웹서비스 추가 다이얼로그에서 ADO DataSet 웹서비스 스토어 이름과 WSDL 위치를 설정한 후 **[확인]** 버튼을 클릭하면 ADO DataSet 웹서비스에 추가된 ADO DataSet 웹서비스 스토어가 표시됩니다.

다이얼로그 옵션

스토어 이름	추가할 ADO DataSet 웹서비스 스토어 이름을 입력합니다.
WSDL 위치	WSDL 문서 위치를 설정합니다. 패러미터: 사용할 패러미터를 패러미터 마법사를 이용하여 설정합니다. [패러미터] 버튼을 클릭하면 패러미터 마법사를 이용하여 WSDL 위치를 설정할 수 있습니다

데이터 셋 추가

1. 데이터 셋을 추가하려면 추가된 스토어를 마우스 오른쪽 버튼으로 클릭하면 나타나는 팝업 메뉴에서 **[데이터 셋 추가]** 메뉴를 클릭합니다.
2. ADO 데이터 셋 추가 다이얼로그에서 ADO 데이터 셋 이름을 입력한 후 **[확인]** 버튼을 클릭하면 쿼리 디자이너의 쿼리 창이 ADO DataSet 웹서비스 데이터 셋 정보를 설정할 수 있도록 변경됩니다.
3. 데이터 셋 정보를 설정한 후 실행 아이콘을 클릭하면 ADO 데이터 셋 정보 다이얼로그가 표시되고 SET_NAME을 설정한 후 **[확인]** 버튼을 클릭하면 스토어 하위에 데이터 셋과 필드가 추가된 것을 확인할 수 있습니다.

다이얼로그 옵션

WebMethod	WSDL에 정의된 Web Method가 콤보박스에 리스트 형태로 표시됩니다.
-----------	--

OZParam	WSDL에 정의된 패러미터 이름을 표시하고 해당 패러미터 값을 설정합니다.
자동 매핑	"#OZParam.패러미터명#" 형태로 입력 패러미터 값을 매핑합니다.

XML 스토어

XML은 웹 상에서 서로 다른 업체의 응용프로그램 간의 데이터 교환을 위한 데이터 표현 언어로써, The World Wide Web Consortium(W3C)이 정의한 데이터를 응용 프로그램으로부터 분리하는 웹 언어입니다. 예를 들어, HTML이 웹 상에서 표준 UI를 구현하는 웹 언어라면 XML은 웹 상에서의 데이터 교환을 위한 웹 언어라고 할 수 있습니다.

DTD/XSD는 XML 문서에 존재할 수 있는 엘리먼트의 종류와 엘리먼트 간의 구조적 연관 관계 등을 나타내는 문법 규칙(Schema)을 정의하는 표준 중의 하나입니다. 따라서 DTD/XSD는 XML 문서에 존재할 수 있는 태그와 각 태그간의 포함 관계를 정의하고, 태그의 순서와 포함할 수 있는 태그의 개수를 정의합니다. 또한 태그가 가질 수 있는 속성과 그 속성이 갖는 값들을 정의합니다.

XML 표준은 XML 분석 단계에서 DTD/XSD와 같은 XML 구성 정보를 이용해 문서의 유효성을 검증함으로써 XML 데이터를 이용하는 응용 프로그램이 데이터를 검증해야 하는 번거로움을 제거합니다. 따라서 XML 데이터를 이용하는 응용프로그램은 XML 표준을 이용함으로써 다른 업체의 응용 프로그램과의 데이터 교환이 손쉬워지는 이점을 갖게 됩니다.

XML 스토어는 오즈가 제공하는 XML 기반 데이터 소스로서, DTD나 XSD가 없는 XML만을 지원하는 파일/HTTP 스토어와는 달리 DTD나 XSD를 이용하는 XML 파일을 데이터 소스로 이용할 수 있습니다. 따라서 XML 스토어는 지원하는 문법 규칙에 따라 XML DTD 스토어와 XML XSD 스토어로 구성됩니다.

스토어 추가

1. XML 스토어를 추가하려면 데이터 창의 **[XML]**을 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 **[XML 데이터 스토어 추가]** 메뉴를 클릭합니다.
2. XML 스토어 추가 다이얼로그의 각 옵션을 설정한 후 **[확인]** 버튼을 클릭하면 추가한 데이터베이스 정보가 표시됩니다.

다이얼로그 옵션

XML 스토어 이름	사용자 데이터 스토어 이름을 설정합니다.
XML 주소 (파일 경로 또는 URL)	서버에서 운용 시 사용할 XML 파일 경로 또는 사용자 애플리케이션 경로를 설정합니다. [패러미터] 버튼을 클릭하면 패러미터 마법사를 이용하여 XML 파일 경로를 설정할 수 있습니다.

	[경로] 버튼을 클릭하여 가져올 XML 파일을 선택합니다.
디자인시 사용할 임시 XML 파일 경로	디자인 시 사용할 XML 파일의 로컬 시스템의 파일 경로를 설정합니다. [패러미터] 버튼을 클릭하면 패러미터 마법사를 이용하여 디자인 시 사용할 임시 XML 파일 경로를 설정할 수 있습니다. [경로] 버튼을 클릭하여 가져올 XML 파일을 선택합니다.
디자인시 사용할 정의 파일 경로(DTD 또는 XSD)	디자인 시 사용할 DTD/XSD 파일의 로컬 시스템 파일 경로를 설정합니다. [패러미터] 버튼을 클릭하면 패러미터 마법사를 이용하여 디자인 시 사용할 파일 경로를 설정할 수 있습니다. [경로] 버튼을 클릭하여 가져올 XML 파일을 선택합니다.
XML 파서(Java)	XML 파서를 설정합니다. 자바 서버에서만 동작합니다. xerces: XML 파서를 xerces로 설정합니다. dom4j: XML 파서를 dom4j로 설정합니다.
서버 로그 파일 이름	파싱 에러 발생 시 XML 파일을 데이터 파일로 저장할 때 저장할 파일 이름을 설정합니다. 에러 발생 시 로그 파일 저장 기능은 XML 주소가 http:// 또는 https://이고, 서버 log.properties의 XMLSTORE_LOG_FILE 옵션이 MANUAL 또는 AUTO인 경우에만 적용됩니다. 앨리어스 옵션을 체크한 경우 서버 로그 파일 이름 옵션은 비활성화되며, 서버 로그 파일 이름은 "XML 스토어 추가 다이얼로그"에서 설정한 값이 아니라 datasource.properties의 alias.FILENAMETOLOG 옵션에서 설정한 값으로 적용됩니다. 실제 생성되는 로그 파일의 파일 이름은 서버 로그 파일 이름에 날짜, 시간, 일련번호를 조합하여 생성되고, 로그 파일의 경로는 서버 log.properties의

	XMLSTORE_LOG_FILE.LOCATION 옵션에서 설정한 값으로 적용됩니다.
URL 인코드 설정	Server URL 인코딩 유무를 설정합니다. 데이터 요청 시 URL의 패러미터 부분에 한국어나 기타 다국어 등이 포함되어 있을 경우 체크하시기 바랍니다.
포스트 타입 전송	URL 패러미터 전송 시 GET 방식을 이용할 것인지 POST 방식을 이용할 것인지 선택합니다. 옵션을 체크하면 POST 방식으로 전송됩니다.
앨리어스	해당 데이터 셋 설정 정보를 별도의 연결 정보 파일에서 가져올지 여부를 설정하며 앨리어스를 사용할 경우 오즈 서버의 datasource.properties 파일에 정의한 앨리어스 이름을 설정합니다. 앨리어스 정보: 파일 앨리어스 정보가 있는 데이터 셋 설정 정보 파일을 설정합니다.

네임스페이스 편집 다이얼로그

"디자인시 사용할 정의 파일 경로"에 설정한 DTD 파일에 정의된 Root Element가 없고, Root Element가 여러 개일 경우 **[확인]** 버튼을 클릭하면 Root Element 선택 다이얼로그가 표시됩니다.

데이터 셋 추가

XML 스토어의 데이터 셋 추가 방법을 디자인 뷰 모드와 매뉴얼 모드로 나누어 설명합니다.

디자인 뷰 모드

1. 데이터 창에 XML 스토어를 마우스 오른쪽 버튼으로 클릭하면 나타나는 팝업 메뉴에서 **[데이터 셋 추가]** 메뉴를 클릭합니다. 데이터 셋 추가 다이얼로그에서 데이터 셋 이름을 입력한 후 **[확인]** 버튼을 클릭하여 데이터 셋을 추가하면 노드 트리 창에 XML 스토어의 노드 구조가 트리 모양으로 표시됩니다.
2. 노드 트리 창의 노드명 앞에는 노드 각각의 특성을 나타내는 아이콘이 있어 노드에 대한 식별을 용이하게 합니다. 노드의 아이콘에 대한 설명은 다음과 같습니다.

	애트리뷰트
	최소 발생 1, 최대 발생 1인 엘리먼트(기본 값)

	최소 발생 0, 최대 발생 제한 없는 엘리먼트
	최소 발생 0, 최대 발생 1인 엘리먼트
	최소 발생 1, 최대 발생 제한 없는 엘리먼트
	트리에서 확장된 엘리먼트
	필드로 선택 불가능한 엘리먼트
	기준 XPath로 지정된 엘리먼트

3. 기준 XPath를 설정합니다. 노드 트리 창에서 기준 XPath로 지정할 노드를 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 **[기준 XPath 지정]** 메뉴를 클릭합니다.
4. 기준 XPath로 지정된 노드는 아이콘이 체크된 상태로 표시되고 노드 이름이 하이라이트되어 표시되며, 기준 XPath에 지정한 기준 XPath가 표시됩니다.
5. 이미 설정된 기준 XPath를 수정하려면 기준 XPath를 나타내는 텍스트 박스에서 다른 값으로 입력하거나 트리 창에서 기준 XPath로 변경할 노드 아이템을 마우스 오른쪽 버튼으로 선택하여 나타나는 **[기준 XPath 지정]** 메뉴를 클릭하여 수정할 수 있습니다.
6. 노드 트리 창에서 노드를 더블 클릭하여 데이터 셋의 필드를 추가합니다. 추가된 노드는 선택된 필드 창에서 확인할 수 있습니다.
7. 선택된 필드 창에서 필드의 이름을 변경하거나 [맨 위로], [위로], [아래로], [맨 아래로] 버튼을 클릭하여 순서를 변경하고, 잘못 추가된 필드는 [삭제] 버튼을 클릭하여 삭제합니다.
8. 데이터 셋에 필드를 모두 추가한 후 실행 아이콘을 클릭하면 추가된 필드가 표시됩니다.

매뉴얼 모드

1. 데이터 창 XML 스토어를 마우스 오른쪽 버튼으로 클릭하면 나타나는 팝업 메뉴에서 **[데이터 셋 추가]** 메뉴를 클릭합니다. 데이터 셋 추가 다이얼로그에서 데이터 셋 이름을 입력한 후 **[확인]** 버튼을 클릭합니다.
2. **[쿼리] > [디자인 모드] > [매뉴얼]** 메뉴를 클릭하여 쿼리 창의 디자인 모드를 매뉴얼 모드로 변경합니다. 노드 트리 창에는 추가된 XML 스토어의 노드 구조가 트리 모양으로 표시됩니다.
3. 노드 트리 창의 노드명 앞에는 노드 각각의 특성을 나타내는 아이콘이 있어 노드에 대한 식별을 용이하게 합니다. 노드의 아이콘에 대한 설명은 다음과 같습니다.

	�트리뷰트
	최소 발생 1, 최대 발생 1인 엘리먼트(기본 값)
	최소 발생 0, 최대 발생 제한 없는 엘리먼트

	최소 발생 0, 최대 발생 1인 엘리먼트
	최소 발생 1, 최대 발생 제한 없는 엘리먼트
	트리에서 확장된 엘리먼트
	필드로 선택 불가능한 엘리먼트
	기준 XPath로 지정된 엘리먼트

4. 기준 XPath를 설정합니다. 기준 XPath에 기준 XPath를 직접 입력하거나 기준 XPath로 지정할 노드를 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 **[노드 경로 복사]** 메뉴를 클릭하여 노드 경로를 복사한 후 기준 XPath에 붙여넣기합니다.
5. 노드 트리 창에서 노드를 더블 클릭하여 데이터 셋의 필드를 추가합니다. 추가된 노드는 선택된 필드 창에서 확인할 수 있습니다.
6. 선택된 필드 창에서 필드의 이름을 변경하거나 [맨 위로], [위로], [아래로], [맨 아래로] 버튼을 클릭하여 순서를 변경하고, 잘못 추가된 필드는 [삭제] 버튼을 클릭하여 삭제합니다.
7. 데이터 셋에 필드를 모두 추가한 후 실행 아이콘을 클릭하면 추가된 필드가 표시됩니다.

SAP 스토어

SAP은 데이터베이스를 이용하여 통합된 업무용 애플리케이션을 제공하며, 최근의 SAP R/3 시스템에서는 문서들을 자동으로 사용자들에게 전달하는 워크 플로우 기능까지 제공하고 있습니다. SAP은 기본적으로 클라이언트 서버 환경을 기본으로 하며, 대부분의 운영 체제 플랫폼을 지원합니다.

SAP의 데이터를 이용하여 보고서를 작성하기 위해서는 별도의 데이터 추출 프로그램 작성 및 데이터 추출 프로그램 호출을 위한 SAP 클라이언트 애플리케이션을 별도로 설치해야 하는 번거로움이 있습니다. SAP 스토어는 별도의 데이터 추출 프로그램 작성 없이 간단히 데이터 셋을 작성할 수 있도록 도와줍니다.

SAP 스토어에서 데이터를 전달받기 위해서는 RFC(Remote Function Call:시스템 밖에서 R/3에 접근하여 데이터를 얻을 수 있는 방법) 모듈을 참조하여 전달받는 결과값을 테이블 형태로 구성합니다.

스토어 추가

1. SAP 스토어를 추가하려면 데이터 창의 [SAP]를 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 **[SAP 스토어 추가]** 메뉴를 클릭합니다.
2. SAP 연결 정보 다이얼로그의 각 옵션을 설정한 후 **[확인]** 버튼을 클릭하면 추가한 SAP 스토어가 표시됩니다.

다이얼로그 옵션

SAP 스토어 이름	SAP 스토어 이름을 설정합니다.
클라이언트	로그인할 클라이언트 번호를 설정합니다.
사용자 ID	로그인 시 사용할 아이디를 설정합니다.
사용자 암호	로그인 시 사용할 암호를 설정합니다. 패러미터 사용: 사용자 패스워드를 사용자 지정 패러미터 값으로 설정합니다.
언어	로그인 시 사용할 언어를 설정합니다.
호스트 이름(주소)	SAP 서버의 IP Address를 설정합니다.
시스템 번호	SAP 서버의 시스템 번호를 설정합니다.
SAP 앨리어스	SAP 스토어 정보를 앨리어스로 설정할지 여부를 설정하며 앨리어스를 사용할 경우 오즈 서버의 sap.properties 파일에 정의한 앨리어스 이름을 설정합니다. 앨리어스 정보 파일: 앨리어스 정보가 있는 데이터 셋 설정 정보 파일을 설정합니다.

데이터 셋 추가

1. 데이터 셋을 추가하려면 추가된 스토어를 마우스 오른쪽 버튼으로 클릭하면 나타나는 팝업 메뉴에서 **[함수 추가]** 메뉴를 클릭합니다.
2. SAP FRC Set 설정 다이얼로그에서 텍스트 박스에 "*", "?"와 문자를 조합하여 입력한 후 **[검색]** 버튼을 클릭하거나 "모든 함수 리스트" 옵션을 체크한 후 **[검색]** 버튼을 클릭하면 다음과 같이 검색한 결과(RFC) 목록이 표시됩니다.
3. 검색 결과 목록에서 원하는 RFC 항목을 더블 클릭한 후 **[확인]** 버튼을 클릭합니다.
4. SAP RFC 패러미터 설정 다이얼로그에서 선택된 RFC에 대한 패러미터 값을 설정한 후 **[확인]** 버튼을 클릭합니다.
5. SAP RFC 데이터셋 이름 설정 다이얼로그에서 데이터셋 정보를 설정한 후 **[확인]** 버튼을 클릭하면 스토어 하위에 데이터 셋이 추가된 것을 확인할 수 있습니다.

CLEAR QUEST 스토어

CLEAR QUEST 스토어는 형상 관리 툴인 IBM Rational Clear Quest 프로그램에서 생성한 DB를 연동할 때 이용하는 데이터 소스입니다.

주의 사항

- CLEAR QUEST에서 지원하는 쿼리 방식 데이터 셋 추가는 현재 지원하지 않으며, 다이나믹 방식 데이터 셋의 필터 정의문은 표시되지 않습니다.
- CLEAR QUEST에 접속한 사용자에게 SQL 보기 권한이 없을 경우 디자인 모드를 매뉴얼 모드로 변경하여 사용할 수 없습니다.
- CLEAR QUEST 스토어를 사용하려면 사용자 PC에 IBM Rational Clear Quest 프로그램이 설치되어 있어야 합니다.
- CLEAR QUEST 스토어를 사용할 경우 데이터 창에 사용 가능한 테이블 리스트가 표시되지 않습니다.
- 디자인 모드를 매뉴얼 모드에서 디자인 뷰 모드로는 변경할 수 없습니다.

스토어 추가

CLEAR QUEST 스토어 추가는 디자인 뷰 모드와 매뉴얼 모드로 추가할 수 있습니다.

디자인 뷰 모드

1. CLEAR QUEST 스토어를 추가하려면 데이터 창의 [CLEAR QUEST]를 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 **[스토어 추가]** 메뉴를 클릭합니다.
2. CLEAR QUEST 스토어 추가 다이얼로그에서 CLEAR QUEST 스토어 이름, 사용자 이름, 패스워드, 데이터베이스를 설정한 후 **[확인]** 버튼을 클릭하면 추가한 스토어 정보가 표시됩니다.

매뉴얼 모드

1. CLEAR QUEST 스토어를 추가하려면 데이터 창의 [CLEAR QUEST]를 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 **[스토어 추가]** 메뉴를 클릭합니다.
2. CLEAR QUEST 스토어 추가 다이얼로그에서 CLEAR QUEST 스토어 이름을 입력한 후 "스키마 리파지토리 목록" 옵션의 **[목록 가져오기]** 버튼을 클릭하여 CLEAR QUEST에 추가된 스키마 리파지토리 목록을 표시합니다.

3. 표시된 스키마 리파지토리 목록 중 연결할 스키마를 더블 클릭하여 해당 스키마의 데이터베이스 목록을 가져옵니다. 사용자 이름, 패스워드, 데이터베이스를 설정한 후 **[확인]** 버튼을 클릭하면 추가한 스토어 정보가 표시됩니다.

데이터 셋 추가

CLEAR QUEST 스토어의 데이터 셋 추가는 디자인 뷰 모드와 매뉴얼 모드로 추가할 수 있습니다.

디자인 뷰 모드 - 레코드 타입

1. CLEAR QUEST 스토어에 데이터 셋 추가 시 레코드 타입은 CLEAR QUEST의 레코드 타입을 선택한 후 해당 레코드 타입과 연동하여 데이터 셋을 추가하는 방식입니다.
2. 데이터 창의 [CLEAR QUEST]에 추가된 스토어를 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 **[데이터 셋 추가]** 메뉴를 클릭한 후 CLEAR QUEST 셋 추가 다이얼로그에서 [레코드 타입] 탭을 선택합니다.
3. CLEAR QUEST 셋 추가 다이얼로그에서 데이터 셋 이름을 입력하고 연결할 레코드 타입을 선택한 후 **[확인]** 버튼을 클릭하면 데이터 셋이 추가되고, [디자인 뷰] 탭에 데이터 셋 정보를 설정할 수 있는 화면이 표시됩니다.
4. 사용 가능한 필드 창에서 필드를 선택한 후 필드 [추가]/[삭제] 버튼을 클릭하여 사용할 필드를 선택하고 필드 순서 이동 버튼([위로], [아래로])을 클릭하여 표시할 필드의 순서를 설정합니다.
5. 사용 가능한 필드 창에서 선택된 필터 창으로 필드를 선택한 후 필드 [추가]/[삭제] 버튼을 클릭하여 사용할 필드를 선택하고 연산자 버튼([OR], [AND])을 클릭하여 선택한 필드의 연산자를 변경합니다.
6. 필터를 설정할 때에는 값을 직접 입력하거나 패러미터 마법사를 이용하여 설정합니다.
7. 옵션을 설정한 후 툴바에서 **[쿼리문 실행 아이콘]**을 클릭하여 쿼리문을 실행합니다.

디자인 뷰 모드 - 워크스페이스

1. CLEAR QUEST 스토어에 데이터 셋 추가 시 워크스페이스는 CLEAR QUEST에 추가된 워크스페이스를 선택한 후 해당 워크페이스와 연동하여 데이터 셋을 추가하는 방식입니다.
2. 데이터 창의 [CLEAR QUEST]에 추가된 스토어를 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 **[데이터 셋 추가]** 메뉴를 클릭한 후 CLEAR QUEST 셋 추가 다이얼로그에서 [워크스페이스] 탭을 선택합니다.
3. CLEAR QUEST 셋 추가 다이얼로그에서 데이터 셋 이름을 입력하고 연결할 워크스페이스를 선택한 후 **[확인]** 버튼을 클릭하면 데이터 셋이 추가되고, [디자인 뷰] 탭에 선택한 워크스페이스에 설정된 필드 및 필터 정보가 화면에 표시됩니다.
4. 옵션을 설정한 후 툴바에서 **[쿼리문 실행 아이콘]**을 클릭하여 쿼리문을 실행합니다.

매뉴얼 모드

1. 데이터 창에 생성된 CLEAR QUEST]에 추가된 스토어를 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 **[데이터 셋 추가]** 메뉴를 클릭합니다.
2. CLEAR QUEST 셋 추가 다이얼로그에서 데이터 셋 이름을 입력한 후 **[확인]** 버튼을 클릭합니다.
3. 추가한 데이터 셋의 쿼리 창에 실행할 쿼리문을 직접 입력합니다.
4. 쿼리문을 입력한 후 [쿼리문 실행 아이콘]을 클릭하여 쿼리문을 실행합니다.

데이터 셋 디자인

본 장에서는 데이터 셋을 디자인하는 방법에 대해 설명합니다.

매뉴얼 모드로 쿼리 디자인

데이터 창에서 데이터 셋을 선택한 후 메뉴바에서 **[쿼리] > [디자인모드] > [매뉴얼]** 메뉴를 선택하거나 툴바에서 **[매뉴얼 아이콘]**을 클릭하면 쿼리 창이 매뉴얼 모드 쿼리 창으로 변경됩니다. 쿼리 창에 쿼리문을 직접 입력하여 쿼리를 디자인합니다.

쿼리문 작성

쿼리 디자인 모드를 매뉴얼 모드로 변경한 후 **[쿼리]** 탭을 선택하고 쿼리문을 직접 입력합니다.

쿼리문 실행

쿼리문을 입력한 후 메뉴바에서 **[실행] > [쿼리문 실행]** 메뉴를 선택하거나 툴바에서 **[쿼리문 실행 아이콘]**을 클릭 또는 키보드의 **<F5>** 키를 눌러 쿼리문을 실행합니다. 쿼리문에 에러가 없을 경우에는 쿼리 창이 **[결과]** 탭으로 선택되고 결과 데이터가 표시됩니다.

결과 필드 확인

쿼리가 제대로 실행되면 해당 데이터 셋 아래에 결과 필드가 표시됩니다.

참고 사항

쿼리문에 주석을 사용한 경우 주석 내용에 사용자 지정 패러미터, 사용자 지정 패러미터 구분자, 마스터셋의 필드 값이 포함되어 있으면 쿼리문이 정상 동작하지 않을 수 있으니 사용 시 주의하시기 바랍니다.

디자인 모드로 쿼리 디자인

디자인 뷰 모드는 SQL문을 정확하게 모르는 일반 개발자를 위해 마법사를 이용하여 손쉽게 쿼리문을 작성할 수 있도록 도와줍니다.

데이터에서 데이터 셋을 선택한 후 메뉴바에서 **[쿼리] > [디자인모드] > [디자인 뷰]** 메뉴를 선택하거나 툴바에서 **[디자인 뷰 아이콘]**을 클릭하면 쿼리 창이 디자인 뷰 모드 쿼리 창으로 변경됩니다.

디자인 뷰 모드의 **[디자인]** 탭은 화면 상단에 선택한 필드를 나타내는 필드 영역, 화면 중앙에 추가한 테이블을 나타내는 테이블 영역, 화면 하단에 쿼리 마법사를 이용하여 작성된 쿼리문을 표시하는 쿼리 영역으로 구성되어 있습니다.

디자인 뷰 모드의 **[디자인]** 탭의 화면에서 오른쪽 마우스 클릭 시 나타나는 **[쿼리 마법사]**는 디자인 뷰 모드에서 사용할 수 있는 쿼리문 제작 도구로서 데이터베이스 밴더별로 SQL 문법을 정확히 모르는 사용자도 단계별로 손쉽게 쿼리문을 작성할 수 있습니다.

테이블 선택

[디자인] 탭을 선택한 후 쿼리에 포함시키고자 하는 테이블을 데이터 창에 표시된 테이블 목록에서 선택하여 쿼리 창의 테이블 영역에 드래그 & 드롭합니다.

선택한 테이블은 쿼리 창의 테이블 영역에 표시되며, 쿼리문의 from절에 삽입됩니다.

조인 생성

테이블이 두 개 이상이면 조인 조건을 생성할 수 있습니다. 조인 조건은 자동으로 감지되어 생성되기도 하며, 수동으로 조인 조건을 추가할 수 있습니다.

조인 자동 설정

쿼리 디자이너의 옵션 중 "**조인 조건 자동 감지**" 옵션이 체크되어 있고 조인을 생성할 수 있는 필드 구조인 경우에는 두 개 이상의 테이블을 추가하면 자동으로 조인이 생성됩니다.

참고 사항

Oracle, Ms-SQL 등 Primary, Foreign Key 조합을 JDBC 함수를 통해 가져올 수 있는 경우 두 테이블의 필드가 Primary, Foreign Key로 연결되어 있으면 자동으로 조인이 생성됩니다.

ODBC 등 DB에서 Primary, Foreign Key 조합을 JDBC 함수를 통해 가져올 수 없는 경우 두 테이블의 필드 이름과 필드 타입이 같으면 자동으로 조인이 생성됩니다.

조인 수동 설정

수동으로 조인 조건을 추가하기 위해서는 테이블에서 원하는 필드를 선택한 다음 다른 테이블의 조인하고자 하는 필드로 드래그 & 드롭하여 연결합니다. 조인 유형을 바꾸기 위해서는 해당 조인 연결선을 선택하여 더블 클릭하거나 마우스 오른쪽 버튼을 클릭하여 나타나는 팝업 메뉴에서 **[조인 유형]** 메뉴를 클릭합니다. 조인 유형 다이얼로그를 이용하여 원하는 조인 유형으로 변경할 수 있습니다. 조인 연산자는 "같다", "작다", "작거나 같다", "크다", "크거나 같다", "다르다"로 설정할 수 있고, 타입은 "내부 조인", "왼쪽 내부 조인", "오른쪽 외부 조인", "전체 외부 조인"으로 설정할 수 있습니다.

조인 삭제

생성된 조인 조건을 삭제하기 위해서는 해당 조인 연결선을 선택한 후 키보드의 **<Delete>** 키를 누르거나 조인 연결선에서 마우스 오른쪽 버튼을 클릭하여 나타나는 팝업 메뉴에서 **[조인 해제]** 메뉴를 클릭합니다.

참고 사항

조인 조건 자동 감지 기능의 사용 여부는 메뉴바의 **[파일] > [옵션]** 메뉴를 클릭하여 나타나는 옵션 다이얼로그의 **[데이터베이스 설정] > [일반]** 탭에서 확인할 수 있습니다.

필드 설정

필드 추가

데이터 셋에 포함하고자 하는 필드를 테이블 영역에 추가된 테이블 다이얼로그에서 선택한 후 필드 영역으로 드래그 & 드롭하거나 더블 클릭하면 필드 영역으로 추가됩니다. **<Shift>** 또는 **<Ctrl>** 키를 이용하여 여러 개의 필드를 한꺼번에 추가할 수 있습니다.

필드 삭제

추가된 필드를 삭제하려면 필드 영역에서 필드를 선택한 후 키보드의 **<Delete>** 키를 누릅니다.

정렬 선택

쿼리 마법사를 이용하여 각 필드의 정렬 방식을 선택합니다. 메뉴바의 **[쿼리] > [쿼리 마법사]** 메뉴를 클릭하거나 쿼리 창에서 마우스 오른쪽 버튼을 클릭하여 나타나는 팝업 메뉴에서 **[쿼리 마법사]** 메뉴를 클릭합니다.

쿼리 마법사에 **[정렬 선택]** 탭을 선택하여 각 필드의 정렬 방식을 설정합니다. 사용 가능한 필드 목록에서 정렬 기준이 되는 필드를 선택하여 추가하고, 오름차순 또는 내림차순으로 정렬 방식을 설정하고, 정렬 순서도 변경합니다.

쿼리문 확인

작성된 쿼리는 쿼리 창의 쿼리 영역에 반영되어 표시됩니다.

테이블 이름 Rename 규칙

데이터 소스가 MDB, MS-SQL, MySQL 등일 때 테이블 이름에 공백 또는 숫자 등이 있을 경우 Rename되는 규칙은 다음과 같습니다.

- 테이블 이름에 공백이 있는 경우 ex) Car Sales -> [Car Sales].필드명
- 테이블 이름이 숫자로 시작되는 경우 ex) 9Sales -> [9Sales].필드명
- 테이블 이름과 필드 이름이 같은 경우 ex) Sales.Sales -> [Sales].Sales

쿼리문 실행

메뉴바의 **[실행] > [쿼리문 실행]** 메뉴를 클릭하거나 툴바에서 **[쿼리문 실행 아이콘]**을 클릭 또는 키보드의 **<F5>** 키를 눌러 쿼리문을 실행합니다.

쿼리 창이 **[결과]** 탭으로 선택되면서 결과 데이터가 표시되고, 해당 데이터 셋 아래에 결과 필드가 표시됩니다.

마스터-디테일 데이터 셋 디자인

하나 이상의 쿼리 데이터 셋은 마스터-디테일 관계로 연결될 수 있습니다.

- 마스터 셋 고객 등급(MemberGrade)
- 디테일 셋 고객 정보(Customer)

쿼리 마법사를 이용하여 고객 등급을 그룹핑한 결과 데이터 셋을 생성합니다.

작성된 쿼리문은 다음과 같습니다.

```
SELECT customer.member_card
FROM customer
GROUP BY customer.member_card
```

쿼리 마법사를 이용하여 고객 등급에 맞게 고객 정보 데이터를 조회하는 데이터 셋을 생성합니다.

작성된 쿼리문은 다음과 같습니다.

```
SELECT customer.customer_id, customer.fname, customer.lname, customer.phone1, customer.gender
FROM customer
WHERE customer.member_card = '#MemberGrade.member_card#'
```

데이터 창에서 디테일 셋으로 설정할 데이터 셋(Customer)을 선택한 후 속성 창에서 마스터셋 속성 값을 마스터셋으로 설정할 데이터 셋 이름(MemberGrade)으로 설정합니다.

다이나믹 데이터 셋 디자인

다이나믹 쿼리

다이나믹 쿼리란 데이터에 대한 필드 정보 등이 운영 시점에 동적으로 결정되도록 하는 쿼리를 말합니다. 이는 동적으로 변하는 필드 정보로 인해 디자인 시점에 필드 정보를 설정할 수 없는 경우에 다이나믹 쿼리를 이용해 구현할 수 있습니다. 따라서 사용자가 표시할 필드나 조건까지 선택해야 하는 경우 다이나믹 쿼리를 사용할 수 있습니다.

다이나믹 데이터 셋 개요

다이나믹 데이터 셋이란 데이터베이스, 사용자 데이터, 파일/HTTP 스토어, XML DTD 스토어를 통해 가져온 데이터를 다이나믹 쿼리 리포트를 구성하기 위한 형태로 변경해 주는 2차 데이터 셋입니다.

다이나믹 데이터 셋은 쿼리를 통해서 만들어지는 동적인 필드 정보를 저장하는 대신 모든 쿼리 결과를 ROWIDX, COLUMN, VALUE 세가지 고정 항목으로 저장하고, 실행 시점에 이들 세가지 고정 항목으로부터 동적인 필드 정보를 찾아내 재구성하는 기능을 제공합니다.

따라서 오즈 개발자는 다이나믹 데이터 셋의 세가지 고정 필드를 이용해 보고서를 디자인하고, 보고서가 바인딩될 때 동적으로 생성되는 쿼리 결과로 보고서를 자동 확장하도록 할 수 있습니다.

다이나믹 데이터 셋이 생성하는 세 가지 항목의 의미는 다음과 같습니다.

ROWIDX	실제 데이터 셋의 레코드 번호
COLUMN	실제 데이터 셋의 필드 이름
VALUE	해당 필드의 데이터 값

예를 들어 테이블 1과 같은 데이터 셋에 다이나믹 데이터 셋을 적용하면 테이블 2와 같이 ROWIDX, COLUMN, VALUE 필드를 갖는 새로운 데이터 pt이 생성되는 것을 알 수 있습니다. 다시 말해, 필드와 해당 정보를 나열하는 일반 데이터 셋에 다이나믹 데이터 셋을 적용하면, Row Index 별로 필드 이름과 그 값을 순서대로 저장하는 형태의 새로운 데이터 셋이 생성됩니다.

• 테이블 1 - 원본

Region	Sales2000	Sales2001
캐나다	157000	890235
동부	5234991	5432123

중서부	1234789	1123454
남부	4563300	4992100
서부	3551090	3901234

• 테이블 2 - 다이나믹 데이터 셋을 이용하여 만든 테이블

ROWIDX	COLUMN	VALUE
0	Region	캐나다
0	Sales2000	257000
0	Sales2001	890235
1	Region	동부
1	Sales2000	5234991
1	Sales2001	5432123
2	Region	중서부
2	Sales2000	1234789
2	Sales2001	1123454
3	Region	남부
3	Sales2000	4563300
3	Sales2001	4992100
4	Region	서부
4	Sales2000	3551090
4	Sales2002	3901234

다이나믹 데이터 셋 추가

- 데이터 창 의 **[데이터베이스]**를 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 **[스토어 추가]** 메뉴를 클릭하여 데이터베이스 연결 정보 추가 다이얼로그에 데이터베이스 연결 정보를 설정하여 데이터베이스 스토어를 추가합니다.
- 추가된 데이터베이스 마우스 오른쪽 버튼으로 클릭하면 나타나는 팝업 메뉴에서 **[다이나믹 데이터 셋 추가]** 메뉴를 클릭합니다.

3. 다이나믹 데이터 셋 추가 다이얼로그에서 데이터 셋 이름과 시작 행 번호를 입력한 후 **[확인]** 버튼을 클릭하여 다이나믹 데이터 셋을 추가합니다.
4. 쿼리 디자이너의 창을 디자인 뷰 또는 매뉴얼 모드로 변경하여 쿼리문을 디자인합니다.
5. **[쿼리문 실행 아이콘]**을 클릭하여 쿼리를 실행하고 다이나믹 데이터 셋에 추가된 필드 정보를 확인합니다. 다이나믹 데이터 셋의 필드는 실제 테이블의 필드명이 아닌 ROWIDX, COLUMN, VALUE로 추가됩니다.

멀티 데이터 셋 디자인

1. 데이터 창에서 추가된 데이터베이스 스토어를 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 **[멀티 데이터 셋 추가]** 메뉴를 클릭합니다.
2. 멀티 데이터 셋 추가 다이얼로그에서 멀티 데이터 셋 이름을 입력한 후 **[확인]** 버튼을 클릭합니다.
3. 쿼리 디자이너의 창을 디자인 뷰 또는 매뉴얼 모드로 변경하여 쿼리문을 디자인합니다.
4. 작성된 쿼리문은 다음과 같습니다.

```
SELECT * FROM Customers
```

```
SELECT * FROM Employees
```

5. **[실행] > [쿼리문 실행]** 메뉴를 클릭하거나 툴바에서 **[쿼리문 실행 아이콘]**을 클릭하여 쿼리를 실행합니다.

멀티 데이터 셋 쿼리문 입력 방법

멀티 데이터 셋의 쿼리 창에 프로시저나 여러 개의 쿼리문을 입력하여 멀티 데이터 셋을 생성합니다. 여러 개의 쿼리문을 입력하는 방법은 다음과 같습니다.

- **방법 1. 하나의 쿼리문을 입력한 후 공백으로 구분하여 다음 쿼리문을 입력합니다.**

```
SELECT * FROM Customers SELECT * FROM Employees
```

- **방법 2. 쿼리문을 입력한 후 줄바꿈하여 다음 쿼리문을 입력합니다.**

```
SELECT * FROM Customers
```

```
SELECT * FROM Employees
```

참고 사항

하위 데이터 셋의 마스터셋 속성 값은 따로 설정할 수 없으며 멀티 데이터 셋의 마스터셋 속성 값을 따릅니다.

멀티 데이터 셋은 매뉴얼 모드에서만 추가할 수 있으며, MSSQL, SYBASE만 지원합니다.

프로시저 데이터 셋 디자인

데이터 셋에서 프로시저를 설정하여 복수 개의 결과 셋을 생성할 수 있습니다.

1. 데이터 창의 **[데이터베이스]**를 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 **[스토어 추가]** 메뉴를 클릭합니다.
2. 데이터베이스 연결 정보 추가 다이얼로그의 각 옵션을 설정한 후 **[확인]** 버튼을 클릭하여 스토어를 추가합니다.
3. 추가된 데이터베이스 스토어를 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 **[프로시저 데이터 셋 추가]** 메뉴를 클릭합니다.
4. 프로시저 데이터 셋 다이얼로그에서 데이터 셋 이름을 입력하고 스키마를 선택한 후 **[조회]** 버튼을 클릭합니다. 결과 창에서 프로시저를 선택하고 **[다음]** 버튼을 클릭합니다.
5. "입력" 옵션의 "값"을 입력하고 출력할 패러미터를 선택한 후 **[마침]** 버튼을 클릭하면 스토어 하위에 데이터 셋이 추가된 것을 확인할 수 있습니다.

하위 데이터 셋 이름 생성 규칙

프로시저 데이터 셋의 하위 셋 이름은 프로시저 데이터 셋 다이얼로그에서 설정한 프로시저 데이터 셋 이름에 **_#**을 추가하여 자동으로 설정됩니다.

예를 들어, 프로시저 데이터 셋 다이얼로그에서 프로시저 데이터 셋 이름을 MULTISSET으로 설정한 경우 MULTISSET_1, MULTISSET_2, ... 형태로 설정됩니다. 데이터 타입이 CURSOR이나 RESULTSET인 경우에만 출력 패러미터 이름이 데이터 셋 이름으로 설정되고, 그 외 타입은 선택된 프로시저 이름이 데이터 셋 이름으로 설정됩니다.

자동 매핑

[자동 매핑] 버튼 클릭 시 "#OZParam.패러미터명#" 형태로 입력 패러미터 값을 매핑합니다.

입력, 출력 설정

실행할 프로시저의 패러미터를 설정하려면 입력 항목에 설정하고, 패러미터 값 입력 시 사용자가 직접 입력하거나 "OZParam" 값을 설정할 수 있으며, "널 적용" 옵션이 True일 경우에는 값을 입력할 수 없습니다. 실행할 프로시저의 결과 값을 가져올 패러미터를 설정하려면, 출력 항목에 설정합니다.

사용자 지정 패러미터 추가

1. 데이터 창 의 **[사용자 지정 패러미터]**를 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 **[패러미터 설정]** 메뉴를 클릭합니다.
2. 사용자 지정 패러미터 다이얼로그에서 Member에 패러미터 이름을 입력하고 속성을 설정한 후 **[확인]** 버튼을 클릭하면 데이터 창에 사용자 지정 패러미터가 추가된 것을 확인할 수 있습니다.

다이얼로그 옵션

이름	추가할 패러미터 이름을 설정합니다.
타입	추가할 패러미터 타입을 설정합니다.
값	추가할 패러미터의 기본 값을 설정합니다.
세션 키	추가할 패러미터의 값을 설정할 세션 키를 설정합니다. 추가할 패러미터의 값을 오즈 뷰어 패러미터에서 설정하는 대신 서버 세션으로 설정할 때 사용합니다.
설명	추가할 패러미터의 설명을 설정합니다.

참고 사항

쿼리문에서 사용자 지정 패러미터를 이용하여 조건문을 설정할 경우 쿼리문 실행 시 패러미터 입력 다이얼로그가 표시되며 사용자 지정 패러미터 다이얼로그에서 해당 패러미터 값이 설정되어 있으면 패러미터 입력 다이얼로그에서 설정한 패러미터 값이 표시되고 설정되어 있지 않으면 빈 값으로 표시됩니다.

수식 필드 추가

생성된 데이터 셋에 수식 필드를 추가할 수 있으며, 쿼리 데이터 셋, UDS 데이터 셋, XML 데이터 셋, CSV 데이터 셋, 다이내믹 데이터 셋, SAP 데이터 셋에서 추가할 수 있습니다.

1. 데이터 창의 **[데이터베이스]**에 생성된 데이터 셋을 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 **[수식 필드 추가]** 메뉴를 클릭합니다.
2. 수식필드 추가 다이얼로그에서 필드 이름을 입력하고, 스크립트 타입을 선택한 후 **[확인]** 버튼을 클릭합니다.
3. 스크립트 창에 스크립트를 입력하면 데이터 창에 수식 필드가 추가된 것을 확인할 수 있습니다.

참고 사항

쿼리문에서 수식 필드를 이용하여 조건문을 설정할 경우 쿼리문 실행 시 패러미터 입력 다이얼로그에서 수식 필드 값이 항상 빈 값으로 표시됩니다.

INCLUDE ODI로 데이터 셋 디자인

리파지토리 서버 혹은 로컬 작업 폴더에 저장된 ODI 파일을 가져와서 현재 편집중인 ODI 파일에 추가하여 사용할 수 있습니다.

INCLUDE ODI 데이터 셋 추가

1. 데이터 창의 **[INCLUDE ODI]**를 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 **[추가]** 메뉴를 클릭합니다.
2. 리파지토리 파일 다이얼로그에서 추가할 ODI 파일을 선택한 후 **[확인]** 버튼을 클릭하면 아래와 같이 ODI 파일 추가 다이얼로그가 나타납니다. Include 필드 이름을 입력하고 **[확인]** 버튼을 클릭합니다.
3. 삽입되는 ODI 파일에 생성되어 있는 스토어 및 데이터 셋, 필드, 사용자 지정 패러미터 등이 Include 필드로 추가됩니다. Include 필드에 의해 추가된 스토어 및 데이터 셋, 필드 등은 수정할 수 없습니다.

INCLUDE ODI 데이터 셋 삭제

1. 삭제할 Include 필드를 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 **[ODI 파일 삭제]** 메뉴를 클릭합니다.
2. 해당 Include 필드의 삭제 여부를 확인하는 창이 나타나면 **[예]** 버튼을 클릭합니다. Include 필드로 추가되었던 스토어 및 데이터 셋, 필드, 사용자 지정 패러미터가 모두 삭제됩니다.

제약 사항

Include ODI를 이용하여 현재 편집 중인 ODI 파일에 새로운 ODI 파일을 추가할 경우 다음과 같은 제약 사항이 있습니다.

- 현재 편집 중인 ODI와 같은 데이터 소스를 사용하는 ODI는 추가할 수 없습니다.
- 현재 편집 중인 ODI와 추가하려는 ODI의 스토어명, 데이터 셋명, 필드명이 중복되는 경우에는 추가할 수 없습니다.
- ODI 추가는 한 단계로만 가능합니다. 즉, A.odi 파일이 추가된 B.odi 파일을 C.odi에 추가할 경우 C.odi에는 A.odi의 데이터 셋 정보는 추가되지 않고, B.odi의 고유한 데이터 셋 정보만 추가됩니다.

컴파일된 질의문 디자인

컴파일된 질의문은 JDBC에서 쿼리 실행 시 "PreparedStatement"를 사용한다는 개념으로, 해당 데이터 셋의 속성 중 "컴파일된 질의문 사용" 속성 값을 "True"로 설정한 경우에 적용됩니다.

컴파일된 질의문을 사용할 경우 한 번 컴파일된 질의문을 재사용하여 실행하므로, 같은 질의문이 여러 번 반복하여 실행될 경우 처리 속도를 높여줍니다. 컴파일된 질의문은 각 데이터베이스에서 제공하는 컴파일된 질의문을 사용하여 쿼리를 수행합니다.

컴파일된 질의문을 디자인 시에 오즈 패러미터를 이용하여 쿼리문을 작성할 경우에는 데이터 셋의 "스크립트 사용"와 "컴파일된 질의문 사용" 속성 값에 따라 각각 다르게 디자인하여야 합니다.

스크립트 사용 = "False", 컴파일된 질의문 사용 = "True"/"False"인 경우

#OZParam.패러미터명# 또는 '#OZParam.패러미터명#' 와 같은 오즈 패러미터 예약어를 독립적으로 사용합니다.

• 오즈 패러미터가 숫자형인 경우

```
SELECT * FROM 테이블명
WHERE 조건필드명 = #OZParam.패러미터명#
```

• 오즈 패러미터가 숫자형이 아닌 경우

```
SELECT * FROM 테이블명
WHERE 조건필드명 = '#OZParam.패러미터명#'
```

스크립트 사용 = "True", 컴파일된 질의문 사용 = "True"인 경우

#OZParam.패러미터명# 또는 '#OZParam.패러미터명#' 와 같은 오즈 패러미터 예약어를 독립적으로 사용할 수 없고, 스크립트 문자열 내에서만 사용 가능합니다.

• 오즈 패러미터가 숫자형인 경우

```
string sql;
sql = "SELECT * FROM 테이블명 WHERE 조건필드명 = #OZParam.패러미터명#";
setQueryString(sql);
```

• 오즈 패러미터가 숫자형이 아닌 경우

```
string sql;
sql = "SELECT * FROM 테이블명 WHERE 조건필드명 = '#OZParam.패러미터명#' ";
setQueryString(sql);
```

스크립트 사용 = "True", 컴파일된 질의문 사용 = "False"인 경우

#OZParam.패러미터명# 또는 '#OZParam.패러미터명#' 와 같은 오즈 패러미터 예약어를 독립적으로 사용하여야 하며, 이때 숫자형이 아닌 오즈 패러미터를 사용할 경우 홑따옴표는 이중따옴표로 묶어야 제대로 동작됩니다.

- 오즈 패러미터가 숫자형인 경우

```
string sql;  
sql = "SELECT * FROM 테이블명 WHERE 조건필드명 = " + #OZParam.패러미터명#;  
setQueryString(sql);
```

- 오즈 패러미터가 숫자형이 아닌 경우

```
string sql;  
sql = "SELECT * FROM 테이블명 WHERE 조건필드명 = " + "'" + #OZParam.패러미터명# + "'";  
setQueryString(sql);
```

오즈 쿼리 컴포넌트

본 장에서는 오즈 쿼리 디자이너에서 지원되는 오즈 데이터 트리, 데이터 스토어, 데이터 셋, 데이터 필드, 사용자 지정 패러미터의 종류와 각 속성에 대해 살펴보겠습니다.

오즈 데이터 트리

오즈 데이터 트리는 오즈 쿼리 디자이너에서 연결할 수 있는 데이터 스토어, 데이터 셋, 사용자 지정 패러미터 등을 나타낸 것으로, 오즈 쿼리 디자이너의 데이터 창에 트리 형태로 표시됩니다.

Name	Description
널 데이터 무시	데이터 값이 null인 경우 null 데이터 무시 여부
사용자 지정 패러미터 구분자	사용자 지정 패러미터 구분자
Concurrent 전송 크기	Concurrent 전송 크기
서버 연결 위치	서버 연결 위치
데이터서비스 허용	데이터 서비스 기능 허용 여부
소수점 앞에 0 표시	데이터 값이 1 보다 작을 경우 소수점 앞에 0을 표시할지 여부
스크립트 에러 무시	스크립트 에러 발생 시 에러 메시지 표시 여부

널 데이터 무시

데이터 값이 null인 경우 null 데이터 무시 여부

Values

True	문자형일 경우는 ""(빈 문자)로, 숫자형일 경우는 0으로 처리
False	데이터 값이 null인 경우 데이터 값을 "<NULL>"로 표시

Reference

쿼리 디자이너 뿐만 아니라 리포트에서 null 데이터를 null 값 그대로 사용하려면 속성 값을 False로 설정하고, null 데이터를 "" 또는 0으로 변경하려면 속성 값을 True로 설정하시기 바랍니다.

참고로, 리포트에서 null 데이터를 사용하려면 쿼리 디자이너에서 "널 데이터 무시"를 False로 설정하고, 리포트 디자이너에서도 데이터 트리의 "Null 데이터 무시"를 False로 설정하여야 합니다.

사용자 지정 패러미터 구분자

사용자 지정 패러미터 구분자

Reference

기본 값은 "#"입니다.

패러미터를 구분하는 구분자를 설정합니다. 예를 들어, 구분자가 "#"으로 지정된 경우 패러미터는 #OZParam.패러미터이름# 형식으로 설정됩니다.

디자인 모드가 매뉴얼이고 데이터 셋의 속성 값 중 "스크립트 사용"이 "True"인 경우 구분자로 디폴트 값인 "#"만 사용할 수 있습니다.

사용자 지정 패러미터는 한 글자만 입력하여야 정상적으로 동작합니다.

알파벳, 숫자 또는 1byte 이상의 문자는 구분자로 사용할 수 없으며 RDB일 경우 ^ 또는 DB 연결 정보에 포함되어 있는 문자열도 구분자로 사용할 수 없습니다. 또한, 사용자 지정 패러미터 이름에 사용자 지정 패러미터 구분자로 설정한 문자가 있을 경우 쿼리 실행 시 에러가 발생합니다. 그러므로 사용자 지정 패러미터 구분자 설정 시 주의하시기 바랍니다.

See Also

[OZParamMark](#)

Concurrent 전송 크기

Concurrent 전송 크기

Reference

클라이언트에서 데이터 요청 시 한꺼번에 가져올 결과 데이터 셋의 크기를 Byte 단위로 설정합니다.

설정된 값만큼의 버퍼 단위로 데이터를 내보냅니다.

속성 값을 "0"으로 설정하면 서버에 설정된 값이 적용됩니다.

이 속성 값은 쿼리 디자이너에서 쿼리문을 실행할 때에는 적용되지 않으며, 리포트 뷰어에서 데이터를 요청할 경우에만 적용되는 속성입니다.

See Also

[ConcurrentFetchSize](#)

서버 연결 위치

서버 연결 위치

Values

Default	쿼리 디자이너 환경 설정 중 "서버 연결 위치" 옵션 값에 따라 연결할 서버를 설정 (기본 값)
OZServer	현재 연결된 리파지토리 서버로 서버를 설정
Local	로컬 서버로 설정

Reference

서버 연결 위치는 오즈 데이터 트리의 "서버 연결 위치" 속성 값과 쿼리 디자이너의 환경 설정 옵션 중 "서버 연결 위치" 옵션 값에 따라 다르게 설정될 수 있습니다.

환경 설정 옵션 값	오즈 데이터 트리 속성 값	설명
OpenType	Default	ODI 파일을 오픈한 위치가 로컬인지 리파지토리 서버인지에 따라 바인딩되는 서버를 선택합니다. 즉, 로컬에서 오픈한 경우에는 로컬 서버로, 리파지토리 서버에서 오픈한 경우에는 리파지토리 서버로 선택됩니다.
	OZServer	연결되어 있는 리파지토리 서버로 선택됩니다. 리파지토리 서버가 연결되지 않았을 경우에는 자동으로 로컬 서버로 선택됩니다.
	Local	무조건 로컬 서버로 선택됩니다.
Local	Default	무조건 로컬 서버로 선택됩니다.
	OZServer	연결되어 있는 리파지토리 서버로 선택됩니다. 리파지토리 서버가 연결되지 않았을 경우에는 자동으로 로컬 서버로 선택됩니다.
	Local	무조건 로컬 서버로 선택됩니다.
OZServer	Default	연결되어 있는 리파지토리 서버로 선택됩니다. 리파지토리 서버가 연결되지 않았을 경우에는 자동으로 로컬 서버로 선택됩니다.
	OZServer	연결되어 있는 리파지토리 서버로 선택됩니다. 리파지토리 서버가 연결되지 않았을 경우에는 자동으로 로컬 서버로 선택됩니다.
	Local	무조건 로컬 서버로 선택됩니다.

ForceLocal	무조건 로컬 서버로 선택됩니다.
ForceOZServer	연결되어 있는 리파지토리 서버로 선택됩니다. 리파지토리 서버가 연결되지 않았을 경우에는 자동으로 로컬 서버로 선택됩니다.
ForceOpenType	ODI 파일을 오픈한 위치가 로컬인지 리파지토리 서버인지에 따라 바인딩되는 서버를 선택합니다. 즉, 로컬에서 오픈한 경우에는 로컬 서버로, 리파지토리 서버에서 오픈한 경우에는 리파지토리 서버로 선택됩니다.

데이터서비스 허용

데이터 서비스 기능 허용 여부

Values

Default	오즈 서버의 ozservice.properties 파일에 설정된 dataservice_defaultType 옵션 값에 따름 (기본 값)
True	데이터 서비스 허용
False	데이터 서비스 거절

소수점 앞에 0 표시

데이터 값이 1 보다 작을 경우 소수점 앞에 0을 표시할지 여부

Values

Default	DB 서버에 설정된 속성 값에 따라 소수점 앞에 0을 표시 (기본 값)
True	DB 서버에 설정된 속성 값에 상관없이 소수점 앞에 무조건 0을 표시
False	DB 서버에 설정된 속성 값에 상관없이 소수점 앞에 무조건 0을 표시하지 않음

Reference

해당 기능은 데이터 타입이 "Numeric" 또는 "Decimal"인 데이터에서만 지원됩니다.

See Also

스크립트 에러 무시

스크립트 에러 발생 시 에러 메시지 표시 여부

Values

True	에러 메시지 표시
False	에러 메시지를 표시하지 않음 (기본 값)

Reference

True로 설정하였을 경우 스크립트 에러 발생 시 에러 메시지가 표시되지 않습니다.

오즈 데이터 스토어

- 데이터베이스 스토어
- 사용자 데이터 스토어
- 그룹 데이터 스토어
- SOAP 스토어
- AOD DataSet 웹 서비스 스토어
- XML 스토어
- SAP 스토어
- CLEAR QUEST 스토어

데이터베이스 스토어

JDBC 또는 ODBC를 통한 데이터베이스 연결 및 데이터 소스 정보를 관리합니다. [데이터베이스] 항목은 다수의 데이터베이스 연결 정보를 설정할 수 있으며, [데이터베이스 스토어] > [데이터베이스 데이터 셋] > [필드]의 계층 구조를 가지고 있습니다.

Name	Description
이름	데이터베이스 스토어 이름
사용자 이름	데이터베이스 사용자 이름
JDBC 드라이버	데이터베이스 스토어 연결을 위한 ODBC/JDBC 드라이버 이름
앨리어스 정보파일 사용	데이터베이스 스토어 연결 정보를 별도의 연결 정보 파일에서 가져올 것인지 여부
앨리어스 이름	데이터베이스 연결 정보 파일에서 원하는 앨리어스 이름
앨리어스 정보파일 이름	데이터베이스 연결 정보 파일 이름
자동 커밋	자동 커밋(Commit)할 것인지 여부
벤더	데이터베이스 벤더 종류
데이터 액션 클래스	데이터 액션 시 사용할 클래스 이름
데이터 액션 초기화 패러미터	데이터 액션 초기화 패러미터

이름

데이터베이스 스토어 이름

See Also

[GetName](#)

사용자 이름

데이터베이스 사용자 이름

JDBC 드라이버

데이터베이스 스토어 연결을 위한 ODBC/JDBC 드라이버 이름

앨리어스 정보파일 사용

데이터베이스 스토어 연결 정보를 별도의 연결 정보 파일에서 가져올 것인지 여부

Values

True	데이터베이스 스토어 연결 정보를 별도의 연결 정보 파일에서 가져옴
False	데이터베이스 스토어 연결 정보를 별도의 연결 정보 파일에서 가져오지 않음

See Also

[UseConnectionPoolAlias](#)

앨리어스 이름

데이터베이스 연결 정보 파일에서 원하는 앨리어스 이름

Reference

데이터베이스 앨리어스는 데이터베이스 연결 정보 파일에서 설정한 여러 개의 데이터베이스 연결을 구분해주는 이름입니다. 오즈 서버에서 실행하기 위해서는 해당 앨리어스 이름이 서버의 db.properties 파일 안에 명시되어 있어야 합니다.

See Also

[ConnectionPoolAlias](#)

앨리어스 정보파일 이름

데이터베이스 연결 정보 파일 이름

자동 커밋

자동 커밋(Commit)할 것인지 여부

Values

True	자동 커밋(Commit)을 함
False	자동 커밋(Commit)을 하지 않음

Reference

자동 커밋 사용은 기본적으로 일반적인 JDBC에서 지원하고 있는 기능입니다. 그러나 사용자 환경에 따라 사용에 제약이 있을 수 있습니다. 예를 들어 Oracle에서 dblink를 이용하여 DB2에 연결하여 사용할 경우 제약 사항이 있으므로 아래와 같이 설정하여 사용하면 됩니다. 자세한 사항은 DB 벤더별 JDBC 문서에서 아래의 항목을 참조하시기 바랍니다.

JDBC	OZ
<code>com.setAutoCommit(false)</code>	자동 커밋=false
<code>com.preparedStatement(sqlstr)</code>	컴파일된 질의문 사용=true

See Also

[AutoCommit](#)

벤더

데이터베이스 벤더 종류

See Also

[VendorName](#)

데이터 액션 클래스

데이터 액션 시 사용할 클래스 이름

데이터 액션 초기화 패러미터

데이터 액션 초기화 패러미터

사용자 데이터 스토어

사용자 데이터(UDS:사용자 데이터 스토어)는 데이터베이스가 아닌 EJB, 파일, 서블릿 등의 사용자 애플리케이션 데이터를 이용하여 보고서를 제작해야 하는 경우 사용하는 Non-DB 데이터 연동 방식입니다.

Name	Description
이름	사용자 데이터 스토어 이름
클래스 이름	UDS 클래스 파일 이름
필드 정의 사용	필드 정의 사용 여부

이름

사용자 데이터 스토어 이름

See Also

[GetName](#)

클래스 이름

UDS 클래스 파일 이름

See Also

[ClassName](#)

필드 정의 사용

오즈 리포트 디자이너에 필드와 타입을 삽입 가능하도록 할지 여부

Values

True	오즈 리포트 디자이너에 필드와 타입 삽입 가능
False	오즈 리포트 디자이너에 필드와 타입 삽입 불가

그룹 데이터 스토어

데이터베이스, 사용자 데이터 등에서 생성된 데이터 소스로부터 여러 개의 새로운 그룹 데이터 소스를 생성하여 관리합니다. 하나의 데이터 소스에서 다수의 그룹 데이터를 생성하여 마스터-디테일 관계를 설정할 수 있습니다.

Name	Description
이름	그룹 데이터 스토어 이름
원본셋	그룹 데이터 스토어의 원본 데이터 셋 이름

이름

그룹 데이터 스토어 이름

See Also

[GetName](#)

원본셋

그룹 데이터 스토어에서 사용할 데이터 메타 혹은 그룹 데이터 스토어의 원본 데이터 셋 이름

See Also

[SourceDataSetName](#)

SOAP 스토어

웹 서비스를 통하여 얻어온 데이터를 데이터 소스로 사용합니다.

Name	Description
이름	SOAP 스토어 이름

이름

SOAP 스토어 이름

See Also

[GetName](#)

ADO DataSet 웹서비스 스토어

ADO DataSet 웹서비스 스토어는 닷넷 웹 서비스에서 ADO DataSet 객체를 데이터 소스로 사용합니다.

Name	Description
이름	ADO DataSet 웹서비스 스토어 이름

이름

ADO 멀티 데이터 셋 이름

XML 스토어

DTD/XSD를 이용하는 XML 형식의 데이터를 데이터 소스로 사용합니다.

Name	Description
이름	XML 스토어 이름

이름

XML 스토어 이름

See Also

[GetName](#)

SAP 스토어

SAP R/3에 정의된 RFC(Remote Function Call) 실행 결과를 데이터 소스로 사용합니다.

Name	Description
클라이언트 이름	로그인할 클라이언트 번호
언어	로그인 시 사용할 언어
호스트 이름(주소)	SAP 서버의 호스트 이름
시스템 번호	SAP 서버의 시스템 번호
데이터 액션 클래스	데이터 액션 시 사용할 클래스 이름
데이터 액션 초기화 패러미터	데이터 액션 초기화 패러미터
SID	SAP Netweaver 서버에 설정된 커넥션 풀의 SID

클라이언트 이름

로그인할 클라이언트 번호

See Also

[ClientName](#)

언어

로그인 시 사용할 언어

See Also

[Language](#)

호스트 이름(주소)

SAP 서버의 호스트 이름

See Also

[HostName](#)

시스템 번호

SAP 서버의 시스템 번호

See Also

[SystemNumber](#)

데이터 액션 클래스

데이터 액션 시 사용할 클래스 이름

데이터 액션 초기화 패러미터

데이터 액션 초기화 패러미터

SID

SAP Netweaver 서버에 설정된 커넥션 풀의 SID

See Also

[SID](#)

CLEAR QUEST 스토어

형상 관리 툴인 IBM Rational Clear Quest 프로그램에서 생성한 DB를 통하여 얻어온 데이터를 데이터 소스로 사용합니다.

Name	Description
이름	CLEAR QUEST 스토어 이름

이름

CLEAR QUEST 스토어 이름

오즈 데이터 셋

- 데이터베이스 데이터 셋
- 다이나믹 데이터 셋
- 멀티 데이터 셋
- 프로시저 데이터 셋
- 사용자 데이터 셋
- 그룹 데이터 셋
- 파일 스토어 XML 데이터 셋
- 파일 스토어 CSV 데이터 셋
- HTTP 스토어 XML 데이터 셋
- HTTP 스토어 CSV 데이터 셋
- SOAP 데이터 셋
- AOD DataSet 웹 서비스 데이터 셋
- XML 데이터 셋
- SAP 데이터 셋
- CLEAR QUEST 데이터 셋

데이터베이스 데이터 셋

대상 데이터베이스에서 특정 질의문으로 가져온 데이터 셋 정보를 관리합니다. 하위에 다수의 필드 정보를 가집니다.

Name	Description
이름	쿼리 데이터 셋 이름
질의문	실행할 쿼리문
마스터셋	마스터셋 이름
최대행 수	최대 행 수
컴파일된 질의문 사용	컴파일된 질의문 사용 여부
JDBC 전송 ROW	결과 셋 반환 시 한꺼번에 가져올 수 있는 행 수

스크립트 사용	스크립트를 이용한 쿼리문 사용이 가능하도록 할지 여부
숨김	데이터 셋을 숨길지 여부
Concurrent 전송 크기	Concurrent 전송 크기
Concurrent 첫 전송 ROW	클라이언트에서 데이터 요청 시 우선적으로 먼저 전송할 데이터의 행수
바인딩 전 데이터 보내기	데이터 셋 바인딩 전에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부
바인딩 후 데이터 보내기	데이터 셋 바인딩 후에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부
쿼리 타임아웃	쿼리 실행 제한 시간
쿼리 에러 무시	쿼리 에러 발생 시 빈 데이터 셋 전송 여부
설명	데이터 셋 설명

이름

쿼리 데이터 셋 이름

See Also

[GetName](#)

질의문

실행할 쿼리문

See Also

[QueryString](#)

마스터셋

마스터셋 이름

Reference

해당 데이터 셋의 마스터 데이터 셋을 설정합니다.

See Also

[GetMasterDataSetName](#)

최대행 수

최대 행 수

Reference

바인딩 시 해당 데이터 셋의 최대 행(Row) 수를 설정합니다.
0 ~ 2147483647 사이의 값으로 최대 행 수를 설정합니다.

See Also

[MaxRowCount](#)

컴파일된 질의문 사용

컴파일된 질의문 사용 여부

Values

True	컴파일된 질의문 사용
False	컴파일된 질의문을 사용 하지 않음

Reference

"True"로 설정하면 서버에 이미 만들어진 질의문이 실행되므로, 같은 질의문이 여러 번 반복 실행될 경우 이 속성 값을 "True"로 설정하면 서버의 성능을 높여줍니다. 컴파일된 질의문은 각 데이터베이스에서 제공하는 컴파일된 질의문을 사용하여 쿼리를 수행합니다. (JDBC에서는 쿼리 실행 시 preparedstatement를 사용합니다.)

이 옵션은 서버에서 질의문을 실행할 때에만 체크되는 옵션이며, 쿼리 디자이너에서 질의문을 실행할 때에는 이 옵션과는 상관없이 컴파일된 질의문이 아닌 일반적인 질의문이 실행됩니다.

See Also

[UseCompiledSQL](#)

JDBC 전송 ROW

결과 셋 반환 시 한꺼번에 가져올 수 있는 행 수

See Also

[JDBCFetchRow](#)

스크립트 사용

스크립트를 이용한 쿼리문 사용이 가능하도록 할지 여부

Values

True	스크립트를 이용한 쿼리문 사용 가능
False	스크립트를 이용한 쿼리문을 사용할 수 없음

See Also

[UseOZScript](#)

숨김

데이터 셋을 숨길지 여부

Values

True	데이터 셋을 숨김
False	데이터 셋을 숨기지 않음

Reference

데이터 셋을 숨길 경우 해당 데이터 셋은 리포트 디자이너의 데이터 트리에 표시되지 않고, 자바 스크립트로도 접근할 수 없습니다.

또한, 오즈 서버가 데이터를 데이터 모듈 단위로 전송할 때 숨김 상태인 데이터 셋은 제외하고 전송합니다.

그룹 데이터 스토어를 사용할 경우 원본셋을 숨김으로 설정하면 원본셋은 서버에서 전송하지 않습니다.

See Also

[Hidden](#)

Concurrent 전송 크기

Concurrent 전송 크기

Reference

클라이언트에서 데이터 요청 시 한꺼번에 가져올 결과 데이터 셋의 크기를 Byte 단위로 설정합니다.

설정된 전송 크기만큼의 버퍼 단위로 데이터를 내보냅니다

속성 값을 0으로 설정하면 서버에 설정된 값이 적용됩니다.

이 속성 값은 쿼리 디자이너에서 쿼리문을 실행할 때에는 적용되지 않으며, 리포트 뷰어에서 데이터를 요청할 경우에만 적용되는 속성입니다. 이 속성을 이용하려면 뷰어의 패치 단위 설정 패러미터를 데이터 셋 단위로 변경하시기 바랍니다.

See Also

[ConcurrentFetchSize](#)

Concurrent 첫 전송 ROW

클라이언트에서 데이터 요청 시 우선적으로 먼저 전송할 데이터의 행수

Reference

대용량 데이터일 경우 첫 데이터를 빠르게 받아 응답 속도를 향상시키고자 할 때 유용하게 사용될 수 있습니다.

속성 값을 0으로 설정하면 서버에 설정된 값이 적용됩니다.

이 속성 값은 쿼리 디자이너에서 쿼리문을 실행할 때에는 적용되지 않으며, 리포트 뷰어에서 데이터를 요청할 경우에만 적용되는 속성입니다. 이 속성을 이용하려면 뷰어의 패치 단위 설정 패러미터를 데이터 셋 단위로 변경하시기 바랍니다.

See Also

[ConcurrentFirstRow](#)

바인딩 전 데이터 보내기

데이터 셋 바인딩 전에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부

Values

True	데이터 셋 바인딩 전에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비움
False	바인딩 전 데이터 보내기 기능을 사용하지 않음

Reference

이 기능은 패치 타입이 concurrent이고 패치 단위가 데이터 모듈인 경우에만 적용됩니다.

See Also

[FlushDataBeforeBinding](#)

바인딩 후 데이터 보내기

데이터 셋 바인딩 후에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부

Values

True	데이터 셋 바인딩 후에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비움
False	바인딩 후 데이터 보내기 기능을 사용하지 않음

Reference

이 기능은 패치 타입이 concurrent이고 패치 단위가 데이터 모듈인 경우에만 적용됩니다.

See Also

[FlushDataAfterBinding](#)

쿼리 타임아웃

쿼리 실행 제한 시간

Reference

0 ~ 2147483647 사이의 값으로 설정할 수 있습니다.

특정 쿼리의 실행 속도가 늦어져 세션이 홀딩되고 시스템의 과부하로 인해 다른 쿼리의 응답 속도도 느려지는 문제를 개선하려면 속성 값을 0보다 큰 값으로 적절히 설정하시기 바랍니다.

See Also

[QueryTimeout](#)

쿼리 에러 무시

쿼리 에러 발생 시 빈 데이터 셋 전송 여부

Values

True	쿼리 에러 발생 시 필드 정보만 있고 결과 셋은 없는 빈 데이터 셋을 전송
False	쿼리 에러 발생 시 데이터 셋 전송 안 함

See Also

[IgnoreQueryError](#)

설명

데이터 셋 설명

다이나믹 데이터 셋

데이터베이스 스토어를 통해 가져온 데이터를 다이나믹 쿼리를 사용하여 동적으로 필드 정보를 가져오는 2차 데이터 셋입니다.

Name	Description
이름	쿼리 데이터 셋 이름
마스터셋	마스터셋 이름
최대행 수	최대 행 수
시작 행 번호	시작 행 번호
숨김	데이터 셋을 숨길지 여부
Concurrent 전송 크기	Concurrent 전송 크기
Concurrent 첫 전송 ROW	클라이언트에서 데이터 요청 시 우선적으로 먼저 전송할 데이터의 행수
컴파일된 SQL문 사용 (원본 셋)	원본 셋에서 컴파일된 질의문 사용 여부
스크립트 사용 (원본 셋)	원본 셋에서 스크립트를 이용한 쿼리문 사용이 가능하도록 할지 여부
바인딩 전 데이터 보내기	데이터 셋 바인딩 전에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부
바인딩 후 데이터 보내기	데이터 셋 바인딩 후에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부

쿼리 타임아웃	쿼리 실행 제한 시간
쿼리 에러 무시	쿼리 에러 발생 시 빈 데이터 셋 전송 여부
설명	데이터 셋 설명

이름

쿼리 데이터 셋 이름

마스터셋

마스터셋 이름

Reference

해당 데이터 셋의 마스터 데이터 셋을 설정합니다.

최대행 수

최대 행 수

Reference

바인딩 시 해당 데이터 셋의 최대 행(Row) 수를 설정합니다.

0 ~ 2147483647 사이의 값으로 최대 행 수를 설정합니다.

시작 행 번호

시작 행 번호

Values

다이나믹 데이터 셋 생성 시 ROW INDEX에 처음 시작되는 번호입니다.

Reference

기본 값은 "0"이며 각 ROW INDEX는 0부터 차례로 번호가 부여됩니다.

사용자가 0 이상의 값을 설정할 수 있습니다.

0 ~ 2147483647 이외의 값은 최대 행 수로 설정할 수 없습니다.

숨김

데이터 셋을 숨길지 여부

Values

True	데이터 셋을 숨김
False	데이터 셋을 숨기지 않음

Reference

데이터 셋을 숨길 경우 해당 데이터 셋은 리포트 디자이너의 데이터 트리에 표시되지 않고, 자바 스크립트로도 접근할 수 없습니다.

또한, 오즈 서버가 데이터를 데이터 모듈 단위로 전송할 때 숨김 상태인 데이터 셋은 제외하고 전송합니다. 그룹 데이터 스토어를 사용할 경우 원본셋을 숨김으로 설정하면 원본셋은 서버에서 전송하지 않습니다.

Concurrent 전송 크기

Concurrent 전송 크기

Reference

클라이언트에서 데이터 요청 시 한꺼번에 가져올 결과 데이터 셋의 크기를 Byte 단위로 설정합니다.

설정된 전송 크기만큼의 버퍼 단위로 데이터를 내보냅니다

속성 값을 0으로 설정하면 서버에 설정된 값이 적용됩니다.

이 속성 값은 쿼리 디자이너에서 쿼리문을 실행할 때에는 적용되지 않으며, 리포트 뷰어에서 데이터를 요청할 경우에만 적용되는 속성입니다. 이 속성을 이용하려면 뷰어의 패치 단위 설정 패러미터를 데이터 셋 단위로 변경하시기 바랍니다.

Concurrent 첫 전송 ROW

클라이언트에서 데이터 요청 시 우선적으로 먼저 전송할 데이터의 행수

Reference

대용량 데이터일 경우 첫 데이터를 빠르게 받아 응답 속도를 향상시키고자 할 때 유용하게 사용될 수 있습니다.

속성 값을 0으로 설정하면 서버에 설정된 값이 적용됩니다.

이 속성 값은 쿼리 디자이너에서 쿼리문을 실행할 때에는 적용되지 않으며, 리포트 뷰어에서 데이터를 요청할 경우에만 적용되는 속성입니다. 이 속성을 이용하려면 뷰어의 패치 단위 설정 패러미터를 데이터 셋 단위로 변경하시기 바랍니다.

컴파일된 SQL문 사용 (원본 셋)

원본 셋에서 컴파일된 질의문 사용 여부

Values

True	컴파일된 질의문 사용
False	컴파일된 질의문을 사용 하지 않음

Reference

"True"로 설정하면 서버에 이미 만들어진 질의문이 실행되므로, 같은 질의문이 여러 번 반복 실행될 경우 이 속성 값을 "True"로 설정하면 서버의 성능을 높여줍니다. 컴파일된 질의문은 각 데이터베이스에서 제공하는 컴파일된 질의문을 사용하여 쿼리를 수행합니다. (JDBC에서는 쿼리 실행 시 preparedstatement를 사용합니다.)

이 옵션은 서버에서 질의문을 실행할 때에만 체크되는 옵션이며, 쿼리 디자이너에서 질의문을 실행할 때에는 이 옵션과는 상관없이 컴파일된 질의문이 아닌 일반적인 질의문이 실행됩니다.

스크립트 사용 (원본 셋)

원본 셋에서 스크립트를 이용한 쿼리문 사용이 가능하도록 할지 여부

Values

True	스크립트를 이용한 쿼리문 사용 가능
False	스크립트를 이용한 쿼리문을 사용할 수 없음

바인딩 전 데이터 보내기

데이터 셋 바인딩 전에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부

Values

True	데이터 셋 바인딩 전에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비움
False	바인딩 전 데이터 보내기 기능을 사용하지 않음

Reference

이 기능은 패치 타입이 concurrent이고 패치 단위가 데이터 모듈인 경우에만 적용됩니다.

바인딩 후 데이터 보내기

데이터 셋 바인딩 후에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부

Values

True	데이터 셋 바인딩 후에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비움
False	바인딩 후 데이터 보내기 기능을 사용하지 않음

Reference

이 기능은 패치 타입이 concurrent이고 패치 단위가 데이터 모듈인 경우에만 적용됩니다.

쿼리 타임아웃

쿼리 실행 제한 시간

Reference

0 ~ 2147483647 사이의 값으로 설정할 수 있습니다.

특정 쿼리의 실행 속도가 늦어져 세션이 홀딩되고 시스템의 과부하로 인해 다른 쿼리의 응답 속도도 느려지는 문제를 개선하려면 속성 값을 0보다 큰 값으로 적절히 설정하시기 바랍니다.

See Also

[QueryTimeout](#)

쿼리 에러 무시

쿼리 에러 발생 시 빈 데이터 셋 전송 여부

Values

True	쿼리 에러 발생 시 필드 정보만 있고 결과 셋은 없는 빈 데이터 셋을 전송
False	쿼리 에러 발생 시 데이터 셋 전송 안 함

See Also

[IgnoreQueryError](#)

설명

데이터 셋 설명

멀티 데이터 셋

데이터 셋에서 프로시저나 여러 개의 쿼리문을 입력하여 생성된 복수 개의 결과 셋을 관리합니다.

Name	Description
이름	쿼리 데이터 셋 이름
질의문	실행할 쿼리문
마스터셋	마스터셋 이름
컴파일된 질의문 사용	원본 셋에서 컴파일된 질의문 사용 여부
스크립트 사용	스크립트를 이용한 쿼리문 사용이 가능하도록 할지 여부
쿼리 타임아웃	쿼리 실행 제한 시간

이름

쿼리 데이터 셋 이름

See Also

[GetName](#)

질의문

실행할 쿼리문

See Also

[QueryString](#)

마스터셋

마스터셋 이름

Reference

해당 데이터 셋의 마스터 데이터 셋을 설정합니다.

See Also

[GetMasterDataSetName](#)

컴파일된 질의문 사용

컴파일된 질의문 사용 여부

Values

True	컴파일된 질의문 사용
False	컴파일된 질의문을 사용 하지 않음

Reference

"True"로 설정하면 서버에 이미 만들어진 질의문이 실행되므로, 같은 질의문이 여러 번 반복 실행될 경우 이 속성 값을 "True"로 설정하면 서버의 성능을 높여줍니다. 컴파일된 질의문은 각 데이터베이스에서 제공하는 컴파일된 질의문을 사용하여 쿼리를 수행합니다. (JDBC에서는 쿼리 실행 시 preparedstatement를 사용합니다.)

이 옵션은 서버에서 질의문을 실행할 때에만 체크되는 옵션이며, 쿼리 디자이너에서 질의문을 실행할 때에는 이 옵션과는 상관없이 컴파일된 질의문이 아닌 일반적인 질의문이 실행됩니다.

See Also

[UseCompiledSQL](#)

스크립트 사용

스크립트를 이용한 쿼리문 사용이 가능하도록 할지 여부

Values

True	스크립트를 이용한 쿼리문 사용 가능
False	스크립트를 이용한 쿼리문을 사용할 수 없음

See Also

[UseOZScript](#)

쿼리 타임아웃

쿼리 실행 제한 시간

Reference

0 ~ 2147483647 사이의 값으로 설정할 수 있습니다.

특정 쿼리의 실행 속도가 늦어져 세션이 홀딩되고 시스템의 과부하로 인해 다른 쿼리의 응답 속도도 느려지는 문제를 개선하려면 속성 값을 0보다 큰 값으로 적절히 설정하시기 바랍니다.

See Also

[QueryTimeout](#)

프로시저 데이터 셋

데이터 셋에서 프로시저를 설정하여 생성된 복수 개의 결과 셋을 관리합니다.

Name	Description
이름	쿼리 데이터 셋 이름
프로시저 이름	프로시저 이름
프로시저 타입	프로시저 타입
타겟 패러미터	타겟 패러미터 이름
마스터셋	마스터셋 이름
쿼리 타임아웃	쿼리 실행 제한 시간

이름

쿼리 데이터 셋 이름

See Also

[GetName](#)

프로시저 이름

프로시저 이름

Values

프로시저 이름을 설정합니다.

See Also

[ProcedureName](#)

프로시저 타입

프로시저 타입

Values

프로시저 타입을 설정합니다.

Reference

프로시저 또는 함수로 설정 가능합니다.

See Also

[ProcedureType](#)

타겟 패러미터

타겟 패러미터 이름

Values

실행된 프로시저의 결과 값을 가져올 패러미터 이름을 설정합니다.

Reference

패러미터의 데이터 타입이 Cursor일 경우에만 표시되며, 그 외 타입의 필드를 참조할 경우에는 빈 값이 표시됩니다.

See Also

[TargetParameter](#)

마스터셋

마스터셋 이름

Reference

해당 데이터 셋의 마스터 데이터 셋을 설정합니다.

See Also

[GetMasterDataSetName](#)

쿼리 타임아웃

쿼리 실행 제한 시간

Reference

0 ~ 2147483647 사이의 값으로 설정할 수 있습니다.

특정 쿼리의 실행 속도가 늦어져 세션이 홀딩되고 시스템의 과부하로 인해 다른 쿼리의 응답 속도도 느려지는 문제를 개선하려면 속성 값을 0보다 큰 값으로 적절히 설정하시기 바랍니다.

See Also

[QueryTimeout](#)

사용자 데이터 셋

사용자 데이터 스토어에서 특정 질의문으로 가져온 데이터 셋 정보를 관리합니다. 하위에 다수의 필드 정보를 가집니다.

Name	Description
이름	사용자 데이터 셋 이름
실행문	실행문
마스터셋	마스터셋 이름
최대행 수	최대 행 수
스크립트 사용	스크립트를 이용한 쿼리문 사용이 가능하도록 할지 여부

숨김	데이터 셋을 숨길지 여부
Concurrent 전송 크기	Concurrent 전송 크기
Concurrent 첫 전송 ROW	클라이언트에서 데이터 요청 시 우선적으로 먼저 전송할 데이터의 행수
바인딩 전 데이터 보내기	데이터 셋 바인딩 전에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부
바인딩 후 데이터 보내기	데이터 셋 바인딩 후에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부
설명	데이터 셋 설명

이름

사용자 데이터 셋 이름

See Also

[GetName](#)

실행문

실행문

Reference

UDS 클래스를 통해 데이터를 가져오기 위한 실행문을 설정합니다.
설정한 실행문은 UDS 클래스에 아규먼트로 전달됩니다.

See Also

[Command](#)

마스터셋

마스터셋 이름

Reference

해당 데이터 셋의 마스터 데이터 셋을 설정합니다.

See Also

[GetMasterDataSetName](#)

최대행 수

최대 행 수

Reference

바인딩 시 해당 데이터 셋의 최대 행(Row) 수를 설정합니다.
0 ~ 2147483647 사이의 값으로 최대 행 수를 설정합니다.

스크립트 사용

스크립트를 이용한 쿼리문 사용이 가능하도록 할지 여부

Values

True	스크립트를 이용한 쿼리문 사용 가능
False	스크립트를 이용한 쿼리문을 사용할 수 없음

See Also

[UseOZScript](#)

숨김

데이터 셋을 숨길지 여부

Values

True	데이터 셋을 숨김
False	데이터 셋을 숨기지 않음

Reference

데이터 셋을 숨길 경우 해당 데이터 셋은 리포트 디자이너의 데이터 트리에 표시되지 않고, 자바 스크립트로도 접근할 수 없습니다.

또한, 오즈 서버가 데이터를 데이터 모듈 단위로 전송할 때 숨김 상태인 데이터 셋은 제외하고 전송합니다.

그룹 데이터 스토어를 사용할 경우 원본셋을 숨김으로 설정하면 원본셋은 서버에서 전송하지 않습니다.

See Also

[Hidden](#)

Concurrent 전송 크기

Concurrent 전송 크기

Reference

클라이언트에서 데이터 요청 시 한꺼번에 가져올 결과 데이터 셋의 크기를 Byte 단위로 설정합니다.

설정된 전송 크기만큼의 버퍼 단위로 데이터를 내보냅니다

속성 값을 0으로 설정하면 서버에 설정된 값이 적용됩니다.

이 속성 값은 쿼리 디자이너에서 쿼리문을 실행할 때에는 적용되지 않으며, 리포트 뷰어에서 데이터를 요청할 경우에만 적용되는 속성입니다. 이 속성을 이용하려면 뷰어의 패치 단위 설정 패러미터를 데이터 셋 단위로 변경하시기 바랍니다.

See Also

[ConcurrentFetchSize](#)

Concurrent 첫 전송 ROW

클라이언트에서 데이터 요청 시 우선적으로 먼저 전송할 데이터의 행수

Reference

대용량 데이터일 경우 첫 데이터를 빠르게 받아 응답 속도를 향상시키고자 할 때 유용하게 사용될 수 있습니다.

속성 값을 0으로 설정하면 서버에 설정된 값이 적용됩니다.

이 속성 값은 쿼리 디자이너에서 쿼리문을 실행할 때에는 적용되지 않으며, 리포트 뷰어에서 데이터를 요청할 경우에만 적용되는 속성입니다. 이 속성을 이용하려면 뷰어의 패치 단위 설정 패러미터를 데이터 셋 단위로 변경하시기 바랍니다.

See Also

[ConcurrentFirstRow](#)

바인딩 전 데이터 보내기

데이터 셋 바인딩 전에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부

Values

True	데이터 셋 바인딩 전에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비움
False	바인딩 전 데이터 보내기 기능을 사용하지 않음

Reference

이 기능은 패치 타입이 concurrent이고 패치 단위가 데이터 모듈인 경우에만 적용됩니다.

See Also

[FlushDataBeforeBinding](#)

바인딩 후 데이터 보내기

데이터 셋 바인딩 후에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부

Values

True	데이터 셋 바인딩 후에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비움
False	바인딩 후 데이터 보내기 기능을 사용하지 않음

Reference

이 기능은 패치 타입이 concurrent이고 패치 단위가 데이터 모듈인 경우에만 적용됩니다.

See Also

[FlushDataAfterBinding](#)

설명

데이터 셋 설명

Values

데이터 셋의 설명을 설정합니다.

그룹 데이터 셋

그룹 데이터 스토어에서 특정 질의문으로 가져온 데이터 셋 정보를 관리합니다. 하위에 다수의 필드 정보를 가집니다.

Name	Description
이름	그룹 데이터 셋 이름
실행문	실행문
마스터셋	마스터셋 이름
Consume Row	원본셋의 데이터를 여러 개의 데이터 셋으로 분할할지 여부
최대행 수	최대 행 수
숨김	데이터 셋을 숨길지 여부
Concurrent 전송 크기	Concurrent 전송 크기
Concurrent 첫 전송 ROW	클라이언트에서 데이터 요청 시 우선적으로 먼저 전송할 데이터의 행수
바인딩 전 데이터 보내기	데이터 셋 바인딩 전에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부
바인딩 후 데이터 보내기	데이터 셋 바인딩 후에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부
설명	데이터 셋 설명

이름

그룹 데이터 셋 이름

See Also

[GetName](#)

실행문

실행문

Reference

그룹 데이터를 생성하기 위한 실행문을 설정합니다.

See Also

[Command](#)

마스터셋

마스터셋 이름

Reference

해당 데이터 셋의 마스터 데이터 셋을 설정합니다.

See Also

[GetMasterDataSetName](#)

Consume Row

원본셋의 데이터를 여러 개의 데이터 셋으로 분할할지 여부

Values

True	원본셋의 데이터를 여러 개의 데이터 셋으로 분할
False	원본셋의 데이터를 여러 개의 데이터 셋으로 분할하지 않음

Reference

한 행의 데이터를 모두 분할하고 다음 행을 가져오는 시점에 "True"를 설정하여야 그룹 데이터 스토어의 바인딩이 정상적으로 작동하게 됩니다. 일반적으로 하나의 그룹 데이터 스토어 내의 데이터 셋들 중 가장 하위의 데이터 셋만 "True"로 설정합니다.

최대행 수

최대 행 수

Reference

바인딩 시 해당 데이터 셋의 최대 행(Row) 수를 설정합니다.

0 ~ 2147483647 사이의 값으로 최대 행 수를 설정합니다.

See Also

[MaxRowCount](#)

숨김

데이터 셋을 숨길지 여부

Values

True	데이터 셋을 숨김
False	데이터 셋을 숨기지 않음

Reference

데이터 셋을 숨길 경우 해당 데이터 셋은 리포트 디자이너의 데이터 트리에 표시되지 않고, 자바 스크립트로도 접근할 수 없습니다.

또한, 오즈 서버가 데이터를 데이터 모듈 단위로 전송할 때 숨김 상태인 데이터 셋은 제외하고 전송합니다.

그룹 데이터 스토어를 사용할 경우 원본셋을 숨김으로 설정하면 원본셋은 서버에서 전송하지 않습니다.

See Also

[Hidden](#)

Concurrent 전송 크기

Concurrent 전송 크기

Reference

클라이언트에서 데이터 요청 시 한꺼번에 가져올 결과 데이터 셋의 크기를 Byte 단위로 설정합니다.

설정된 전송 크기만큼의 버퍼 단위로 데이터를 내보냅니다

속성 값을 0으로 설정하면 서버에 설정된 값이 적용됩니다.

이 속성 값은 쿼리 디자이너에서 쿼리문을 실행할 때에는 적용되지 않으며, 리포트 뷰어에서 데이터를 요청할 경우에만 적용되는 속성입니다. 이 속성을 이용하려면 뷰어의 패치 단위 설정 패러미터를 데이터 셋 단위로 변경하시기 바랍니다.

See Also

[ConcurrentFetchSize](#)

Concurrent 첫 전송 ROW

클라이언트에서 데이터 요청 시 우선적으로 먼저 전송할 데이터의 행수

Reference

대용량 데이터일 경우 첫 데이터를 빠르게 받아 응답 속도를 향상시키고자 할 때 유용하게 사용될 수 있습니다.

속성 값을 0으로 설정하면 서버에 설정된 값이 적용됩니다.

이 속성 값은 쿼리 디자이너에서 쿼리문을 실행할 때에는 적용되지 않으며, 리포트 뷰어에서 데이터를 요청할 경우에만 적용되는 속성입니다. 이 속성을 이용하려면 뷰어의 패치 단위 설정 패러미터를 데이터 셋 단위로 변경하시기 바랍니다.

See Also

[ConcurrentFirstRow](#)

바인딩 전 데이터 보내기

데이터 셋 바인딩 전에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부

Values

True	데이터 셋 바인딩 전에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비움
False	바인딩 전 데이터 보내기 기능을 사용하지 않음

Reference

이 기능은 패치 타입이 concurrent이고 패치 단위가 데이터 모듈인 경우에만 적용됩니다.

See Also

[FlushDataBeforeBinding](#)

바인딩 후 데이터 보내기

데이터 셋 바인딩 후에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부

Values

True	데이터 셋 바인딩 후에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비움
False	바인딩 후 데이터 보내기 기능을 사용하지 않음

Reference

이 기능은 패치 타입이 concurrent이고 패치 단위가 데이터 모듈인 경우에만 적용됩니다.

See Also

[FlushDataAfterBinding](#)

설명

데이터 셋 설명

Values

데이터 셋의 설명을 설정합니다.

파일 스토어 XML 데이터 셋

사용자 애플리케이션에서 생성한 XML 파일이나 스트림 형식의 텍스트 데이터를 데이터 소스로 사용합니다.

Name	Description
이름	파일 스토어 XML 데이터 셋 이름
마스터셋	마스터셋 이름
최대행 수	최대 행 수
데이터 셋 노드 이름	XML 파일의 데이터 셋 노드 이름
레코드셋 노드 이름	XML 파일의 레코드셋 노드 이름
숨김	데이터 셋을 숨길지 여부
Concurrent 전송 크기	Concurrent 전송 크기
Concurrent 첫 전송 ROW	클라이언트에서 데이터 요청 시 우선적으로 먼저 전송할 데이터의 행수

바인딩 전 데이터 보내기	데이터 셋 바인딩 전에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부
바인딩 후 데이터 보내기	데이터 셋 바인딩 후에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부
설명	데이터 셋 설명

이름

파일 스토어의 XML 데이터 셋 이름

See Also

[GetName](#)

마스터셋

마스터셋 이름

Reference

해당 데이터 셋의 마스터 데이터 셋을 설정합니다.

See Also

[GetMasterDataSetName](#)

최대행 수

최대 행 수

Reference

바인딩 시 해당 데이터 셋의 최대 행(Row) 수를 설정합니다.

0 ~ 2147483647 사이의 값으로 최대 행 수를 설정합니다.

See Also

[MaxRowCount](#)

데이터 셋 노드 이름

XML 파일의 데이터 셋 노드 이름

See Also

[DataSetNodeName](#)

레코드셋 노드 이름

XML 파일의 레코드셋 노드 이름

See Also

[RecordNodeName](#)

숨김

데이터 셋을 숨길지 여부

Values

True	데이터 셋을 숨김
False	데이터 셋을 숨기지 않음

Reference

데이터 셋을 숨길 경우 해당 데이터 셋은 리포트 디자이너의 데이터 트리에 표시되지 않고, 자바 스크립트로도 접근할 수 없습니다.

또한, 오즈 서버가 데이터를 데이터 모듈 단위로 전송할 때 숨김 상태인 데이터 셋은 제외하고 전송합니다.

그룹 데이터 스토어를 사용할 경우 원본셋을 숨김으로 설정하면 원본셋은 서버에서 전송하지 않습니다.

See Also

[Hidden](#)

Concurrent 전송 크기

Concurrent 전송 크기

Reference

클라이언트에서 데이터 요청 시 한꺼번에 가져올 결과 데이터 셋의 크기를 Byte 단위로 설정합니다.

설정된 전송 크기만큼의 버퍼 단위로 데이터를 내보냅니다

속성 값을 0으로 설정하면 서버에 설정된 값이 적용됩니다.

이 속성 값은 쿼리 디자이너에서 쿼리문을 실행할 때에는 적용되지 않으며, 리포트 뷰어에서 데이터를 요청할 경우에만 적용되는 속성입니다. 이 속성을 이용하려면 뷰어의 패치 단위 설정 패러미터를 데이터 셋 단위로 변경하시기 바랍니다.

See Also

[ConcurrentFetchSize](#)

Concurrent 첫 전송 ROW

클라이언트에서 데이터 요청 시 우선적으로 먼저 전송할 데이터의 행수

Reference

대용량 데이터일 경우 첫 데이터를 빠르게 받아 응답 속도를 향상시키고자 할 때 유용하게 사용될 수 있습니다.

속성 값을 0으로 설정하면 서버에 설정된 값이 적용됩니다.

이 속성 값은 쿼리 디자이너에서 쿼리문을 실행할 때에는 적용되지 않으며, 리포트 뷰어에서 데이터를 요청할 경우에만 적용되는 속성입니다. 이 속성을 이용하려면 뷰어의 패치 단위 설정 패러미터를 데이터 셋 단위로 변경하시기 바랍니다.

See Also

[ConcurrentFirstRow](#)

바인딩 전 데이터 보내기

데이터 셋 바인딩 전에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부

Values

True	데이터 셋 바인딩 전에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비움
False	바인딩 전 데이터 보내기 기능을 사용하지 않음

Reference

이 기능은 패치 타입이 concurrent이고 패치 단위가 데이터 모듈인 경우에만 적용됩니다.

See Also

[FlushDataBeforeBinding](#)

바인딩 후 데이터 보내기

데이터 셋 바인딩 후에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부

Values

True	데이터 셋 바인딩 후에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비움
False	바인딩 후 데이터 보내기 기능을 사용하지 않음

Reference

이 기능은 패치 타입이 concurrent이고 패치 단위가 데이터 모듈인 경우에만 적용됩니다.

See Also

[FlushDataAfterBinding](#)

설명

데이터 셋 설명

Values

데이터 셋의 설명을 설정합니다.

파일 스토어 CSV 데이터 셋

사용자 애플리케이션에서 생성한 CSV 파일이나 스트림 형식의 텍스트 데이터를 데이터 소스로 사용합니다.

Name	Description
이름	파일 스토어 CSV 데이터 셋 이름
마스터셋	마스터셋 이름
최대행 수	최대 행 수
필드 구분자	필드 구분자

라인 구분자	라인 구분자
문자 셋	CSV 파일의 문자셋
널 마크	데이터가 널인 경우 표시할 값
숨김	데이터 셋을 숨길지 여부
Concurrent 전송 크기	Concurrent 전송 크기
Concurrent 첫 전송 ROW	클라이언트에서 데이터 요청 시 우선적으로 먼저 전송할 데이터의 행수
바인딩 전 데이터 보내기	데이터 셋 바인딩 전에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부
바인딩 후 데이터 보내기	데이터 셋 바인딩 후에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부
설명	데이터 셋 설명

이름

파일 스토어의 CSV 데이터 셋 이름

See Also

[GetName](#)

마스터셋

마스터셋 이름

Reference

해당 데이터 셋의 마스터 데이터 셋을 설정합니다.

See Also

[GetMasterDataSetName](#)

최대행 수

최대 행 수

Reference

바인딩 시 해당 데이터 셋의 최대 행(Row) 수를 설정합니다.
0 ~ 2147483647 사이의 값으로 최대 행 수를 설정합니다.

See Also

[MaxRowCount](#)

필드 구분자

필드 구분자

Reference

필드의 구분자를 지정하는 문자를 나타냅니다.

See Also

[ColumnDelimiter](#)

라인 구분자

라인 구분자

Reference

라인의 구분자를 지정하는 문자를 나타냅니다.

Reference

기본 값은 "LINEFEED"입니다.

라인 구분자는 영문 대소문자를 구분합니다.

속성 값이 "LINEFEED"일 경우 라인 구분자는 개행 문자(wn)로 설정됩니다.

기존 버전의 쿼리 디자이너에서 저장된 ODI의 CSV 데이터 셋 라인 구분자는 기본 값인 "LINEFEED"로 설정됩니다.

문자 셋

CSV 파일의 문자 셋

See Also

[CharacterSet](#)

널 마크

데이터가 널인 경우 표시할 값

See Also

[NullMark](#)

숨김

데이터 셋을 숨길지 여부

Values

True	데이터 셋을 숨김
False	데이터 셋을 숨기지 않음

Reference

데이터 셋을 숨길 경우 해당 데이터 셋은 리포트 디자이너의 데이터 트리에 표시되지 않고, 자바 스크립트로도 접근할 수 없습니다.

또한, 오즈 서버가 데이터를 데이터 모듈 단위로 전송할 때 숨김 상태인 데이터 셋은 제외하고 전송합니다. 그룹 데이터 스토어를 사용할 경우 원본셋을 숨김으로 설정하면 원본셋은 서버에서 전송하지 않습니다.

See Also

[Hidden](#)

Concurrent 전송 크기

Concurrent 전송 크기

Reference

클라이언트에서 데이터 요청 시 한꺼번에 가져올 결과 데이터 셋의 크기를 Byte 단위로 설정합니다.

설정된 전송 크기만큼의 버퍼 단위로 데이터를 내보냅니다

속성 값을 0으로 설정하면 서버에 설정된 값이 적용됩니다.

이 속성 값은 쿼리 디자이너에서 쿼리문을 실행할 때에는 적용되지 않으며, 리포트 뷰어에서 데이터를 요청할 경우에만 적용되는 속성입니다. 이 속성을 이용하려면 뷰어의 패치 단위 설정 패러미터를 데이터 셋 단위로 변경하시기 바랍니다.

See Also

[ConcurrentFetchSize](#)

Concurrent 첫 전송 ROW

클라이언트에서 데이터 요청 시 우선적으로 먼저 전송할 데이터의 행수

Reference

대용량 데이터일 경우 첫 데이터를 빠르게 받아 응답 속도를 향상시키고자 할 때 유용하게 사용될 수 있습니다.

속성 값을 0으로 설정하면 서버에 설정된 값이 적용됩니다.

이 속성 값은 쿼리 디자이너에서 쿼리문을 실행할 때에는 적용되지 않으며, 리포트 뷰어에서 데이터를 요청할 경우에만 적용되는 속성입니다. 이 속성을 이용하려면 뷰어의 패치 단위 설정 패러미터를 데이터 셋 단위로 변경하시기 바랍니다.

See Also

[ConcurrentFirstRow](#)

바인딩 전 데이터 보내기

데이터 셋 바인딩 전에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부

Values

True	데이터 셋 바인딩 전에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비움
False	바인딩 전 데이터 보내기 기능을 사용하지 않음

Reference

이 기능은 패치 타입이 concurrent이고 패치 단위가 데이터 모듈인 경우에만 적용됩니다.

See Also

[FlushDataBeforeBinding](#)

바인딩 후 데이터 보내기

데이터 셋 바인딩 후에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부

Values

True	데이터 셋 바인딩 후에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비움
False	바인딩 후 데이터 보내기 기능을 사용하지 않음

Reference

이 기능은 패치 타입이 concurrent이고 패치 단위가 데이터 모듈인 경우에만 적용됩니다.

See Also

[FlushDataAfterBinding](#)

설명

데이터 셋 설명

Values

데이터 셋의 설명을 설정합니다.

HTTP 스토어 XML 데이터 셋

사용자 애플리케이션에서 생성한 XML 파일이나 스트림 형식의 텍스트 데이터를 데이터 소스로 사용합니다.

Name	Description
이름	HTTP 스토어 XML 데이터 셋 이름
마스터셋	마스터셋 이름
최대행 수	최대 행 수
데이터 셋 노드 이름	XML 파일의 데이터 셋 노드 이름
레코드셋 노드 이름	XML 파일의 레코드셋 노드 이름

숨김	데이터 셋을 숨길지 여부
Concurrent 전송 크기	Concurrent 전송 크기
Concurrent 첫 전송 Row	클라이언트에서 데이터 요청 시 우선적으로 먼저 전송할 데이터의 행수
바인딩 전 데이터 보내기	데이터 셋 바인딩 전에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부
바인딩 후 데이터 보내기	데이터 셋 바인딩 후에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부
설명	데이터 셋 설명

이름

HTTP 스토어의 XML 데이터 셋 이름

마스터셋

마스터셋 이름

Reference

해당 데이터 셋의 마스터 데이터 셋을 설정합니다.

최대행 수

최대 행 수

Reference

바인딩 시 해당 데이터 셋의 최대 행(Row) 수를 설정합니다.

0 ~ 2147483647 사이의 값으로 최대 행 수를 설정합니다.

See Also

[MaxRowCount](#)

데이터 셋 노드 이름

XML 파일의 데이터 셋 노드 이름

See Also

[DataSetNodeName](#)

레코드셋 노드 이름

XML 파일의 레코드셋 노드 이름

See Also

[RecordNodeName](#)

숨김

데이터 셋을 숨길지 여부

Values

True	데이터 셋을 숨김
False	데이터 셋을 숨기지 않음

Reference

데이터 셋을 숨길 경우 해당 데이터 셋은 리포트 디자이너의 데이터 트리에 표시되지 않고, 자바 스크립트로 접근할 수 없습니다.

또한, 오즈 서버가 데이터를 데이터 모듈 단위로 전송할 때 숨김 상태인 데이터 셋은 제외하고 전송합니다. 그룹 데이터 스토어를 사용할 경우 원본셋을 숨김으로 설정하면 원본셋은 서버에서 전송하지 않습니다.

See Also

[Hidden](#)

Concurrent 전송 크기

Concurrent 전송 크기

Reference

클라이언트에서 데이터 요청 시 한꺼번에 가져올 결과 데이터 셋의 크기를 Byte 단위로 설정합니다.

설정된 전송 크기만큼의 버퍼 단위로 데이터를 내보냅니다

속성 값을 0으로 설정하면 서버에 설정된 값이 적용됩니다.

이 속성 값은 쿼리 디자이너에서 쿼리문을 실행할 때에는 적용되지 않으며, 리포트 뷰어에서 데이터를 요청할 경우에만 적용되는 속성입니다. 이 속성을 이용하려면 뷰어의 패치 단위 설정 패러미터를 데이터 셋 단위로 변경하시기 바랍니다.

See Also

[ConcurrentFetchSize](#)

Concurrent 첫 전송 ROW

클라이언트에서 데이터 요청 시 우선적으로 먼저 전송할 데이터의 행수

Reference

대용량 데이터일 경우 첫 데이터를 빠르게 받아 응답 속도를 향상시키고자 할 때 유용하게 사용될 수 있습니다.

속성 값을 0으로 설정하면 서버에 설정된 값이 적용됩니다.

이 속성 값은 쿼리 디자이너에서 쿼리문을 실행할 때에는 적용되지 않으며, 리포트 뷰어에서 데이터를 요청할 경우에만 적용되는 속성입니다. 이 속성을 이용하려면 뷰어의 패치 단위 설정 패러미터를 데이터 셋 단위로 변경하시기 바랍니다.

See Also

[ConcurrentFirstRow](#)

바인딩 전 데이터 보내기

데이터 셋 바인딩 전에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부

Values

True	데이터 셋 바인딩 전에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비움
False	바인딩 전 데이터 보내기 기능을 사용하지 않음

Reference

이 기능은 패치 타입이 concurrent이고 패치 단위가 데이터 모듈인 경우에만 적용됩니다.

See Also

[FlushDataBeforeBinding](#)

바인딩 후 데이터 보내기

데이터 셋 바인딩 후에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부

Values

True	데이터 셋 바인딩 후에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비움
False	바인딩 후 데이터 보내기 기능을 사용하지 않음

Reference

이 기능은 패치 타입이 concurrent이고 패치 단위가 데이터 모듈인 경우에만 적용됩니다.

See Also

[FlushDataAfterBinding](#)

설명

데이터 셋 설명

Values

데이터 셋의 설명을 설정합니다.

HTTP 스토어 CSV 데이터 셋

사용자 애플리케이션에서 생성한 CSV 파일이나 스트림 형식의 텍스트 데이터를 데이터 소스로 사용합니다.

Name	Description
이름	HTTP 스토어 CSV 데이터 셋 이름
마스터셋	마스터셋 이름
최대행 수	최대 행 수
필드 구분자	필드 구분자
라인 구분자	라인 구분자

문자 셋	CSV 파일의 문자셋
널 마크	데이터가 널인 경우 표시할 값
숨김	데이터 셋을 숨길지 여부
Concurrent 전송 크기	Concurrent 전송 크기
Concurrent 첫 전송 ROW	클라이언트에서 데이터 요청 시 우선적으로 먼저 전송할 데이터의 행수
바인딩 전 데이터 보내기	데이터 셋 바인딩 전에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부
바인딩 후 데이터 보내기	데이터 셋 바인딩 후에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부
설명	데이터 셋 설명

이름

HTTP 스토어의 CSV 데이터 셋 이름

마스터셋

마스터셋 이름

Reference

해당 데이터 셋의 마스터 데이터 셋을 설정합니다.

최대행 수

최대 행 수

Reference

바인딩 시 해당 데이터 셋의 최대 행(Row) 수를 설정합니다.

0 ~ 2147483647 사이의 값으로 최대 행 수를 설정합니다.

See Also

[MaxRowCount](#)

필드 구분자

필드 구분자

Reference

필드의 구분자를 지정하는 문자를 나타냅니다.

See Also

[ColumnDelimiter](#)

라인 구분자

라인 구분자

Reference

라인의 구분자를 지정하는 문자를 나타냅니다.

기본 값은 "LINEFEED"입니다.

라인 구분자는 영문 대소문자를 구분합니다.

속성 값이 "LINEFEED"일 경우 라인 구분자는 개행 문자(wn)로 설정됩니다.

기존 버전의 쿼리 디자이너에서 저장된 ODI의 CSV 데이터 셋 라인 구분자는 기본 값인 "LINEFEED"로 설정됩니다.

문자 셋

CSV 파일의 문자 셋

See Also

[CharacterSet](#)

널 마크

데이터가 널인 경우 표시할 값

See Also

[NullMark](#)

숨김

데이터 셋을 숨길지 여부

Values

True	데이터 셋을 숨김
False	데이터 셋을 숨기지 않음

Reference

데이터 셋을 숨길 경우 해당 데이터 셋은 리포트 디자이너의 데이터 트리에 표시되지 않고, 자바 스크립트로도 접근할 수 없습니다.

또한, 오즈 서버가 데이터를 데이터 모듈 단위로 전송할 때 숨김 상태인 데이터 셋은 제외하고 전송합니다. 그룹 데이터 스토어를 사용할 경우 원본셋을 숨김으로 설정하면 원본셋은 서버에서 전송하지 않습니다.

See Also

[Hidden](#)

Concurrent 전송 크기

Concurrent 전송 크기

Reference

클라이언트에서 데이터 요청 시 한꺼번에 가져올 결과 데이터 셋의 크기를 Byte 단위로 설정합니다.

설정된 전송 크기만큼의 버퍼 단위로 데이터를 내보냅니다

속성 값을 0으로 설정하면 서버에 설정된 값이 적용됩니다.

이 속성 값은 쿼리 디자이너에서 쿼리문을 실행할 때에는 적용되지 않으며, 리포트 뷰어에서 데이터를 요청할 경우에만 적용되는 속성입니다. 이 속성을 이용하려면 뷰어의 패치 단위 설정 패러미터를 데이터 셋 단위로 변경하시기 바랍니다.

See Also

[ConcurrentFetchSize](#)

Concurrent 첫 전송 ROW

클라이언트에서 데이터 요청 시 우선적으로 먼저 전송할 데이터의 행수

Reference

대용량 데이터일 경우 첫 데이터를 빠르게 받아 응답 속도를 향상시키고자 할 때 유용하게 사용될 수 있습니다.

속성 값을 0으로 설정하면 서버에 설정된 값이 적용됩니다.

이 속성 값은 쿼리 디자이너에서 쿼리문을 실행할 때에는 적용되지 않으며, 리포트 뷰어에서 데이터를 요청할 경우에만 적용되는 속성입니다. 이 속성을 이용하려면 뷰어의 패치 단위 설정 패러미터를 데이터 셋 단위로 변경하시기 바랍니다.

See Also

[ConcurrentFirstRow](#)

바인딩 전 데이터 보내기

데이터 셋 바인딩 전에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부

Values

True	데이터 셋 바인딩 전에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비움
False	바인딩 전 데이터 보내기 기능을 사용하지 않음

Reference

이 기능은 패치 타입이 concurrent이고 패치 단위가 데이터 모듈인 경우에만 적용됩니다.

See Also

[FlushDataBeforeBinding](#)

바인딩 후 데이터 보내기

데이터 셋 바인딩 후에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부

Values

True	데이터 셋 바인딩 후에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비움
False	바인딩 후 데이터 보내기 기능을 사용하지 않음

Reference

이 기능은 패치 타입이 concurrent이고 패치 단위가 데이터 모듈인 경우에만 적용됩니다.

See Also

[FlushDataAfterBinding](#)

설명

데이터 셋 설명

Values

데이터 셋의 설명을 설정합니다.

SOAP 데이터 셋

웹 서비스를 통하여 얻어온 데이터를 데이터 소스로 사용합니다.

Name	Description
이름	SOAP 스토어 데이터 셋 이름
마스터셋	마스터셋 이름
숨김	데이터 셋을 숨길지 여부
Concurrent 전송 크기	Concurrent 전송 크기
Concurrent 첫 전송 ROW	클라이언트에서 데이터 요청 시 우선적으로 먼저 전송할 데이터의 행수
바인딩 전 데이터 보내기	데이터 셋 바인딩 전에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부
바인딩 후 데이터 보내기	데이터 셋 바인딩 후에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부
기준 XML 경로	기준 XML 경로
설명	데이터 셋 설명

이름

SOAP 스토어 데이터 셋 이름

마스터셋

마스터셋 이름

Reference

해당 데이터 셋의 마스터 데이터 셋을 설정합니다.

숨김

데이터 셋을 숨길지 여부

Values

True	데이터 셋을 숨김
False	데이터 셋을 숨기지 않음

Reference

데이터 셋을 숨길 경우 해당 데이터 셋은 리포트 디자이너의 데이터 트리에 표시되지 않고, 자바 스크립트로도 접근할 수 없습니다.

또한, 오즈 서버가 데이터를 데이터 모듈 단위로 전송할 때 숨김 상태인 데이터 셋은 제외하고 전송합니다. 그룹 데이터 스토어를 사용할 경우 원본셋을 숨김으로 설정하면 원본셋은 서버에서 전송하지 않습니다.

Concurrent 전송 크기

Concurrent 전송 크기

Reference

클라이언트에서 데이터 요청 시 한꺼번에 가져올 결과 데이터 셋의 크기를 Byte 단위로 설정합니다.

설정된 전송 크기만큼의 버퍼 단위로 데이터를 내보냅니다.

속성 값을 0으로 설정하면 서버에 설정된 값이 적용됩니다.

이 속성 값은 쿼리 디자이너에서 쿼리문을 실행할 때에는 적용되지 않으며, 리포트 뷰어에서 데이터를 요청할 경우에만 적용되는 속성입니다. 이 속성을 이용하려면 뷰어의 패치 단위 설정 패러미터를 데이터 셋 단위로 변경하시기 바랍니다.

Concurrent 첫 전송 ROW

클라이언트에서 데이터 요청 시 우선적으로 먼저 전송할 데이터의 행수

Reference

대용량 데이터일 경우 첫 데이터를 빠르게 받아 응답 속도를 향상시키고자 할 때 유용하게 사용될 수 있습니다.

속성 값을 0으로 설정하면 서버에 설정된 값이 적용됩니다.

이 속성 값은 쿼리 디자이너에서 쿼리문을 실행할 때에는 적용되지 않으며, 리포트 뷰어에서 데이터를 요청할 경우에만 적용되는 속성입니다. 이 속성을 이용하려면 뷰어의 패치 단위 설정 패러미터를 데이터 셋 단위로 변경하시기 바랍니다.

바인딩 전 데이터 보내기

데이터 셋 바인딩 전에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부

Values

True	데이터 셋 바인딩 전에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비움
False	바인딩 전 데이터 보내기 기능을 사용하지 않음

Reference

이 기능은 패치 타입이 concurrent이고 패치 단위가 데이터 모듈인 경우에만 적용됩니다.

바인딩 후 데이터 보내기

데이터 셋 바인딩 후에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부

Values

True	데이터 셋 바인딩 후에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비움
False	바인딩 후 데이터 보내기 기능을 사용하지 않음

Reference

이 기능은 패치 타입이 concurrent이고 패치 단위가 데이터 모듈인 경우에만 적용됩니다.

기준 XML 경로

기준 XML 경로

설명

데이터 셋 설명

Values

데이터 셋의 설명을 설정합니다.

ADO DataSet 웹서비스 데이터 셋

ADO DataSet 웹서비스 스토어는 닷넷 웹 서비스에서 ADO DataSet 객체를 데이터 소스로 사용합니다.

Name	Description
이름	ADO 멀티 데이터 셋 이름
마스터셋	마스터셋 이름
EndPoint	정의된 EndPoint 표시
SOAPAction	SOAPAction 정보 표시

이름

ADO 멀티 데이터 셋 이름

마스터셋

마스터셋 이름

Reference

해당 데이터 셋의 마스터 데이터 셋을 설정합니다.

EndPoint

설정된 EndPoint

SOAPAction

SOAPAction 정보

XML 데이터 셋

DTD/XSD를 이용하는 XML 형식의 데이터를 데이터 소스로 사용합니다.

Name	Description
이름	XML 스토어 데이터 셋 이름
마스터셋	마스터셋 이름
숨김	데이터 셋을 숨길지 여부
Concurrent 전송 크기	Concurrent 전송 크기
Concurrent 첫 전송 ROW	클라이언트에서 데이터 요청 시 우선적으로 먼저 전송할 데이터의 행수
바인딩 전 데이터 보내기	데이터 셋 바인딩 전에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부
바인딩 후 데이터 보내기	데이터 셋 바인딩 후에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부
설명	데이터 셋 설명

이름

XML 스토어 데이터 셋 이름

See Also

[GetName](#)

마스터셋

마스터셋 이름

Reference

해당 데이터 셋의 마스터 데이터 셋을 설정합니다.

See Also

[GetMasterDataSetName](#)

숨김

데이터 셋을 숨길지 여부

Values

True	데이터 셋을 숨김
False	데이터 셋을 숨기지 않음

Reference

데이터 셋을 숨길 경우 해당 데이터 셋은 리포트 디자이너의 데이터 트리에 표시되지 않고, 자바 스크립트로도 접근할 수 없습니다.

또한, 오즈 서버가 데이터를 데이터 모듈 단위로 전송할 때 숨김 상태인 데이터 셋은 제외하고 전송합니다. 그룹 데이터 스토어를 사용할 경우 원본셋을 숨김으로 설정하면 원본셋은 서버에서 전송하지 않습니다.

See Also

[Hidden](#)

Concurrent 전송 크기

Concurrent 전송 크기

Reference

클라이언트에서 데이터 요청 시 한꺼번에 가져올 결과 데이터 셋의 크기를 Byte 단위로 설정합니다.

설정된 전송 크기만큼의 버퍼 단위로 데이터를 내보냅니다

속성 값을 0으로 설정하면 서버에 설정된 값이 적용됩니다.

이 속성 값은 쿼리 디자이너에서 쿼리문을 실행할 때에는 적용되지 않으며, 리포트 뷰어에서 데이터를 요청할 경우에만 적용되는 속성입니다. 이 속성을 이용하려면 뷰어의 패치 단위 설정 패러미터를 데이터 셋 단위로 변경하시기 바랍니다.

See Also

[ConcurrentFetchSize](#)

Concurrent 첫 전송 ROW

클라이언트에서 데이터 요청 시 우선적으로 먼저 전송할 데이터의 행수

Reference

대용량 데이터일 경우 첫 데이터를 빠르게 받아 응답 속도를 향상시키고자 할 때 유용하게 사용될 수 있습니다.

속성 값을 0으로 설정하면 서버에 설정된 값이 적용됩니다.

이 속성 값은 쿼리 디자이너에서 쿼리문을 실행할 때에는 적용되지 않으며, 리포트 뷰어에서 데이터를 요청할 경우에만 적용되는 속성입니다. 이 속성을 이용하려면 뷰어의 패치 단위 설정 패러미터를 데이터 셋 단위로 변경하시기 바랍니다.

See Also

[ConcurrentFirstRow](#)

바인딩 전 데이터 보내기

데이터 셋 바인딩 전에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부

Values

True	데이터 셋 바인딩 전에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비움
False	바인딩 전 데이터 보내기 기능을 사용하지 않음

Reference

이 기능은 패치 타입이 concurrent이고 패치 단위가 데이터 모듈인 경우에만 적용됩니다.

See Also

[FlushDataBeforeBinding](#)

바인딩 후 데이터 보내기

데이터 셋 바인딩 후에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부

Values

True	데이터 셋 바인딩 후에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비움
False	바인딩 후 데이터 보내기 기능을 사용하지 않음

Reference

이 기능은 패치 타입이 concurrent이고 패치 단위가 데이터 모듈인 경우에만 적용됩니다.

See Also

[FlushDataAfterBinding](#)

설명

데이터 셋 설명

Values

데이터 셋의 설명을 설정합니다.

SAP 데이터 셋

SAP R/3에 정의된 RFC(Remote Function Call) 실행 결과를 데이터 소스로 사용합니다.

Name	Description
이름	SAP 스토어 데이터 셋 이름
데이터 셋	RFC를 통해 생성된 데이터 셋 이름
숨김	데이터 셋을 숨길지 여부
Concurrent 전송 크기	Concurrent 전송 크기
Concurrent 첫 전송 ROW	클라이언트에서 데이터 요청 시 우선적으로 먼저 전송할 데이터의 행수
바인딩 전 데이터 보내기	데이터 셋 바인딩 전에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부
바인딩 후 데이터 보내기	데이터 셋 바인딩 후에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부
설명	데이터 셋 설명

이름

SAP 스토어 데이터 셋 이름

See Also

[GetName](#)

데이터셋

RFC를 통해 생성된 데이터 셋 이름

숨김

데이터 셋을 숨길지 여부

Values

True	데이터 셋을 숨김
False	데이터 셋을 숨기지 않음

Reference

데이터 셋을 숨길 경우 해당 데이터 셋은 리포트 디자이너의 데이터 트리에 표시되지 않고, 자바 스크립트로도 접근할 수 없습니다.

또한, 오즈 서버가 데이터를 데이터 모듈 단위로 전송할 때 숨김 상태인 데이터 셋은 제외하고 전송합니다.

그룹 데이터 스토어를 사용할 경우 원본셋을 숨김으로 설정하면 원본셋은 서버에서 전송하지 않습니다.

See Also

[Hidden](#)

Concurrent 전송 크기

Concurrent 전송 크기

Reference

클라이언트에서 데이터 요청 시 한꺼번에 가져올 결과 데이터 셋의 크기를 Byte 단위로 설정합니다.

설정된 전송 크기만큼의 버퍼 단위로 데이터를 내보냅니다

속성 값을 0으로 설정하면 서버에 설정된 값이 적용됩니다.

이 속성 값은 쿼리 디자이너에서 쿼리문을 실행할 때에는 적용되지 않으며, 리포트 뷰어에서 데이터를 요청할 경우에만 적용되는 속성입니다. 이 속성을 이용하려면 뷰어의 패치 단위 설정 패러미터를 데이터 셋 단위로 변경하시기 바랍니다.

See Also

[ConcurrentFetchSize](#)

Concurrent 첫 전송 ROW

클라이언트에서 데이터 요청 시 우선적으로 먼저 전송할 데이터의 행수

Reference

대용량 데이터일 경우 첫 데이터를 빠르게 받아 응답 속도를 향상시키고자 할 때 유용하게 사용될 수 있습니다.

속성 값을 0으로 설정하면 서버에 설정된 값이 적용됩니다.

이 속성 값은 쿼리 디자이너에서 쿼리문을 실행할 때에는 적용되지 않으며, 리포트 뷰어에서 데이터를 요청할 경우에만 적용되는 속성입니다. 이 속성을 이용하려면 뷰어의 패치 단위 설정 패러미터를 데이터 셋 단위로 변경하시기 바랍니다.

See Also

[ConcurrentFirstRow](#)

바인딩 전 데이터 보내기

데이터 셋 바인딩 전에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부

Values

True	데이터 셋 바인딩 전에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비움
False	바인딩 전 데이터 보내기 기능을 사용하지 않음

Reference

이 기능은 패치 타입이 concurrent이고 패치 단위가 데이터 모듈인 경우에만 적용됩니다.

See Also

[FlushDataBeforeBinding](#)

바인딩 후 데이터 보내기

데이터 셋 바인딩 후에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부

Values

True	데이터 셋 바인딩 후에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비움
False	바인딩 후 데이터 보내기 기능을 사용하지 않음

Reference

이 기능은 패치 타입이 concurrent이고 패치 단위가 데이터 모듈인 경우에만 적용됩니다.

See Also

[FlushDataAfterBinding](#)

설명

데이터 셋 설명

Values

데이터 셋의 설명을 설정합니다.

CLEAR QUEST 데이터 셋

형상 관리 툴인 IBM Rational Clear Quest 프로그램에서 생성한 DB를 통하여 얻어온 데이터를 데이터 소스로 사용합니다.

Name	Description
이름	CLEAR QUEST 스토어 데이터 셋 이름
타입	CLEAR QUEST 스토어 데이터 타입
마스터셋	마스터셋 이름
최대행 수	최대 행 수
스크립트 사용	스크립트를 이용한 쿼리문 사용이 가능하도록 할지 여부
숨김	데이터 셋을 숨길지 여부
Concurrent 전송 크기	Concurrent 전송 크기
Concurrent 첫 전송 ROW	클라이언트에서 데이터 요청 시 우선적으로 먼저 전송할 데이터의 행수

바인딩 전 데이터 보내기	데이터 셋 바인딩 전에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부
바인딩 후 데이터 보내기	데이터 셋 바인딩 후에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부
설명	데이터 셋 설명

이름

CLEAR QUEST 스토어 데이터 셋 이름

타입

CLEAR QUEST 스토어 데이터 타입

마스터셋

마스터셋 이름

Reference

해당 데이터 셋의 마스터 데이터 셋을 설정합니다.

최대행 수

최대 행 수

Reference

바인딩 시 해당 데이터 셋의 최대 행(Row) 수를 설정합니다.

0 ~ 2147483647 사이의 값으로 최대 행 수를 설정합니다.

스크립트 사용

스크립트를 이용한 쿼리문 사용이 가능하도록 할지 여부

Values

True	스크립트를 이용한 쿼리문 사용 가능
------	---------------------

False	스크립트를 이용한 쿼리문을 사용할 수 없음
-------	-------------------------

숨김

데이터 셋을 숨길지 여부

Values

True	데이터 셋을 숨김
False	데이터 셋을 숨기지 않음

Reference

데이터 셋을 숨길 경우 해당 데이터 셋은 리포트 디자이너의 데이터 트리에 표시되지 않고, 자바 스크립트로도 접근할 수 없습니다.

또한, 오즈 서버가 데이터를 데이터 모듈 단위로 전송할 때 숨김 상태인 데이터 셋은 제외하고 전송합니다. 그룹 데이터 스토어를 사용할 경우 원본셋을 숨김으로 설정하면 원본셋은 서버에서 전송하지 않습니다.

Concurrent 전송 크기

Concurrent 전송 크기

Reference

클라이언트에서 데이터 요청 시 한꺼번에 가져올 결과 데이터 셋의 크기를 Byte 단위로 설정합니다.

설정된 전송 크기만큼의 버퍼 단위로 데이터를 내보냅니다.

속성 값을 0으로 설정하면 서버에 설정된 값이 적용됩니다.

이 속성 값은 쿼리 디자이너에서 쿼리문을 실행할 때에는 적용되지 않으며, 리포트 뷰어에서 데이터를 요청할 경우에만 적용되는 속성입니다. 이 속성을 이용하려면 뷰어의 패치 단위 설정 패러미터를 데이터 셋 단위로 변경하시기 바랍니다.

Concurrent 첫 전송 ROW

클라이언트에서 데이터 요청 시 우선적으로 먼저 전송할 데이터의 행수

Reference

대용량 데이터일 경우 첫 데이터를 빠르게 받아 응답 속도를 향상시키고자 할 때 유용하게 사용될 수 있습니다.

속성 값을 0으로 설정하면 서버에 설정된 값이 적용됩니다.

이 속성 값은 쿼리 디자이너에서 쿼리문을 실행할 때에는 적용되지 않으며, 리포트 뷰어에서 데이터를 요청할 경우에만 적용되는 속성입니다. 이 속성을 이용하려면 뷰어의 패치 단위 설정 패러미터를 데이터 셋 단위로 변경하시기 바랍니다.

바인딩 전 데이터 보내기

데이터 셋 바인딩 전에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부

Values

True	데이터 셋 바인딩 전에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비움
False	바인딩 전 데이터 보내기 기능을 사용하지 않음

Reference

이 기능은 패치 타입이 concurrent이고 패치 단위가 데이터 모듈인 경우에만 적용됩니다.

바인딩 후 데이터 보내기

데이터 셋 바인딩 후에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부

Values

True	데이터 셋 바인딩 후에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비움
False	바인딩 후 데이터 보내기 기능을 사용하지 않음

Reference

이 기능은 패치 타입이 concurrent이고 패치 단위가 데이터 모듈인 경우에만 적용됩니다.

설명

데이터 셋 설명

Values

데이터 셋의 설명을 설정합니다.

오즈 데이터 필드

- [데이터 셋 필드](#)
- [SOAP 데이터 셋 필드](#)
- [XML 데이터 셋 필드](#)

데이터 셋 필드

데이터 셋의 필드 정보를 관리합니다.

Name	Description
이름	데이터 필드 이름A
타입	데이터 필드 타입
복호화	데이터 필드 복호화 여부
설명	데이터 필드 설명

이름

데이터 필드 이름

See Also

[GetFieldName](#)

타입

데이터 필드 타입

See Also

[GetFieldType](#)

복호화

데이터 필드 복호화 여부

Values

True	복호화
False	복호화하지 않음 (기본 값)

Reference

해당 기능은 자바 서버에서만 지원하며, 복호화 여부는 오즈 뷰어에서 확인 가능합니다.

해당 기능을 사용하려면 자바 서버 설정 파일 중 spmgr.properties 파일의 datafield_usersecurity.Active 옵션을 true로 설정하고, datafield_usersecurity.Class 옵션에 복호화 클래스를 등록하여야 합니다.

See Also

[IsDecrypt](#)

설명

데이터 필드 설명

SOAP 데이터 셋 필드

SOAP 데이터 셋의 필드 정보를 관리합니다.

Name	Description
이름	SOAP 데이터 필드 이름
타입	SOAP 데이터 필드 타입
XPath	SOAP 데이터 필드 XPath
복호화	SOAP 데이터 필드 복호화 여부
설명	SOAP 데이터 필드 설명

이름

SOAP 데이터 필드 이름

타입

SOAP 데이터 필드 타입

XPath

SOAP 데이터 필드 XPath

복호화

SOAP 데이터 필드 복호화 여부

Values

True	복호화
False	복호화하지 않음 (기본 값)

Reference

해당 기능은 자바 서버에서만 지원하며, 복호화 여부는 오즈 뷰어에서 확인 가능합니다.

해당 기능을 사용하려면 자바 서버 설정 파일 중 spmgr.properties 파일의 datafield_usersecurity.Active 옵션을 true로 설정하고, datafield_usersecurity.Class 옵션에 복호화 클래스를 등록하여야 합니다.

설명

SOAP 데이터 필드 설명

XML 데이터 셋 필드

XML 데이터 셋의 필드 정보를 관리합니다.

Name	Description
이름	XML 데이터 필드 이름
타입	XML 데이터 필드 타입
XPath	XML 데이터 필드 XPath
복호화	XML 데이터 필드 복호화 여부
설명	XML 데이터 필드 설명

이름

XML 데이터 필드 이름

See Also

[GetFieldName](#)

타입

XML 데이터 필드 타입

See Also

[GetFieldType](#)

XPath

XML 데이터 필드 XPath

See Also

[GetXPath](#)

복호화

XML 데이터 필드 복호화 여부

Values

True	복호화
False	복호화하지 않음 (기본 값)

Reference

해당 기능은 자바 서버에서만 지원하며, 복호화 여부는 오즈 뷰어에서 확인 가능합니다.

해당 기능을 사용하려면 자바 서버 설정 파일 중 spmgr.properties 파일의 datafield_usersecurity.Active 옵션을 true로 설정하고, datafield_usersecurity.Class 옵션에 복호화 클래스를 등록하여야 합니다.

See Also

IsDecrypt

설명

XML 데이터 필드 설명

사용자 지정 패러미터

사용자가 임의로 정의하여 사용하는 패러미터 변수 목록을 관리합니다.

Name	Description
이름	사용자 지정 패러미터 이름
타입	사용자 지정 패러미터 타입
값	사용자 지정 패러미터 값
암호화	사용자 지정 패러미터 암호화 여부
세션 키	사용자 지정 패러미터의 값을 설정할 세션 키
설명	사용자 지정 패러미터 설명

이름

사용자 지정 패러미터 이름

타입

사용자 지정 패러미터 타입

값

사용자 지정 패러미터 값

Reference

사용자 지정 패러미터의 초기 값으로 디자인 시 보고서의 테스트를 위해 사용자가 임의로 설정할 수 있습니다.

암호화

사용자 지정 패러미터 암호화 여부

Values

True	암호화
False	암호화하지 않음 (기본 값)

Reference

해당 기능을 사용하려면 자바 또는 닷넷 서버 설정 파일 중 `spmgr.properties` 파일의 `odiparam_usersecurity.Active` 옵션을 `true`로 설정하고, `odiparam_usersecurity.Class` 옵션에 암호화 클래스를 등록하여야 합니다.

세션 키

사용자 지정 패러미터의 값을 설정할 세션 키

Reference

세션 키 속성은 서블릿 타입 서버에서만 적용됩니다.

세션 키 속성은 보안을 강화하기 위해 보고서에서 사용자 지정 패러미터 값을 뷰어 패러미터에서 설정하지 않고, 서버 세션을 이용하여 패러미터 값을 설정할 때 사용합니다.

설명

사용자 지정 패러미터 설명

Include 필드

기존 작성되어 리파지토리 서버 혹은 LocalExplorer에 있는 ODI 파일을 Include한 스토어를 그대로 사용합니다.

Name	Description
이름	Include된 필드 이름
OZURL	Include된 ODI 파일 경로

이름

Include된 필드 이름

OZURL

Include된 ODI 파일 경로

Reference

Include된 ODI 파일의 카테고리나 파일 이름을 URL 형태로 표시합니다.

설정된 ODI 파일에 대한 경로인 OZ URL은 리파지토리 서버에 연결되어 있는 경우에는 [Repository Root]의 경로를 의미하며, 리파지토리 서버에 연결되어 있지 않은 경우에는 [LocalExplorer]의 경로를 의미합니다.

예) ozp:///카테고리명/파일명

자바 스크립트

오즈 쿼리 디자이너에서는 오즈 데이터 객체를 제어하는 자바 스크립트 기반의 함수를 제공합니다.

본 장에서는 오즈 쿼리 디자이너에서 제공하는 자바 스크립트 API와 문법, 자바 스크립트 API를 입력하고 비교하는 방법 등에 대해 설명합니다.

스크립트 입력, 비교

본 장에서는 오즈 쿼리 디자이너에서 자바 스크립트를 입력하고, 입력된 스크립트를 비교하는 방법에 대해 설명합니다.

스크립트 입력

스크립트를 입력할 오즈 데이터 객체의 이벤트를 선택한 후 스크립트 편집 창에 스크립트를 입력합니다. 오즈 데이터 트리 창 또는 스크립트 편집 창의 오즈 데이터 객체 콤보박스에서 스크립트를 입력할 오즈 데이터 객체를 선택한 후 스크립트 편집 창의 이벤트 콤보박스를 선택하여 스크립트 창을 활성화합니다. 해당 이벤트에서 제어할 스크립트를 입력합니다.

스크립트 미리보기

메뉴바에서 [편집] - [스크립트 미리보기] 메뉴를 클릭하면 스크립트 미리보기 창에 ODI에 입력되어 있는 모든 스크립트가 표시됩니다.

스크립트 자동 완성 규칙

스크립트 편집 창에 스크립트 입력 시 입력한 문자열에 따라 속성, 함수, 상수가 자동 완성되어 표시됩니다. 속성, 함수, 상수가 자동 완성되는 규칙은 다음 표와 같습니다.

입력 방식	설명
"This." 없이 <Ctrl + Space> 키를 눌렀을 경우	모든 속성, 함수, 상수가 팝업 헬프 창에 표시됩니다.
"This." 없이 "문자열" 입력 후 <Ctrl + Space> 키를 눌렀을 경우	입력한 문자열로 시작되는 모든 속성, 함수, 상수가 팝업 헬프 창에 자동으로 선택되어 표시됩니다.
"This." 입력 후 <Ctrl + Space> 키를 눌렀을 경우	해당 오즈 데이터 객체에서 사용할 수 있는 속성 또는 함수가 팝업 헬프 창에 표시됩니다.
"This.문자열" 입력 후 <Ctrl + Space> 키를 눌렀을 경우	해당 오즈 데이터 객체에서 사용할 수 있는 속성 또는 함수 중 입력한 문자열로 시작되는 속성 또는 함수가 팝업 헬

	프 창에 자동으로 선택되어 표시됩니다. 만일, 입력한 문자열로 시작되는 속성 또는 함수가 유일한 경우에는 팝업 헬프 창이 뜨지 않고 속성 또는 함수의 전체 이름으로 입력됩니다.
"상수명." 입력 후 <Ctrl + Space> 키를 눌렀을 경우	해당 상수 값이 팝업 헬프 창에 표시됩니다.
"상수명.문자열" 입력 후 <Ctrl + Space> 키를 눌렀을 경우	해당 상수 값 중 입력한 문자열로 시작되는 상수 값이 팝업 헬프 창에 자동으로 선택되어 표시됩니다. 만일, 입력한 문자열로 시작되는 상수 값이 유일한 경우에는 팝업 헬프 창이 뜨지 않고 상수 값의 전체 이름으로 입력됩니다.

스크립트 비교

본 장에서는 ODI 파일에 입력된 스크립트를 파일 또는 버전 간 비교하는 방법에 대해 설명합니다.

비교할 파일 및 버전 선택

파일에 입력된 스크립트 비교 시 로컬 파일 간 비교, 리파지토리 서버의 파일과 로컬 파일 비교, 리파지토리 서버의 파일 버전 간 비교, 리파지토리 서버의 파일의 특정 버전과 파일 비교 등 4가지 방법으로 비교할 파일 및 버전을 선택할 수 있습니다.

로컬 파일 간 비교

리파지토리 창의 작업 폴더 또는 로컬 탐색기에서 스크립트를 비교할 ODI 파일을 마우스 오른쪽 버튼을 클릭하여 나타나는 팝업 메뉴에서 [스크립트 비교] 메뉴를 클릭합니다.

"비교 옵션 다이얼로그"의 원본 파일에 선택한 파일의 전체 경로가 표시됩니다.

원본 파일과 스크립트를 비교할 대상 파일을 [...] 버튼을 클릭하면 나타나는 "리파지토리 서버 다이얼로그"에서 선택하면, 대상 파일에 선택한 파일의 전체 경로가 표시됩니다.

리파지토리 서버의 파일과 로컬 파일 비교

리파지토리 창의 리파지토리 서버에서 스크립트를 비교할 ODI 파일을 마우스 오른쪽 버튼을 클릭하여 나타나는 팝업 메뉴에서 [스크립트 비교] 메뉴를 클릭합니다.

"비교 옵션 다이얼로그"의 원본 파일에 선택한 파일의 전체 경로와 리파지토리 서버 정보가 표시되고, 대상 파일에는 자동으로 작업 폴더에서 동일한 경로에 있는 파일을 찾아 파일의 전체 경로로 표시합니다. 단, 동일한 경로에 있는 파일을 찾지 못했을 경우에는 빈 값으로 표시됩니다.

자동으로 표시된 대상 파일이 아닌 다른 파일을 선택하고자 할 경우 [...] 버튼을 클릭하면 나타나는 "리파지토리 파일 다이얼로그"에서 다른 파일을 선택합니다.

리파지토리 서버의 파일 버전 간 비교

리파지토리 창의 리파지토리 서버에서 스크립트를 비교할 ODI 파일을 마우스 오른쪽 버튼을 클릭하여 나타나는 팝업 메뉴에서 [히스토리 보기] 메뉴를 클릭합니다.

[히스토리 보기] 메뉴를 클릭하면 선택한 아이템의 히스토리를 나타낸 다이얼로그가 나타납니다.

스크립트를 비교할 버전을 선택한 후 [스크립트 비교 아이콘]을 클릭합니다.

리파지토리 서버의 파일의 특정 버전과 파일 비교

리파지토리 창의 리파지토리 서버에서 스크립트를 비교할 ODI 파일을 마우스 오른쪽 버튼을 클릭하여 나타나는 팝업 메뉴에서 [히스토리 보기] 메뉴를 클릭합니다.

[히스토리 보기] 메뉴를 클릭하면 선택한 아이템의 히스토리를 나타낸 다이얼로그가 나타납니다.

스크립트를 비교할 특정 버전을 선택한 후 [스크립트 비교 아이콘]을 클릭합니다.

"비교 옵션 다이얼로그"의 원본 파일에 선택한 파일의 전체 경로, 리파지토리 서버 정보, 버전 정보가 표시되고 대상 파일에는 자동으로 작업 폴더에서 동일한 경로에 있는 파일을 찾아 파일의 전체 경로로 표시합니다. 단, 동일한 경로에 있는 파일을 찾지 못했을 경우에는 빈 값으로 표시됩니다.

자동으로 표시된 대상 파일이 아닌 다른 파일을 선택하고자 할 경우 [...] 버튼을 클릭하면 나타나는 "리파지토리 파일 다이얼로그"에서 다른 파일을 선택합니다.

스크립트 비교

"스크립트 비교 다이얼로그"에서 스크립트를 비교합니다.

스크립트를 비교한 결과를 텍스트의 글꼴 색으로 구분하여 표시합니다.

글꼴 색	설명
파랑	원본 파일에만 스크립트가 있을 경우
초록	대상 파일에만 스크립트가 있을 경우
빨강	같은 라인에 입력된 스크립트가 다를 경우

스크립트가 다른 라인으로 커서를 이동하려면 [다음 차이점 이동], [이전 차이점 이동] 아이콘을 클릭하여 이동할 수 있으며 해당 라인에 빨간색 화살표가 표시됩니다.

스크립트 팝업 헬프 창

스크립트 팝업 헬프 창은 자바 스크립트의 속성, 함수, 상수를 리스트 형태로 나타낸 팝업 창으로, 자바 스크립트 작성을 쉽게 할 수 있도록 도와줍니다.

전체 보기

스크립트 편집 창에서 빈 문자 다음에 <Ctrl + Space> 키를 누르면 스크립트 팝업 헬프 창이 나타납니다. 이 때 나타난 스크립트 팝업 헬프 창에는 모든 속성, 함수, 상수가 아이템으로 표시됩니다.

스크립트 팝업 헬프 창이 뜬 상태에서 속성명, 함수명, 상수명을 입력하면 입력한 문자열에 해당하는 아이템으로 자동으로 포커스가 이동하고, 이 때 <Tab> 키를 누르거나 마우스로 더블 클릭하면 자동으로 입력됩니다.

스크립트 팝업 헬프 창 각각의 아이템 앞에는 해당 아이템이 어떤 내용인지 나타내는 아이콘이 있어서 해당 아이템이 속성, 함수, 상수인지 또는 오즈 데이터 객체인지를 쉽게 식별할 수 있습니다.

각각의 아이콘에 대한 간략한 설명은 다음과 같습니다.

아이콘	설명
	속성
	함수
	상수
	오즈 데이터 모듈
	오즈 데이터 스토어
	오즈 데이터 셋

오즈 데이터 객체 별 함수 보기

"This."을 입력하면 해당 오즈 데이터 객체가 가질 수 있는 속성 및 함수를 아이템으로 나타낸 스크립트 팝업 헬프 창이 자동으로 표시됩니다.

상수 값 보기

상수명을 입력하고 "."를 입력하면 해당 상수가 가질 수 있는 값을 아이템으로 나타낸 스크립트 팝업 헬프 창이 자동으로 표시됩니다.

자바 스크립트 문법

쿼리 디자이너에서 제공하는 자바 스크립트의 기본 문법에 대해 살펴 보겠습니다. 자바 스크립트는 대소문자를 구분합니다.

주석

// 한 줄을 주석으로 처리합니다.

/* */ 사이에 있는 문자를 주석으로 처리합니다.

상수

This 자기 자신의 객체를 지칭합니다. 자바 스크립트에서 속성을 설정하거나 함수를 사용할 경우 객체 이름을 사용하지 않고 "This." 형태로 입력합니다.

자료형

var 변수의 내용에 따라 데이터 형이 결정되는 타입으로 자바 스크립트의 변수 선언 시 사용됩니다. 오즈 쿼리 디자이너에서 자바 스크립트 사용 시에는 var 타입의 자료형만 사용 가능합니다.

연산자

자바 스크립트에서 사용 가능한 연산자는 다음과 같습니다.

• 산술 연산자

- + 더하기/문자열 접합(Concatenation)
- 빼기/단항 연산자(Unary operator)
- * 곱하기
- / 나누기
- % 나머지 연산자(Modules)

• 대입 연산자

- = 단순대입

• 비교 연산자

== 동등
!= 동등하지 않음
< 작다
<= 작거나 같다
> 크다
>= 크거나 같다

• 비교 연산자

& 비트 AND(int형만 가능)
| 비트 OR(int형만 가능)
^ 비트 XOR(int형만 가능)
~ 부정(int형만 가능)
&& 논리 AND(boolean형만 가능)
|| 논리 OR(boolean형만 가능)
! 부정(boolean형만 가능)

조건문

if-else 문 주어진 조건에 따라 문장을 선택적으로 실행합니다.

반복문

while 문 주어진 조건이 true인 동안 문장을 반복 실행합니다.

자바 스크립트 API

본 장에서는 오즈 쿼리 디자이너에서 제공하는 자바 스크립트의 속성, 함수, 이벤트, 상수 등에 대해 설명합니다.

OZObject

- [DataModule](#)
- [DataStore](#)
- [DataSet](#)
- [DataField](#)

DataModule

- [Property](#)
- [Method](#)
- [Event](#)

Property

Name	Description
ConcurrentFetchSize	Concurrent 바인딩 시 데이터 전송 사이즈
OZParamMark	패러미터 구분자
ShowZeroDecimalPoint	소수점 앞에 0표시 옵션

ConcurrentFetchSize

Concurrent 바인딩 시 데이터 전송 사이즈

Prototype

`integer ConcurrentFetchSize`

Reference

fetch 타입이 Concurrent 경우에만 적용됩니다.

Example

```
var dm = This.GetDataModule();  
_DEBUG("ConcurrentFetchSize " + dm.ConcurrentFetchSize);
```

See Also

[Concurrent 전송 크기](#)

OZParamMark

패러미터 구분자

Prototype

[string OZParamMark](#)

Example

```
var dm = This.GetDataModule();  
_DEBUG("OZParamMark " + dm.OZParamMark);
```

See Also

[사용자 지정 패러미터 구분자](#)

ShowZeroDecimalPoint

소수점 앞에 0표시 옵션

Prototype

[integer ShowZeroDecimalPoint](#)

Reference

자바 서버만 해당됩니다.

서버 설정이 default 경우에만 설정이 유효합니다.

Example

```
var dm = This.GetDataModule();
_DEBUG("ShowZeroDecimalPoint " + dm.ShowZeroDecimalPoint);
```

See Also

[소수점 앞에 0 표시](#), [ShowZeroDecimalPointConst](#)

Method

Name	Description
GetClientDataModuleType	클라이언트 데이터 모듈 타입을 가져옵니다.
GetDataSet	해당 이름의 데이터셋 객체를 가져옵니다.
GetDataSetNames	추가된 데이터셋 이름을 모두 가져옵니다.
GetDataStore	해당 이름의 데이터 스토어를 가져옵니다.
GetDataStoreNames	추가된 데이터 스토어 이름을 모두 가져옵니다.
GetFetchType	fetch 타입을 가져옵니다.
GetFetchUnit	fetch 단위를 가져옵니다.
GetFileName	ODI 파일 이름을 가져옵니다.
GetOZParam	해당 이름의 사용자 지정 패러미터 값을 가져옵니다.
GetOZParamNames	추가된 사용자 패러미터 이름을 모두 가져옵니다.
GetServerDataModuleType	서버 데이터 모듈 타입을 가져옵니다.
GetServerType	서버 타입을 가져옵니다.
IsCachedDataRequest	캐시 데이터 사용 여부를 가져옵니다.
IsCompressedDataRequest	데이터 모듈 압축 전송 여부를 가져옵니다.
IsDesignTime	실행 시점이 Design Time인지 여부를 가져옵니다.
IsEnableDataSet	해당 이름의 데이터셋 사용 여부를 가져옵니다.

GetClientDataModuleType

클라이언트 데이터 모듈 타입을 가져옵니다.

Prototype

integer [GetClientDataModuleType\(\)](#)

Example

```
var dm = This.GetDataModule();  
var ClientDataModuleType = dm.GetClientDataModuleType();
```

See Also

[ODIClientDMTypeConst](#)

GetDataSet

해당 이름의 데이터셋 객체를 가져옵니다.

Prototype

object [GetDataSet\(string DataSetName\)](#)

Parameters

DataSetName	데이터셋 이름
-------------	---------

Example

```
var dm = This.GetDataModule();  
var DataSetName = dm.GetDataSet("dataset1");
```

GetDataSetNames

추가된 데이터셋 이름을 모두 가져옵니다.

Prototype

array [GetDataSetNames\(\)](#)

Example

```
var dm = This.GetDataModule();  
var DataSetNames = new Array();
```

```
DataSetNames = dm.GetDataSetNames();
```

GetDataStore

해당 이름의 데이터 스토어를 가져옵니다.

Prototype

object GetDataStore(string DataStoreName)

Parameters

DataStoreName	데이터 스토어 이름
---------------	------------

Example

```
var dm = This.GetDataModule();  
var DataStoreName = dm.GetDataStore("datastore1");
```

GetDataSetNames

추가된 데이터 스토어 이름을 모두 가져옵니다.

Prototype

array GetDataStoreNames()

Example

```
var dm = This.GetDataModule();  
var DataStoreNames = new Array();  
DataStoreNames = dm.GetDataStoreNames();
```

GetServerType

fetch 타입을 가져옵니다.

Prototype

integer GetFetchType()

Example

```
var dm = This.GetDataModule();  
var FetchType = dm.GetFetchType();
```

See Also

[ODIFetchTypeConst](#)

GetFetchUnit

fetch 단위를 가져옵니다.

Prototype

integer GetFetchUnit()

Reference

바인딩 요청 시 사용 가능합니다.

Example

```
var dm = This.GetDataModule();  
var FetchUnit = dm.GetFetchUnit();
```

See Also

[ODIFetchUnitConst](#)

GetFileName

ODI 파일 이름을 가져옵니다.

Prototype

string GetFileName()

Example

```
var dm = This.GetDataModule();  
var FileName = dm.GetFileName();
```

GetOZParam

해당 이름의 사용자 지정 패러미터 값을 가져옵니다.

Prototype

string GetOZParam(string OZParamName)

Parameters

OZParamName	사용자 지정 패러미터 이름
-------------	----------------

Example

```
var dm = This.GetDataModule();  
var OZParamName = dm.GetOZParam("ozparam1");
```

GetOZParamNames

추가된 사용자 패러미터 이름을 모두 가져옵니다.

Prototype

array GetOZParamNames()

Example

```
var dm = This.GetDataModule();  
var OZParamNames = new Array();  
OZParamNames = dm.GetOZParamNames();
```

GetServerDataModuleType

서버 데이터 모듈 타입을 가져옵니다.

Prototype

integer GetServerDataModuleType()

Reference

바인딩 요청 시 사용 가능합니다.

Example

```
var dm = This.GetDataModule();  
var ServerDataModuleType = dm.GetServerDataModuleType();
```

See Also

[ODIServerDMTypeConst](#)

GetServerType

서버 타입을 가져옵니다.

Prototype

integer GetServerType()

Example

```
var dm = This.GetDataModule();  
var ServerType = dm.GetServerType();
```

See Also

[OZServerTypeConst](#)

IsCachedDataRequest

캐시 데이터 사용 여부를 가져옵니다.

Prototype

bool IsCachedDataRequest()

Reference

바인딩 요청 시 사용 가능합니다.

Example

```
var dm = This.GetDataModule();  
if(dm.IsCachedDataRequest() == true) {  
    _DEBUG("Return true");  
} else {  
    _DEBUG("Return false");  
}
```

IsCompressedDataRequest

데이터 모듈 압축 전송 여부를 가져옵니다.

Prototype

bool IsCompressedDataRequest()

Reference

바인딩 요청 시 사용 가능합니다.

Example

```
var dm = This.GetDataModule();
if(dm.IsCompressedDataRequest() == true) {
    _DEBUG("Return true");
} else {
    _DEBUG("Return false");
}
```

IsDesignTime

실행 시점이 Design Time인지 여부를 가져옵니다.

Prototype

bool IsDesignTime()

Reference

바인딩 요청 시 사용 가능합니다.

Example

```
var dm = This.GetDataModule();
if(dm.IsDesignTime() == true) {
    _DEBUG("Return true");
} else {
    _DEBUG("Return false");
}
```

IsEnableDataSet

해당 이름의 데이터셋 사용 여부를 가져옵니다.

Prototype

bool IsEnableDataSet(string DataSetName)

Parameters

DataSetName	데이터셋 이름
-------------	---------

Example

```
var dm = This.GetDataModule();  
if(dm.IsEnableDataSet("dataset1") == true) {  
    _DEBUG("Return true");  
} else {  
    _DEBUG("Return false");  
}
```

Event

Name	Description
Functions	모든 바인딩 및 데이터 액션 전에 실행됩니다.
OnPostBind	바인딩 요청을 처리 완료한 후에 발생합니다.
OnPostDataAction	요청한 데이터 액션을 처리 완료한 후에 발생합니다.
OnPreBind	바인딩 요청을 처리하기 직전에 발생합니다.
OnPreDataAction	요청한 데이터 액션을 처리하기 직전에 발생합니다.

Functions

모든 바인딩 및 데이터 액션 전에 실행됩니다.

Prototype

void Functions()

Reference

전역 변수, 함수를 선언할 수 있습니다.

OnPostBind

바인딩 요청을 처리 완료한 후에 발생합니다.

Prototype

`void OnPostBind()`

OnPostDataAction

요청한 데이터 액션을 처리 완료한 후에 발생합니다.

Prototype

`void OnPostDataAction()`

OnPreBind

바인딩 요청을 처리하기 직전에 발생합니다.

Prototype

`void OnPreBind()`

OnPreDataAction

요청한 데이터 액션을 처리하기 직전에 발생합니다.

Prototype

`void OnPreDataAction()`

DataStore

- [OZDBStore](#)
- [OZGroupDataStore](#)
- [OZSAPStore](#)
- [OZSOAPStore](#)
- [OZUserDataStore](#)
- [OZXPathStore](#)

OZDBStore

- [Property](#)
- [Method](#)
- [Event](#)

Property

Name	Description
AutoCommit	자동 커밋 여부
ConnectionPoolAlias	데이터베이스 커넥션 풀의 앨리어스 이름
DecodeCharaterSet	데이터베이스 커넥션의 디코딩 셋
EncodeCharacterSet	데이터베이스 커넥션의 인코딩 셋
UseConnectionPoolAlias	데이터베이스 커넥션 풀의 앨리어스 정보 파일 사용 여부
VendorName	데이터베이스 벤더 이름

AutoCommit

자동 커밋 여부

Prototype

bool [AutoCommit](#)

Example

```
var dst = This.GetDataStore();  
_DEBUG("AutoCommit " + dst.AutoCommit);
```

See Also

[자동 커밋](#)

ConnectionPoolAlias

데이터베이스 커넥션 풀의 앨리어스 이름

Prototype

string [ConnectionPoolAlias](#)

Example

```
var dst = This.GetDataStore();  
_DEBUG("ConnectionPoolAlias " + dst.ConnectionPoolAlias);
```

See Also

[앨리어스 이름](#)

DecodeCharaterSet

데이터베이스 커넥션의 디코딩 셋

Prototype

string [DecodeCharaterSet](#)

Example

```
var dst = This.GetDataStore();  
_DEBUG("DecodeCharaterSet " + dst.DecodeCharaterSet);
```

EncodeCharacterSet

데이터베이스 커넥션의 인코딩 셋

Prototype

string [EncodeCharacterSet](#)

Example

```
var dst = This.GetDataStore();  
_DEBUG("EncodeCharacterSet " + dst.EncodeCharacterSet);
```

UseConnectionPoolAlias

데이터베이스 커넥션 풀의 앨리어스 정보파일 사용 여부

Prototype

bool UseConnectionPoolAlias

Example

```
var dst = This.GetDataStore();  
_DEBUG("UseConnectionPoolAlias " + dst.UseConnectionPoolAlias);
```

See Also

[앨리어스 정보파일 사용](#)

VendorName

데이터베이스 벤더 이름

Prototype

string VendorName

Example

```
var dst = This.GetDataStore();  
_DEBUG("VendorName " + dst.VendorName);
```

See Also

[벤더](#)

Method

Name	Description
GetConnection	현재 데이터 스토어의 데이터베이스 커넥션을 가져옵니다.
GetConnectionKey	오즈 서버의 커넥션 풀에서 현재 데이터 스토어에 매칭되는 커넥션 키 값을 가져옵니다.
GetConnectionKeyItemCount	Connection URL에 필요한 아이템 개수를 가져옵니다.
GetConnectionKeyItemNames	Connection URL에 필요한 아이템 이름을 가져옵니다.
GetConnectionKeyItemValue	커넥션에 필요한 Item 값을 가져옵니다.

GetDataModule	데이터 모듈을 가져옵니다.
GetName	데이터 스토어 이름을 가져옵니다.
SetConnectionKeyItem	지정한 아이템 이름의 아이템 값을 설정합니다.

GetConnection

현재 데이터 스토어의 데이터베이스 커넥션을 가져옵니다.

Prototype

object [GetConnection\(\)](#)

Reference

데이터 스토어가 초기화된 후에 실행할 수 있습니다.

Example

```
var dst = This.GetDataStore();
var Connection = dst.GetConnection();
```

GetConnectionKey

오즈 서버의 커넥션 풀에서 현재 데이터 스토어에 매칭되는 커넥션 키 값을 가져옵니다.

Prototype

string [GetConnectionKey\(\)](#)

Reference

[UseConnectionPoolAlias](#) 속성 값이 true이면 null을 리턴합니다.

Example

```
var dst = This.GetDataStore();
var ConnectionKey = dst.GetConnectionKey();
```

GetConnectionKeyItemCount

Connection URL에 필요한 아이템 개수를 가져옵니다.

Prototype

integer `GetConnectionKeyItemCount()`

Example

```
var dst = This.GetDataStore();  
var ConnectionKeyItemCount = dst.GetConnectionKeyItemCount();
```

GetConnectionKeyItemNames

Connection URL에 필요한 아이템 이름을 가져옵니다.

Prototype

array `GetConnectionKeyItemNames()`

Example

```
var dst = This.GetDataStore();  
var ConnectionKeyItemNames = new Array();  
ConnectionKeyItemNames = dst.GetConnectionKeyItemNames();
```

GetConnectionKeyItemValue

커넥션에 필요한 Item 값을 가져옵니다.

Prototype

string `GetConnectionKeyItemValue(string ConnectionKeyItemName)`

Parameters

ConnectionKeyItemName	아이템 이름
-----------------------	--------

Example

```
var dst = This.GetDataStore();  
var ConnectionKeyItemValue = dst.GetConnectionKeyItemValue("value1");
```

GetDataModule

데이터 모듈을 가져옵니다.

Prototype

object GetDataModule()

Example

```
var dst = This.GetDataStore();  
var DataModule = dst.GetDataModule();
```

GetName

데이터 스토어 이름을 가져옵니다.

Prototype

string GetName()

Example

```
var dst = This.GetDataStore();  
var Name = dst.GetName();
```

See Also

[이름](#)

SetConnectionKeyItem

지정한 아이템 이름의 아이템 값을 설정합니다.

Prototype

void SetConnectionKeyItem(string ConnectionKeyItemName, string
ConnectionKeyItemValue)

Parameters

ConnectionKeyItemName	아이템 이름
ConnectionKeyItemValue	아이템 값

Example

```
var dst = This.GetDataStore();  
dst.SetConnectionKeyItem("Item1", "Value1");
```

Event

Name	Description
OnFinalize	바인딩 요청 시 데이터 스토어 삭제 직전에 발생합니다.
OnInitialize	바인딩 요청 시 데이터 스토어 초기화 직전에 발생합니다.
OnPostInitialize	바인딩 요청 시 데이터 스토어가 초기화된 직후에 발생합니다.
OnTransactionCommit	데이터 액션 요청 시 Commit 실행 직전에 발생합니다.
OnTransactionRollback	데이터 액션 요청 시 Rollback 실행 직전에 발생합니다.
OnTransactionStart	데이터 액션 요청 시 데이터 스토어 초기화 직전에 발생합니다.

OnFinalize

바인딩 요청 시 데이터 스토어 삭제 직전에 발생합니다.

Prototype

`void OnFinalize()`

OnInitialize

바인딩 요청 시 데이터 스토어 초기화 직전에 발생합니다.

Prototype

`void OnInitialize()`

Reference

해당 이벤트에서 속성 값을 변경할 수 있습니다.

OnPostInitialize

바인딩 요청 시 데이터 스토어가 초기화된 직후에 발생합니다.

Prototype

`void OnPostInitialize()`

OnTransactionCommit

데이터 액션 요청 시 Commit 실행 직전에 발생합니다.

Prototype

`void OnTransactionCommit()`

Reference

Client로 전송될 문자열을 리턴할 수 있습니다.

null 또는 string 이외의 타입을 리턴하며 리턴 값이 없을 경우 서버의 commit 로직을 수행합니다.

OnTransactionRollback

데이터 액션 요청 시 Rollback 실행 직전에 발생합니다.

Prototype

`void OnTransactionRollback()`

Reference

Rollback 여부를 리턴할 수 있습니다.

true를 리턴하면 서버의 Rollback 로직은 수행하지 않습니다.

OnTransactionStart

데이터 액션 요청 시 데이터 스토어 초기화 직전에 발생합니다.

Prototype

`void OnTransactionStart()`

OZGroupDataStore

- [Property](#)
- [Method](#)
- [Event](#)

Property

Name	Description
SourceDataSetName	그룹 데이터 스토어의 원본 셋 이름

SourceDataSetName

그룹 데이터 스토어의 원본 셋 이름

Prototype

string [SourceDataSetName](#)

Example

```
var dst = This.GetDataStore();  
_DEBUG("SourceDataSetName " + dst.SourceDataSetName);
```

See Also

[원본 셋](#)

Property

Name	Description
GetDataModule	데이터 모듈을 가져옵니다.
GetName	데이터 스토어 이름을 가져옵니다.

GetDataModule

데이터 모듈을 가져옵니다.

Prototype

object [GetDataModule\(\)](#)

Example

```
var dst = This.GetDataStore();  
var DataModule = dst.GetDataModule();
```

GetName

데이터 스토어 이름을 가져옵니다.

Prototype

string GetName()

Example

```
var dst = This.GetDataStore();  
var Name = dst.GetName();
```

See Also

[이름](#)

Event

Name	Description
OnFinalize	바인딩 요청 시 데이터 스토어 삭제 직전에 발생합니다.
OnInitialize	바인딩 요청 시 데이터 스토어 초기화 직전에 발생합니다.
OnPostInitialize	바인딩 요청 시 데이터 스토어가 초기화된 직후에 발생합니다.

OnFinalize

바인딩 요청 시 데이터 스토어 삭제 직전에 발생합니다.

Prototype

void OnFinalize()

OnInitialize

바인딩 요청 시 데이터 스토어 초기화 직전에 발생합니다.

Prototype

`void OnInitialize()`

Reference

해당 이벤트에서 속성 값을 변경할 수 있습니다.

OnPostInitialize

바인딩 요청 시 데이터 스토어가 초기화된 직후에 발생합니다.

Prototype

`void OnPostInitialize()`

OZSAPStore

- [Property](#)
- [Method](#)
- [Event](#)

Property

Name	Description
ClientName	클라이언트 이름
ConnectionPoolAlias	SAP 스토어 연결 정보 파일
FunctionList	SAP에서 실행할 ABAP 함수 이름
HostName	SAP 서버의 IP 주소
Language	로그인 시 사용할 언어
SID	SAP Netweaver 서버에 설정된 커넥션 풀의 SID
SystemNumber	SAP 서버의 시스템 번호
UseConnectionPoolAlias	SAP 스토어 연결 정보 파일 사용 여부
UserID	사용자 ID

UserPassword	사용자 암호
--------------	--------

ClientName

클라이언트 이름

Prototype

string ClientName

Example

```
var dst = This.GetDataStore();  
_DEBUG("ClientName " + dst.ClientName);
```

See Also

[클라이언트 이름](#)

ConnectionPoolAlias

SAP 스토어 연결 정보 파일

Prototype

string ConnectionPoolAlias

Example

```
var dst = This.GetDataStore();  
_DEBUG("ConnectionPoolAlias " + dst.ConnectionPoolAlias);
```

FunctionList

SAP에서 실행할 ABAP 함수 이름

Prototype

array FunctionList

Example

```
var dst = This.GetDataStore();
```

```
var FunctionList = new Array();
for(var i = 0; i < FunctionList.length; i++){
    _DEBUG("FunctionList[" + i + "] " + dst.FunctionList[i]);
}
```

HostName

SAP 서버의 IP 주소

Prototype

string HostName

Example

```
var dst = This.GetDataStore();
_DEBUG("HostName " + dst.HostName);
```

See Also

[호스트 이름\(주소\)](#)

Language

로그인 시 사용할 언어

Prototype

string Language

Example

```
var dst = This.GetDataStore();
_DEBUG("Language " + dst.Language);
```

See Also

[언어](#)

SID

SAP Netweaver 서버에 설정된 커넥션 풀의 SID

Prototype

string [SID](#)

Example

```
var dst = This.GetDataStore();  
_DEBUG("SID " + dst.SID);
```

See Also

[SID](#)

SystemNumber

SAP 서버의 시스템 번호

Prototype

string [SystemNumber](#)

Example

```
var dst = This.GetDataStore();  
_DEBUG("SystemNumber " + dst.SystemNumber);
```

See Also

[시스템 번호](#)

UseConnectionPoolAlias

SAP 스토어 연결 정보 파일 사용 여부

Prototype

bool [UseConnectionPoolAlias](#)

Example

```
var dst = This.GetDataStore();  
_DEBUG("UseConnectionPoolAlias " + dst.UseConnectionPoolAlias);
```

UserID

사용자 ID

Prototype

`string UserID`

Example

```
var dst = This.GetDataStore();  
_DEBUG("UserID " + dst.UserID);
```

UserPassword

사용자 암호

Prototype

`string UserPassword`

Example

```
var dst = This.GetDataStore();  
_DEBUG("UserPassword " + dst.UserPassword);
```

Method

Name	Description
GetDataModule	데이터 모듈을 가져옵니다.
GetName	데이터 스토어 이름을 가져옵니다.

GetDataModule

데이터 모듈을 가져옵니다.

Prototype

`object GetDataModule()`

Example

```
var dst = This.GetDataStore();  
var DataModule = dst.GetDataModule();
```

GetName

데이터 스토어 이름을 가져옵니다.

Prototype

string GetName()

Example

```
var dst = This.GetDataStore();  
var Name = dst.GetName();
```

Event

Name	Description
OnFinalize	바인딩 요청 시 데이터 스토어 삭제 직전에 발생합니다.
OnInitialize	바인딩 요청 시 데이터 스토어 초기화 직전에 발생합니다.
OnPostInitialize	바인딩 요청 시 데이터 스토어가 초기화된 직후에 발생합니다.

OnFinalize

바인딩 요청 시 데이터 스토어 삭제 직전에 발생합니다.

Prototype

void OnFinalize()

OnInitialize

바인딩 요청 시 데이터 스토어 초기화 직전에 발생합니다.

Prototype

void OnInitialize()

Reference

해당 이벤트에서 속성 값을 변경할 수 있습니다.

OnPostInitialize

바인딩 요청 시 데이터 스토어가 초기화된 직후에 발생합니다.

Prototype

void OnPostInitialize()

OZSOAPStore

- [Property](#)
- [Method](#)
- [Event](#)

Property

Name	Description
EndPoint	실제 서비스를 요청할 URL
SOAPAction	Operation에 정의된 SOAPAction 값
TargetNamespace	WSDL에 정의된 Target 네임스페이스
WSDLURL	WSDL 위치 URL

EndPoint

실제 서비스를 요청할 URL

Prototype

string EndPoint

Example

```
var dst = This.GetDataStore();  
_DEBUG("EndPoint " + dst.EndPoint);
```

SOAPAction

Operation에 정의된 SOAPAction 값

Prototype

string SOAPAction

Example

```
var dst = This.GetDataStore();  
_DEBUG("SOAPAction " + dst.SOAPAction);
```

TargetNamespace

WSDL에 정의된 Target 네임스페이스

Prototype

string TargetNamespace

Example

```
var dst = This.GetDataStore();  
_DEBUG("TargetNamespace " + dst.TargetNamespace);
```

WSDLURL

WSDL 위치 URL

Prototype

string WSDLURL

Example

```
var dst = This.GetDataStore();  
_DEBUG("WSDLURL " + dst.WSDLURL);
```

Method

Name	Description
GetDataModule	데이터 모듈을 가져옵니다.
GetName	데이터 스토어 이름을 가져옵니다.
GetOperationName	Operation 이름을 가져옵니다.
GetPortName	포트 이름을 가져옵니다.
GetServiceName	서비스 이름을 가져옵니다.

GetDataModule

데이터 모듈을 가져옵니다.

Prototype

object [GetDataModule\(\)](#)

Example

```
var dst = This.GetDataStore();  
var DataModule = dst.GetDataModule();
```

GetName

데이터 스토어 이름을 가져옵니다.

Prototype

string [GetName\(\)](#)

Example

```
var dst = This.GetDataStore();  
var Name = dst.GetName();
```

See Also

[이름](#)

GetOperationName

Operation 이름을 가져옵니다.

Prototype

string [GetOperationName\(\)](#)

Example

```
var dst = This.GetDataStore();  
var OperationName = dst.GetOperationName();
```

GetPortName

포트 이름을 가져옵니다.

Prototype

string [GetPortName\(\)](#)

Example

```
var dst = This.GetDataStore();  
var PortName = dst.GetPortName();
```

GetServiceName

서비스 이름을 가져옵니다.

Prototype

string [GetServiceName\(\)](#)

Example

```
var dst = This.GetDataStore();  
var ServiceName = dst.GetServiceName();
```

Event

Name	Description
OnFinalize	바인딩 요청 시 데이터 스토어 삭제 직전에 발생합니다.
OnInitialize	바인딩 요청 시 데이터 스토어 초기화 직전에 발생합니다.

OnPostInitialize	바인딩 요청 시 데이터 스토어가 초기화된 직후에 발생합니다.
------------------	-----------------------------------

OnFinalize

바인딩 요청 시 데이터 스토어 삭제 직전에 발생합니다.

Prototype

`void OnFinalize()`

OnInitialize

바인딩 요청 시 데이터 스토어 초기화 직전에 발생합니다.

Prototype

`void OnInitialize()`

Reference

해당 이벤트에서 속성 값을 변경할 수 있습니다.

OnPostInitialize

바인딩 요청 시 데이터 스토어가 초기화된 직후에 발생합니다.

Prototype

`void OnPostInitialize()`

OZUserDataStore

- [Property](#)
- [Method](#)
- [Event](#)

Property

Name	Description
------	-------------

ClassName	UDS 클래스 이름
-----------	------------

ClassName

UDS 클래스 이름

Prototype

string ClassName

Example

```
var dst = This.GetDataStore();  
_DEBUG("ClassName " + dst.ClassName);
```

See Also

[클래스 이름](#)

Method

Name	Description
GetDataModule	데이터 모듈을 가져옵니다.
GetName	데이터 스토어 이름을 가져옵니다.

GetDataModule

데이터 모듈을 가져옵니다.

Prototype

object GetDataModule()

Example

```
var dst = This.GetDataStore();  
var DataModule = dst.GetDataModule();
```

GetName

데이터 스토어 이름을 가져옵니다.

Prototype

string GetName()

Example

```
var dst = This.GetDataStore();  
var Name = dst.GetName();
```

See Also

[이름](#)

Event

Name	Description
OnFinalize	바인딩 요청 시 데이터 스토어 삭제 직전에 발생합니다.
OnInitialize	바인딩 요청 시 데이터 스토어 초기화 직전에 발생합니다.
OnPostInitialize	바인딩 요청 시 데이터 스토어가 초기화된 직후에 발생합니다.
OnTransactionCommit	데이터 액션 요청 시 Commit 실행 직전에 발생합니다.
OnTransactionRollback	데이터 액션 요청 시 Rollback 실행 직전에 발생합니다.
OnTransactionStart	데이터 액션 요청 시 데이터 스토어 초기화 직전에 발생합니다.

OnFinalize

바인딩 요청 시 데이터 스토어 삭제 직전에 발생합니다.

Prototype

void OnFinalize()

OnInitialize

바인딩 요청 시 데이터 스토어 초기화 직전에 발생합니다.

Prototype

void OnInitialize()

Reference

해당 이벤트에서 속성 값을 변경할 수 있습니다.

OnPostInitialize

바인딩 요청 시 데이터 스토어가 초기화된 직후에 발생합니다.

Prototype

void OnPostInitialize()

OnTransactionCommit

데이터 액션 요청 시 Commit 실행 직전에 발생합니다.

Prototype

void OnTransactionCommit()

Reference

Client로 전송될 문자열을 리턴할 수 있습니다.

null 또는 string 이외의 타입을 리턴하며 리턴 값이 없을 경우 서버의 commit 로직을 수행합니다.

OnTransactionRollback

데이터 액션 요청 시 Rollback 실행 직전에 발생합니다.

Prototype

void OnTransactionRollback()

Reference

Rollback 여부를 리턴할 수 있습니다.

true를 리턴하면 서버의 Rollback 로직은 수행하지 않습니다.

OnTransactionStart

데이터 액션 요청 시 데이터 스토어 초기화 직전에 발생합니다.

Prototype

`void OnTransactionStart()`

OZXPathStore

- [Property](#)
- [Method](#)
- [Event](#)

Property

Name	Description
DoPost	URL 패러미터 전송 시 포스트 방식으로 전송할지 여부
EncodingURL	Server URL 인코딩 여부
SchemaURL	서버에서 운용 시 사용할 XSD 또는 DTD 파일 경로
TestXMLURL	디자인 시 사용할 XML 파일의 로컬 시스템 파일 경로
XMLParser	XML 파서
XMLURL	서버에서 운용 시 사용할 XML 파일 경로

DoPost

URL 패러미터 전송 시 포스트 방식으로 전송할지 여부

Prototype

`bool DoPost`

Example

```
var dst = This.GetDataStore();  
_DEBUG("DoPost " + dst.DoPost);
```

EncodingURL

Server URL 인코딩 여부

Prototype

bool EncodingURL

Example

```
var dst = This.GetDataStore();  
_DEBUG("EncodingURL " + dst.EncodingURL);
```

SchemaURL

서버에서 운용 시 사용할 XSD 또는 DTD 파일 경로

Prototype

string SchemaURL

Example

```
var dst = This.GetDataStore();  
_DEBUG("SchemaURL " + dst.SchemaURL);
```

TestXMLURL

디자인 시 사용할 XML 파일의 로컬 시스템 파일 경로

Prototype

string TestXMLURL

Example

```
var dst = This.GetDataStore();  
_DEBUG("TestXMLURL " + dst.TestXMLURL);
```

XMLParser

XML 파서

Prototype

string XMLParser

Values

공백	서버에서 설정한 값
xerces	xerces
dom4j	dom4j

Reference

자바 서버에서만 동작합니다.

Example

```
var dst = This.GetDataStore();  
_DEBUG("XMLParser " + dst.XMLParser);
```

XMLURL

서버에서 운용 시 사용할 XML 파일 경로

Prototype

string XMLURL

Example

```
var dst = This.GetDataStore();  
_DEBUG("XMLURL " + dst.XMLURL);
```

Method

Name	Description
GetDataModule	데이터 모듈을 가져옵니다.
GetName	데이터 스토어 이름을 가져옵니다.

GetDataModule

데이터 모듈을 가져옵니다.

Prototype

object GetDataModule()

Example

```
var dst = This.GetDataStore();  
var DataModule = dst.GetDataModule();
```

GetName

데이터 스토어 이름을 가져옵니다.

Prototype

string GetName()

Example

```
var dst = This.GetDataStore();  
var Name = dst.GetName();
```

See Also

[이름](#)

Event

Name	Description
OnFinalize	바인딩 요청 시 데이터 스토어 삭제 직전에 발생합니다.
OnInitialize	바인딩 요청 시 데이터 스토어 초기화 직전에 발생합니다.
OnPostInitialize	바인딩 요청 시 데이터 스토어가 초기화된 직후에 발생합니다.

OnFinalize

바인딩 요청 시 데이터 스토어 삭제 직전에 발생합니다.

Prototype

void OnFinalize()

OnInitialize

바인딩 요청 시 데이터 스토어 초기화 직전에 발생합니다.

Prototype

`void OnInitialize()`

Reference

해당 이벤트에서 속성 값을 변경할 수 있습니다.

OnPostInitialize

바인딩 요청 시 데이터 스토어가 초기화된 직후에 발생합니다.

Prototype

`void OnPostInitialize()`

DataSet

- [OZCSVDataSet](#)
- [OZDBDataSet](#)
- [OZDBMultiDataSet](#)
- [OZDynamicDataSet](#)
- [OZGroupDataSet](#)
- [OZSAPDataSet](#)
- [OZUserDataSet](#)
- [OZXMLDataSet](#)
- [OZXPathDataSet](#)

OZCSVDataSet

- [Property](#)
- [Method](#)
- [Event](#)

Property

Name	Description
CharacterSet	문자 셋
ColumnDelimiter	컬럼 구분자
ConcurrentFetchSize	데이터 요청 시 한꺼번에 가져올 결과 데이터 셋의 크기
ConcurrentFirstRow	데이터 요청 시 처음 전송할 데이터의 행 수
ConnectionRetryCount	에러 발생 시 연결을 재시도하는 횟수
DoPost	URL의 패러미터 전송 시 포스트 방식으로 전송할지 여부
EncodingURL	Server URL 인코딩 유무
FlushDataAfterBinding	데이터 셋 바인딩 전에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부
FlushDataBeforeBinding	데이터 셋 바인딩 후에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부
HasFieldType	필드 타입 정보 포함 여부
Hidden	데이터 셋 숨김 여부
HTTPLogFileName	로그 파일 이름
MaxRowCount	최대 행 수
NullMark	데이터가 널인 경우 표시할 값
ParseErrorCode	에러 표시 코드
RowDelimiter	라인 구분자
URL	데이터를 가져올 파일 또는 사용자 애플리케이션의 HTTP 주소

CharacterSet

문자 셋

Prototype

`string CharacterSet`

Example

```
var ds = This.GetDataSet();  
_DEBUG("CharacterSet " + ds.CharacterSet);
```

See Also

[문자 셋\(파일 스토어 CSV 데이터 셋\)](#), [문자 셋\(HTTP 스토어 CSV 데이터 셋\)](#)

ColumnDelimiter

컬럼 구분자

Prototype

string ColumnDelimiter

Example

```
var ds = This.GetDataSet();  
_DEBUG("ColumnDelimiter " + ds.ColumnDelimiter);
```

See Also

[필드 구분자\(파일 스토어 CSV 데이터 셋\)](#), [필드 구분자\(HTTP 스토어 CSV 데이터 셋\)](#)

ConcurrentFetchSize

데이터 요청 시 한꺼번에 가져올 결과 데이터 셋의 크기

Prototype

integer ConcurrentFetchSize

Reference

fetch 타입이 Concurrent일 경우에만 해당 속성이 적용됩니다.

Example

```
var ds = This.GetDataSet();  
_DEBUG("ConcurrentFetchSize " + ds.ConcurrentFetchSize);
```

See Also

[Concurrent 전송 크기\(파일 스토어 CSV 데이터 셋\)](#), [Concurrent 전송 크기\(HTTP 스토어 CSV 데이터 셋\)](#)

ConcurrentFirstRow

데이터 요청 시 처음 전송할 데이터의 행 수

Prototype

integer ConcurrentFirstRow

Reference

fetch 타입이 Concurrent일 경우에만 해당 속성이 적용됩니다.

Example

```
var ds = This.GetDataSet();  
_DEBUG("ConcurrentFirstRow " + ds.ConcurrentFirstRow);
```

See Also

[Concurrent 첫 전송 Row\(파일 스토어 CSV 데이터 셋\)](#), [Concurrent 첫 전송 Row\(HTTP 스토어 CSV 데이터 셋\)](#)

ConnectionRetryCount

에러 발생 시 연결을 재시도하는 횟수

Prototype

integer ConnectionRetryCount

Example

```
var ds = This.GetDataSet();  
_DEBUG("ConnectionRetryCount " + ds.ConnectionRetryCount);
```

DoPost

URL의 패러미터 전송 시 포스트 방식으로 전송할지 여부

Prototype

bool DoPost

Reference

부모 스토어가 HTTP 스토어일 경우에만 적용됩니다.

Example

```
var ds = This.GetDataSet();  
_DEBUG("DoPost " + ds.DoPost);
```

EncodingURL

Server URL 인코딩 유무를 설정합니다.

Prototype

bool EncodingURL

Reference

부모 스토어가 HTTP 스토어일 경우에만 적용됩니다.

Example

```
var ds = This.GetDataSet();  
_DEBUG("EncodingURL " + ds.EncodingURL);
```

FlushDataAfterBinding

데이터 셋 바인딩 전에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부

Prototype

bool FlushDataAfterBinding

Example

```
var ds = This.GetDataSet();  
_DEBUG("FlushDataAfterBinding " + ds.FlushDataAfterBinding);
```

See Also

바인딩 후 데이터 보내기(파일 스토어 CSV 데이터 셋), 바인딩 후 데이터 보내기(HTTP 스토어 CSV 데이터 셋)

FlushDataBeforeBinding

데이터 셋 바인딩 후에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부

Prototype

bool FlushDataBeforeBinding

Example

```
var ds = This.GetDataSet();  
_DEBUG("FlushDataBeforeBinding " + ds.FlushDataBeforeBinding);
```

See Also

[바인딩 전 데이터 보내기\(파일 스토어 CSV 데이터 셋\)](#), [바인딩 전 데이터 보내기\(HTTP 스토어 CSV 데이터 셋\)](#)

HasFieldType

필드 타입 정보 포함 여부

Prototype

bool HasFieldType

Example

```
var ds = This.GetDataSet();  
_DEBUG("HasFieldType " + ds.HasFieldType);
```

Hidden

데이터 셋 숨김 여부

Prototype

bool Hidden

Example

```
var ds = This.GetDataSet();  
_DEBUG("Hidden " + ds.Hidden);
```

See Also

[숨김\(파일 스토어 CSV 데이터 셋\)](#), [숨김\(HTTP 스토어 CSV 데이터 셋\)](#)

HTTPLogFileName

로그 파일 이름

Prototype

string HTTPLogFileName

Reference

부모 스토어가 HTTP 스토어일 경우에만 적용됩니다.

Example

```
var ds = This.GetDataSet();  
_DEBUG("HTTPLogFileName " + ds.HTTPLogFileName);
```

MaxRowCount

최대 행 수

Prototype

integer MaxRowCount

Example

```
var ds = This.GetDataSet();  
_DEBUG("MaxRowCount " + ds.MaxRowCount);
```

See Also

[최대행 수\(파일 스토어 CSV 데이터 셋\)](#), [최대행 수\(HTTP 스토어 CSV 데이터 셋\)](#)

NullMark

데이터가 널인 경우 표시할 값

Prototype

string NullMark

Example

```
var ds = This.GetDataSet();  
_DEBUG("NullMark " + ds.NullMark);
```

See Also

[널 마크\(파일 스토어 CSV 데이터 셋\)](#), [널 마크\(HTTP 스토어 CSV 데이터 셋\)](#)

ParseErrorCode

에러 표시 코드

Prototype

string ParseErrorCode

Example

```
var ds = This.GetDataSet();  
_DEBUG("ParseErrorCode " + ds.ParseErrorCode);
```

RowDelimiter

라인 구분자

Prototype

string RowDelimiter

Example

```
var ds = This.GetDataSet();  
_DEBUG("RowDelimiter " + ds.RowDelimiter);
```

URL

데이터를 가져올 파일 또는 사용자 애플리케이션의 HTTP 주소

Prototype

string URL

Reference

부모 스토어가 HTTP, 파일 스토어일 경우에만 적용됩니다.

Example

```
var ds = This.GetDataSet();
_DEBUG("URL " + ds.URL);
```

Method

Name	Description
GetCalculatedFieldNames	해당 데이터셋에 추가된 모든 수식 필드 이름을 가져옵니다.
GetDataFieldNames	해당 데이터셋에 추가된 모든 데이터 필드 이름을 가져옵니다.
GetDataModule	데이터 모듈을 가져옵니다.
GetDataStore	데이터 스토어를 가져옵니다.
GetDetailDataSets	디테일셋 이름을 가져옵니다.
GetField	해당 이름의 필드 객체를 가져옵니다.
GetMasterDataSetName	마스터셋 이름을 가져옵니다.
GetName	데이터셋 이름을 가져옵니다.
IsFileStore	데이터 스토어가 파일 스토어인지 여부를 가져옵니다.
IsHTTPStore	데이터 스토어가 HTTP 스토어인지 여부를 가져옵니다.
IsMasterDataSet	해당 데이터셋이 마스터셋인지 여부를 가져옵니다.

GetCalculatedFieldNames

해당 데이터셋에 추가된 모든 수식 필드 이름을 가져옵니다.

Prototype

array [GetCalculatedFieldNames\(\)](#)

Example

```
var ds = This.GetDataSet();
```

```
var CalculatedFieldNames = new Array();  
CalculatedFieldNames = ds.GetCalculatedFieldNames();
```

GetDataFieldNames

해당 데이터셋에 추가된 모든 데이터 필드 이름을 가져옵니다.

Prototype

array [GetDataFieldNames\(\)](#)

Example

```
var ds = This.GetDataSet();  
var DataFieldNames = new Array();  
DataFieldNames = ds.GetDataFieldNames();
```

GetDataModule

데이터 모듈을 가져옵니다.

Prototype

object [GetDataModule\(\)](#)

Example

```
var ds = This.GetDataSet();  
var DataModule = ds.GetDataModule();
```

GetDataStore

데이터 스토어를 가져옵니다.

Prototype

object [GetDataStore\(\)](#)

Example

```
var ds = This.GetDataSet();  
var DataStore = ds.GetDataStore();
```

GetDetailDataSets

디테일셋 이름을 가져옵니다.

Prototype

array GetDetailDataSets()

Example

```
var ds = This.GetDataSet();  
var DetailDataSets = new Array();  
DetailDataSets = ds.GetDetailDataSets();
```

GetField

해당 이름의 필드 객체를 가져옵니다.

Prototype

object GetField(string FieldName)

Parameters

FieldName	필드 이름
-----------	-------

Reference

데이터 필드 또는 수식 필드 객체를 가져오며, 리턴되는 필드의 종류를 알기 위해서는 IsCalculatedField() 함수를 이용하시기 바랍니다.

Example

```
var ds = This.GetDataSet();  
var Field = ds.GetField("field1");
```

See Also

[IsCalculatedField\(OZCalculatedField\)](#), [IsCalculatedField\(OZDataField\)](#),
[IsCalculatedField\(OZXPathDataField\)](#)

GetMasterDataSetName

마스터셋 이름을 가져옵니다.

Prototype

string GetMasterDataSetName()

Reference

마스터셋이 없을 경우 공백 값을 가져옵니다.

Example

```
var ds = This.GetDataSet();  
var MasterDataSetName = ds.GetMasterDataSetName();
```

See Also

[마스터셋\(파일 스토어\)](#), [마스터셋\(XML 스토어\)](#)

GetName

데이터셋 이름을 가져옵니다.

Prototype

string GetName()

Example

```
var ds = This.GetDataSet();  
var Name = ds.GetName();
```

See Also

[이름\(파일 스토어\)](#), [이름\(XML 스토어\)](#)

IsFileStore

데이터 스토어가 파일 스토어인지 여부를 가져옵니다.

Prototype

bool IsFileStore()

Example

```
var ds = This.GetDataSet();
if(ds.IsFileStore() == true) {
    _DEBUG("Return true");
} else {
    _DEBUG("Return false");
}
```

IsHTTPStore

데이터 스토어가 HTTP 스토어인지 여부를 가져옵니다.

Prototype

bool IsHTTPStore()

Example

```
var ds = This.GetDataSet();
if(ds.IsHTTPStore() == true) {
    _DEBUG("Return true");
} else {
    _DEBUG("Return false");
}
```

IsMasterDataSet

해당 데이터셋이 마스터셋인지 여부를 가져옵니다.

Prototype

bool IsMasterDataSet()

Example

```
var ds = This.GetDataSet();
if(ds.IsMasterDataSet() == true) {
    _DEBUG("Return true");
} else {
    _DEBUG("Return false");
}
```

Event

Name	Description
OnFinalize	바인딩 요청 시 데이터 스토어 삭제 직전에 발생합니다.
OnInitialize	바인딩 요청 시 데이터 스토어 초기화 직전에 발생합니다.

OnFinalize

바인딩 요청 시 데이터 스토어 삭제 직전에 발생합니다.

Prototype

`void OnFinalize()`

OnInitialize

바인딩 요청 시 데이터 스토어 초기화 직전에 발생합니다.

Prototype

`void OnInitialize()`

Reference

해당 이벤트에서 속성 값을 변경할 수 있습니다.

OZDBDataSet

- [Property](#)
- [Method](#)
- [Event](#)

Property

Name	Description
ConcurrentFetchSize	데이터 요청 시 한꺼번에 가져올 결과 데이터 셋의 크기
ConcurrentFirstRow	데이터 요청 시 처음 전송할 데이터의 행 수

FlushDataAfterBinding	데이터 셋 바인딩 전에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부
FlushDataBeforeBinding	데이터 셋 바인딩 후에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부
Hidden	데이터 셋 숨김 여부
IgnoreQueryError	쿼리 에러 발생 시 빈 데이터 셋 전송 여부
JDBCFetchRow	JDBC 전송 ROW 즉, 결과 셋 반환 시 한꺼번에 가져올 수 있는 행 수
MaxRowCount	최대 행 수
QueryString	쿼리문
QueryTimeout	쿼리 실행 제한 시간
TargetParameter	실행된 프로시저의 결과 값을 가져올 패러미터 이름
UseCompiledSQL	컴파일된 질의문 사용 여부
UseOZScript	스크립트 사용 여부

ConcurrentFetchSize

데이터 요청 시 한꺼번에 가져올 결과 데이터 셋의 크기

Prototype

integer [ConcurrentFetchSize](#)

Reference

fetch 타입이 Concurrent일 경우에만 해당 속성이 적용됩니다.

Example

```
var ds = This.GetDataSet();
_DEBUG("ConcurrentFetchSize " + ds.ConcurrentFetchSize);
```

See Also

[Concurrent](#) 전송 크기

ConcurrentFirstRow

데이터 요청 시 처음 전송할 데이터의 행 수

Prototype

integer ConcurrentFirstRow

Reference

fetch 타입이 Concurrent일 경우에만 해당 속성이 적용됩니다.

Example

```
var ds = This.GetDataSet();  
_DEBUG("ConcurrentFirstRow " + ds.ConcurrentFirstRow);
```

See Also

[Concurrent 첫 전송 Row](#)

FlushDataAfterBinding

데이터 셋 바인딩 전에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부

Prototype

bool FlushDataAfterBinding

Example

```
var ds = This.GetDataSet();  
_DEBUG("FlushDataAfterBinding " + ds.FlushDataAfterBinding);
```

See Also

[바인딩 후 데이터 보내기](#)

FlushDataBeforeBinding

데이터 셋 바인딩 후에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부

Prototype

bool FlushDataBeforeBinding

Example

```
var ds = This.GetDataSet();  
_DEBUG("FlushDataBeforeBinding " + ds.FlushDataBeforeBinding);
```

See Also

[바인딩 전 데이터 보내기](#)

Hidden

데이터 셋 숨김 여부

Prototype

bool Hidden

Example

```
var ds = This.GetDataSet();  
_DEBUG("Hidden " + ds.Hidden);
```

See Also

[숨김](#)

IgnoreQueryError

쿼리 에러 발생 시 빈 데이터 셋 전송 여부

Prototype

bool IgnoreQueryError

Reference

IgnoreQueryError 속성 값을 true로 설정할 경우 쿼리 실행 시 에러가 발생하면 필드 정보만 있고 결과 셋은 없는 빈 데이터 셋을 전송합니다.

Example

```
var ds = This.GetDataSet();  
_DEBUG("IgnoreQueryError " + ds.IgnoreQueryError);
```

See Also

[쿼리 에러 무시](#)

JDBCFetchRow

JDBC 전송 ROW 즉, 결과 셋 반환 시 한꺼번에 가져올 수 있는 행 수

Prototype

integer JDBCFetchRow

Example

```
var ds = This.GetDataSet();  
_DEBUG("JDBCFetchRow " + ds.JDBCFetchRow);
```

MaxRowCount

최대행 수

Prototype

integer MaxRowCount

Example

```
var ds = This.GetDataSet();  
_DEBUG("MaxRowCount " + ds.MaxRowCount);
```

See Also

[최대행 수](#)

QueryString

쿼리문

Prototype

string QueryString

Reference

멀티 데이터 셋, 프로시저 데이터 셋을 실행하여 생성된 데이터 셋에서는 해당 속성이 적용되지 않습니다.

Example

```
var ds = This.GetDataSet();  
_DEBUG("QueryString " + ds.QueryString);
```

See Also

[질의문](#)

QueryTimeout

쿼리 실행 제한 시간

Prototype

[integer QueryTimeout](#)

Reference

0 ~ 2147483647 사이의 값으로 설정할 수 있습니다.

특정 쿼리의 실행 속도가 늦어져 세션이 홀딩되고 시스템의 과부하로 인해 다른 쿼리의 응답 속도도 느려지는 문제를 개선하려면 QueryTimeout 속성을 0보다 큰 값으로 적절히 설정하시기 바랍니다.

Example

```
var ds = This.GetDataSet();  
_DEBUG("QueryTimeout " + ds.QueryTimeout);
```

See Also

[쿼리 타임아웃](#)

TargetParameter

실행된 프로시저의 결과 값을 가져올 패러미터 이름

Prototype

[string TargetParameter](#)

Reference

프로시저 데이터 셋일 경우에만 적용됩니다.

패러미터의 데이터 타입이 Cursor일 경우에만 표시되며, 그 외 타입의 필드를 참조할 경우에는 빈 값이 표시됩니다.

Example

```
var ds = This.GetProcedureDataSet();  
_DEBUG("TargetParameter " + ds.TargetParameter);
```

See Also

[타겟 패러미터](#)

UseCompiledSQL

컴파일된 질의문 사용 여부

Prototype

bool UseCompiledSQL

Reference

멀티 데이터 셋, 프로시저 데이터 셋을 실행하여 생성된 데이터 셋에서는 해당 속성이 적용되지 않습니다.

Example

```
var ds = This.GetDataSet();  
_DEBUG("UseCompiledSQL " + ds.UseCompiledSQL);
```

See Also

[컴파일된 질의문 사용](#)

UseOZScript

스크립트 사용 여부

Prototype

bool UseOZScript

Reference

멀티 데이터 셋, 프로시저 데이터 셋을 실행하여 생성된 데이터 셋에서는 해당 속성이 적용되지 않습니다.

Example

```
var ds = This.GetDataSet();
_DEBUG("UseOZScript " + ds.UseOZScript);
```

See Also

[스크립트 사용](#)

Method

Name	Description
GetCalculatedFieldNames	해당 데이터셋에 추가된 모든 수식 필드 이름을 가져옵니다
GetDataFieldNames	해당 데이터셋에 추가된 모든 데이터 필드 이름을 가져옵니다.
GetDataModule	데이터 모듈을 가져옵니다.
GetDataStore	데이터 스토어를 가져옵니다.
GetDestinationFields	Destination 필드 이름을 가져옵니다.
GetDestinationValue	Destination 필드 값을 가져옵니다.
GetDestinationValues	Destination 필드 값을 가져옵니다.
GetDetailDataSets	디테일셋 이름을 가져옵니다.
GetDrivenDataSetIndex	멀티 데이터 셋 또는 프로시저 데이터 셋을 실행하여 생성된 데이터 셋의 현재 인덱스를 가져옵니다.
GetField	해당 이름의 필드 객체를 가져옵니다.
GetMasterDataSetName	마스터셋 이름을 가져옵니다.
GetMultiDataSet	멀티 데이터 셋을 가져옵니다.
GetName	데이터셋 이름을 가져옵니다.
GetProcedureDataSet	프로시저 데이터 셋을 가져옵니다.
GetSourceFields	Source 필드 이름을 가져옵니다.
GetSourceValue	Source 필드 값을 가져옵니다.

GetSourceValues	Source 필드 값을 가져옵니다.
IsMasterDataSet	해당 데이터셋이 마스터셋인지 여부를 가져옵니다.

GetCalculatedFieldNames

해당 데이터셋에 추가된 모든 수식 필드 이름을 가져옵니다.

Prototype

array [GetCalculatedFieldNames\(\)](#)

Example

```
var ds = This.GetDataSet();
var CalculatedFieldNames = new Array();
CalculatedFieldNames = ds.GetCalculatedFieldNames();
```

GetDataFieldNames

해당 데이터셋에 추가된 모든 데이터 필드 이름을 가져옵니다.

Prototype

array [GetDataFieldNames\(\)](#)

Example

```
var ds = This.GetDataSet();
var DataFieldNames = new Array();
DataFieldNames = ds.GetDataFieldNames();
```

GetDataModule

데이터 모듈을 가져옵니다.

Prototype

object [GetDataModule\(\)](#)

Example

```
var ds = This.GetDataSet();
```

```
var DataModule = ds.GetDataModule();
```

GetDestinationFields

Destination 필드 이름을 가져옵니다.

Prototype

array GetDestinationFields()

Reference

OnInsert, OnUpdate, OnDelete 이벤트에서 실행한 경우에만 동작하며, 그 외 이벤트에서 실행하였을 경우 null을 리턴합니다.

Example

```
var ds = This.GetDataSet();
var DestinationFields = new Array();
DestinationFields = ds.GetDestinationFields();
```

GetDestinationValue

Destination 필드 값을 가져옵니다.

Prototype

var GetDestinationValue(string FieldName)

Parameters

FieldName	Destination 필드 값을 가져올 필드 이름
-----------	-----------------------------

Reference

OnInsert, OnUpdate, OnDelete 이벤트에서 실행한 경우에만 동작하며, 그 외 이벤트에서 실행하였을 경우 null을 리턴합니다.

리턴되는 필드 값은 필드 타입에 따라 문자, 숫자, bool, Date 객체, [OZBinary](#) 객체로 리턴됩니다.

필드 타입에 따라 리턴되는 값의 타입은 아래 표를 참고하시기 바랍니다.

Name	Description
TINYINT	숫자 값

OZSMALLINT	숫자 값
INTEGER	숫자 값
BIGINT	숫자 값
REAL	숫자 값
FLOAT	숫자 값
DOUBLE	숫자 값
BIT	true, false
CHAR	문자 값
VARCHAR	문자 값
LONGVARCHAR	문자 값
NUMERIC	숫자 값
DECIMAL	숫자 값
DATE	자바 스크립트 Date 객체
TIME	자바 스크립트 Date 객체
TIMESTAMP	자바 스크립트 Date 객체
BINARY	OZBinary 객체
VARBINARY	OZBinary 객체
LONGBINARY	OZBinary 객체
BLOB	OZBinary 객체
CLOB	문자 값

Example

```

var sourceNames = This.GetSourceFields();
PrintArray(This, sourceNames, "Source names ");
PrintArray(This, This.GetSourceValues(), "Source values ");

var destNames = This.GetDestinationFields();
PrintArray(This, destNames, "Destination names ");
PrintArray(This, This.GetDestinationValues(), "Destination values ");

```

```
if(destNames != undefined) {  
    for(var i = 0; i < destNames.length; i++) {  
        var value = This.GetDestinationValue(destNames[i]);  
        _DEBUG("Value of '" + destNames[i] + "' is " + value);  
    }  
}  
  
return "Success";
```

GetDestinationValues

Destination 필드 값을 가져옵니다.

Prototype

array [GetDestinationValues\(\)](#)

Reference

OnInsert, OnUpdate, OnDelete 이벤트에서 실행한 경우에만 동작하며, 그 외 이벤트에서 실행하였을 경우 null을 리턴합니다.

리턴되는 필드 값은 필드 타입에 따라 문자, 숫자, bool, Date 객체, [OZBinary](#) 객체로 리턴됩니다.

필드 타입에 따라 리턴되는 값의 타입은 [GetDestinationValue\(\)](#) 함수를 참고하시기 바랍니다.

Example

```
var ds = This.GetDataSet();  
var DestinationValues = new Array();  
DestinationValues = ds.GetDestinationValues();
```

GetDataStore

데이터 스토어를 가져옵니다.

Prototype

object [GetDataStore\(\)](#)

Example

```
var ds = This.GetDataSet();
```

```
var DataStore = ds.GetDataStore();
```

GetDetailDataSets

디테일셋 이름을 가져옵니다.

Prototype

array GetDetailDataSets()

Example

```
var ds = This.GetDataSet();  
var DetailDataSets = new Array();  
DetailDataSets = ds.GetDetailDataSets();
```

GetDrivenDataSetIndex

멀티 데이터 셋 또는 프로시저 데이터 셋을 실행하여 생성된 데이터 셋의 현재 인덱스를 가져옵니다.

Prototype

integer GetDrivenDataSetIndex()

Reference

멀티 데이터 셋 또는 프로시저 데이터 셋을 실행하여 생성된 데이터 셋일 경우에만 적용됩니다.

Example

```
var ds = This.GetDataSet();  
DrivenDataSetIndex = ds.GetDrivenDataSetIndex();
```

GetField

해당 이름의 필드 객체를 가져옵니다.

Prototype

object GetField(string FieldName)

Parameters

FieldName	필드 이름
-----------	-------

Reference

데이터 필드 또는 수식 필드 객체를 가져오며, 리턴되는 필드의 종류를 알기 위해서는 IsCalculatedField() 함수를 이용하시기 바랍니다.

Example

```
var ds = This.GetDataSet();  
var Field = ds.GetField("field1");
```

See Also

[IsCalculatedField\(OZCalculatedField\)](#), [IsCalculatedField\(OZDataField\)](#),
[IsCalculatedField\(OZXPathDataField\)](#)

GetName

데이터셋 이름을 가져옵니다.

Prototype

string GetName()

Example

```
var ds = This.GetDataSet();  
var Name = ds.GetName();
```

GetMultiDataSet

멀티 데이터 셋을 가져옵니다.

Prototype

object GetMultiDataSet()

Reference

멀티 데이터 셋을 실행하여 생성된 데이터 셋일 경우에만 적용됩니다.

Example

```
var ds = This.GetDataSet();  
MultiDataSet = ds.GetMultiDataSet();
```

GetMasterDataSetName

마스터셋 이름을 가져옵니다.

Prototype

string GetMasterDataSetName()

Reference

마스터셋이 없을 경우 공백 값을 가져옵니다.

Example

```
var ds = This.GetDataSet();  
var MasterDataSetName = ds.GetMasterDataSetName();
```

See Also

[마스터셋](#)

GetProcedureDataSet

프로시저 데이터 셋을 가져옵니다.

Prototype

object GetProcedureDataSet()

Reference

프로시저 데이터 셋을 실행하여 생성된 데이터 셋일 경우에만 적용됩니다.

Example

```
var ds = This.GetDataSet();  
ProcedureDataSet = ds.GetProcedureDataSet();
```

GetSourceFields

Source 필드 이름을 가져옵니다.

Prototype

array [GetSourceFields\(\)](#)

Reference

OnInsert, OnUpdate, OnDelete 이벤트에서 실행한 경우에만 동작하며, 그 외 이벤트에서 실행하였을 경우 null을 리턴합니다.

Example

```
var ds = This.GetDataSet();  
var SourceFields = new Array();  
SourceFields = ds.GetSourceFields();
```

GetSourceValue

Source 필드 값을 가져옵니다.

Prototype

var [GetSourceValue\(string FieldName\)](#)

Parameters

FieldName	Source 필드 값을 가져올 필드 이름
-----------	------------------------

Reference

OnInsert, OnUpdate, OnDelete 이벤트에서 실행한 경우에만 동작하며, 그 외 이벤트에서 실행하였을 경우 null을 리턴합니다.

리턴되는 필드 값은 필드 타입에 따라 문자, 숫자, bool, Date 객체, [OZBinary](#) 객체로 리턴됩니다.

필드 타입에 따라 리턴되는 값의 타입은 아래 표를 참고하시기 바랍니다.

Name	Description
TINYINT	숫자 값
OZSMALLINT	숫자 값
INTEGER	숫자 값
BIGINT	숫자 값

REAL	숫자 값
FLOAT	숫자 값
DOUBLE	숫자 값
BIT	true, false
CHAR	문자 값
VARCHAR	문자 값
LONGVARCHAR	문자 값
NUMERIC	숫자 값
DECIMAL	숫자 값
DATE	자바 스크립트 Date 객체
TIME	자바 스크립트 Date 객체
TIMESTAMP	자바 스크립트 Date 객체
BINARY	OZBinary 객체
VARBINARY	OZBinary 객체
LONGBINARY	OZBinary 객체
BLOB	OZBinary 객체
CLOB	문자 값

Example

```

var sourceNames = This.GetSourceFields();
PrintArray(This, sourceNames, "Source names ");
PrintArray(This, This.GetSourceValues(), "Source values ");

var destNames = This.GetDestinationFields();
PrintArray(This, destNames, "Destination names ");
PrintArray(This, This.GetDestinationValues(), "Destination values ");

if(sourceNames != undefined) {
    for(var i = 0; i < sourceNames.length; i++) {
        var value = This.GetSourceValue(sourceNames[i]);
        _DEBUG("Value of '" + sourceNames[i] + "' is " + value);
    }
}

```

```
    }  
}  
  
return "Success";
```

GetSourceValues

Source 필드 값을 가져옵니다.

Prototype

array [GetSourceValues\(\)](#)

Reference

OnInsert, OnUpdate, OnDelete 이벤트에서 실행한 경우에만 동작하며, 그 외 이벤트에서 실행하였을 경우 null을 리턴합니다.

리턴되는 필드 값은 필드 타입에 따라 문자, 숫자, bool, Date 객체, [OZBinary](#) 객체로 리턴됩니다.

필드 타입에 따라 리턴되는 값의 타입은 [GetSourceValue\(\)](#) 함수를 참고하시기 바랍니다.

Example

```
var ds = This.GetDataSet();  
var SourceValues = new Array();  
SourceValues = ds.GetSourceValues();
```

IsMasterDataSet

해당 데이터셋이 마스터셋인지 여부를 가져옵니다.

Prototype

bool [IsMasterDataSet\(\)](#)

Example

```
var ds = This.GetDataSet();  
if(ds.IsMasterDataSet() == true) {  
    _DEBUG("Return true");  
} else {  
    _DEBUG("Return false");  
}
```

Event

Name	Description
OnFinalize	바인딩 요청 시 데이터 스토어 삭제 직전에 발생합니다.
OnInitialize	바인딩 요청 시 데이터 스토어 초기화 직전에 발생합니다.
OnPostBind	바인딩 실행 후에 발생합니다.
OnPreBind	바인딩 실행 직전에 발생합니다.

OnFinalize

바인딩 요청 시 데이터 스토어 삭제 직전에 발생합니다.

Prototype

`void OnFinalize()`

OnInitialize

바인딩 요청 시 데이터 스토어 초기화 직전에 발생합니다.

Prototype

`void OnInitialize()`

Reference

해당 이벤트에서 속성 값을 변경할 수 있습니다.

OnPostBind

바인딩 실행 후에 발생합니다.

Prototype

`void OnPostBind()`

OnPreBind

바인딩 실행 직전에 발생합니다.

Prototype

void OnPreBind()

Reference

동적으로 쿼리를 변경하거나 자체 쿼리 실행으로 결과셋 생성이 가능합니다.

디테일셋일 경우 마스터셋의 결과셋 행 수만큼 이벤트가 발생합니다.

OZJSResultSet 객체를 가져올 수 있습니다.

null 또는 OZJSResultSet 이외의 타입을 리턴하며 리턴이 없을 경우 서버의 결과셋 생성 로직을 수행합니다

.

OZDBMultiDataSet

- [Property](#)
- [Method](#)
- [Event](#)

Property

Name	Description
QueryString	쿼리문
QueryTimeout	쿼리 실행 제한 시간
UseCompiledSQL	컴파일된 질의문 사용 여부
UseOZScript	스크립트 사용 여부

QueryString

쿼리문

Prototype

string QueryString

Example

```
var ds = This.GetMultiDataSet();  
_DEBUG("QueryString " + ds.QueryString);
```

See Also

[질의문](#)

QueryTimeout

쿼리 실행 제한 시간

Prototype

integer QueryTimeout

Reference

0 ~ 2147483647 사이의 값으로 설정할 수 있습니다.

특정 쿼리의 실행 속도가 늦어져 세션이 홀딩되고 시스템의 과부하로 인해 다른 쿼리의 응답 속도도 느려지는 문제를 개선하려면 QueryTimeout 속성을 0보다 큰 값으로 적절히 설정하시기 바랍니다.

Example

```
var ds = This.GetDataSet();  
_DEBUG("QueryTimeout " + ds.QueryTimeout);
```

See Also

[쿼리 타임아웃](#)

UseCompiledSQL

컴파일된 질의문 사용 여부

Prototype

bool UseCompiledSQL

Example

```
var ds = This.GetMultiDataSet();  
_DEBUG("UseCompiledSQL " + ds.UseCompiledSQL);
```

See Also

[컴파일된 질의문 사용](#)

UseOZScript

스크립트 사용 여부

Prototype

bool UseOZScript

Example

```
var ds = This.GetMultiDataSet();  
_DEBUG("UseOZScript " + ds.UseOZScript);
```

See Also

[스크립트 사용](#)

Method

Name	Description
GetDataModule	데이터 모듈을 가져옵니다.
GetDataStore	데이터 스토어를 가져옵니다.
GetDrivenDataSets	멀티 데이터 셋을 실행하여 생성된 데이터 셋을 가져옵니다.
GetMasterDataSetName	마스터셋 이름을 가져옵니다.
GetName	데이터셋 이름을 가져옵니다.

GetDataModule

데이터 모듈을 가져옵니다.

Prototype

object GetDataModule()

Example

```
var ds = This.GetMultiDataSet();  
var DataModule = ds.GetDataModule();
```

GetDataStore

데이터 스토어를 가져옵니다.

Prototype

object GetDataStore()

Example

```
var ds = This.GetMultiDataSet();  
var DataStore = ds.GetDataStore();
```

GetDrivenDataSets

멀티 데이터 셋을 실행하여 생성된 데이터 셋으로 OZDBDataSet 객체를 배열로 가져옵니다.

Prototype

array GetDrivenDataSets()

Example

```
var ds = This.GetMultiDataSet();  
var DrivenDataSets = new Array();  
DrivenDataSets = ds.GetDrivenDataSets();
```

GetMasterDataSetName

마스터셋 이름을 가져옵니다.

Prototype

string GetMasterDataSetName()

Reference

마스터셋이 없을 경우 공백 값을 가져옵니다.

Example

```
var ds = This.GetMultiDataSet();  
var MasterDataSetName = ds.GetMasterDataSetName();
```

See Also

[마스터셋](#)

GetName

데이터셋 이름을 가져옵니다.

Prototype

string GetName()

Example

```
var ds = This.GetMultiDataSet();  
var Name = ds.GetName();
```

Event

Name	Description
OnFinalize	바인딩 요청 시 데이터 스토어 삭제 직전에 발생합니다.
OnInitialize	바인딩 요청 시 데이터 스토어 초기화 직전에 발생합니다.
OnPostBind	바인딩 실행 후에 발생합니다.
OnPreBind	바인딩 실행 직전에 발생합니다.

OnFinalize

바인딩 요청 시 데이터 스토어 삭제 직전에 발생합니다.

Prototype

void OnFinalize()

OnInitialize

바인딩 요청 시 데이터 스토어 초기화 직전에 발생합니다.

Prototype

`void OnInitialize()`

Reference

해당 이벤트에서 속성 값을 변경할 수 있습니다.

OnPostBind

바인딩 실행 후에 발생합니다.

Prototype

`void OnPostBind()`

OnPreBind

바인딩 실행 직전에 발생합니다.

Prototype

`void OnPreBind()`

Reference

동적으로 쿼리를 변경하거나 자체 쿼리 실행으로 결과셋 생성이 가능합니다.

디테일셋일 경우 마스터셋의 결과셋 행 수만큼 이벤트가 발생합니다.

OZJSResultSet 객체를 가져올 수 있습니다.

null 또는 OZJSResultSet 이외의 타입을 리턴하며 리턴이 없을 경우 서버의 결과셋 생성 로직을 수행합니다

.

OZDBProcedureDataSet

- [Property](#)
- [Method](#)
- [Event](#)

Property

Name	Description
ProcedureName	프로시저 이름
ProcedureType	프로시저 타입
QueryTimeout	쿼리 실행 제한 시간

ProcedureName

프로시저 이름

Prototype

string [ProcedureName](#)

Example

```
var ds = This.GetProcedureDataSet();  
_DEBUG("ProcedureName " + ds.ProcedureName);
```

See Also

[프로시저 이름](#)

ProcedureType

프로시저 타입

Prototype

integer [ProcedureType](#)

Example

```
var ds = This.GetProcedureDataSet();  
_DEBUG("ProcedureType " + ds.ProcedureType);
```

See Also

[프로시저 타입](#), [ProcedureTypeConst](#)

QueryTimeout

쿼리 실행 제한 시간

Prototype

integer QueryTimeout

Reference

0 ~ 2147483647 사이의 값으로 설정할 수 있습니다.

특정 쿼리의 실행 속도가 늦어져 세션이 홀딩되고 시스템의 과부하로 인해 다른 쿼리의 응답 속도도 느려지는 문제를 개선하려면 QueryTimeout 속성을 0보다 큰 값으로 적절히 설정하시기 바랍니다.

Example

```
var ds = This.GetDataSet();
_DEBUG("QueryTimeout " + ds.QueryTimeout);
```

See Also

[쿼리 타임아웃](#)

Method

Name	Description
AddProcedureParameter	프로시저 패러미터를 추가합니다.
GetDataModule	데이터 모듈을 가져옵니다.
GetDataStore	데이터 스토어를 가져옵니다.
GetDrivenDataSets	멀티 데이터 셋을 실행하여 생성된 데이터 셋을 가져옵니다.
GetMasterDataSetName	마스터셋 이름을 가져옵니다.
GetName	데이터셋 이름을 가져옵니다.
GetProcedureParameterNames	추가된 프로시저 패러미터 이름을 모두 가져옵니다.
GetProcedureParameter	프로시저 패러미터를 가져옵니다.
SetProcedureParameter	프로시저 패러미터 값을 설정합니다.
RemoveProcedureParameter	프로시저 패러미터를 삭제합니다.
RemoveAllProcedureParameter	프로시저 패러미터를 모두 삭제합니다.

AddProcedureParameter

프로시저 패러미터를 추가합니다.

Prototype

void AddProcedureParameter(object ProcedureParameter)

Parameters

ProcedureParameter	추가할 프로시저 패러미터 객체
--------------------	------------------

Example

```
var ProcedureParamNames = This.GetProcedureParameterNames();
var params = new Array();
for(var i=0; i< ProcedureParamNames.length; i++) {
    This.AddProcedureParameter(params[i]);
}
```

See Also

[OZProcedureParameter](#)

GetDataModule

데이터 모듈을 가져옵니다.

Prototype

object GetDataModule()

Example

```
var ds = This.GetProcedureDataSet();
var DataModule = ds.GetDataModule();
```

GetDataStore

데이터 스토어를 가져옵니다.

Prototype

object GetDataStore()

Example

```
var ds = This.GetProcedureDataSet();  
var DataStore = ds.GetDataStore();
```

GetDrivenDataSets

프로시저 데이터 셋을 실행하여 생성된 데이터 셋으로 OZDBDataSet 객체를 배열로 가져옵니다.

Prototype

array GetDrivenDataSets()

Example

```
var ds = This.GetProcedureDataSet();  
var DrivenDataSets = new Array();  
DrivenDataSets = ds.GetDrivenDataSets();
```

GetMasterDataSetName

마스터셋 이름을 가져옵니다.

Prototype

string GetMasterDataSetName()

Reference

마스터셋이 없을 경우 공백 값을 가져옵니다.

Example

```
var ds = This.GetProcedureDataSet();  
var MasterDataSetName = ds.GetMasterDataSetName();
```

See Also

[마스터셋](#)

GetName

데이터셋 이름을 가져옵니다.

Prototype

string GetName()

Example

```
var ds = This.GetProcedureDataSet();  
var Name = ds.GetName();
```

GetProcedureParameter

프로시저 패러미터를 가져옵니다.

Prototype

object GetProcedureParameter(string ParameterName)

Parameters

ParameterName	프로시저 패러미터 이름
---------------	--------------

Example

```
var ProcedureParamNames = This.GetProcedureParameterNames();  
var params = new Array();  
for(var i=0; i< ProcedureParamNames.length; i++) {  
    params[i] = This.GetProcedureParameter(ProcedureParamNames[i]);  
}
```

GetProcedureParameterNames

추가된 프로시저 패러미터 이름을 모두 가져옵니다.

Prototype

array GetProcedureParameterNames()

Example

```
var ds = This.GetProcedureDataSet();  
var ProcedureParamNames = new Array();
```

```
ProcedureParamNames = ds.GetProcedureParameterNames();
```

SetProcedureParameter

프로시저 패러미터 값을 설정합니다.

Prototype

```
void SetProcedureParameter(integer ParameterIndex, object ProcedureParameter)
```

Parameters

ParameterIndex	프로시저 패러미터 인덱스 (One-Based Index)
ProcedureParameter	값을 설정할 패러미터 객체

Example

```
var ds = This.GetProcedureDataSet();
var ProcedureParameter = This.GetProcedureParameter("param1");
ds.SetProcedureParameter(1, ProcedureParameter);
```

RemoveProcedureParameter

프로시저 패러미터를 삭제합니다.

Prototype

```
void RemoveProcedureParameter(string ParameterName)
```

Parameters

ParameterName	삭제할 프로시저 패러미터 이름
---------------	------------------

Example

```
var ds = This.GetProcedureDataSet();
ds.RemoveProcedureParameter("param1");
```

RemoveAllProcedureParameter

프로시저 패러미터를 모두 삭제합니다.

Prototype

void RemoveAllProcedureParameter()

Example

```
var ds = This.GetProcedureDataSet();  
var Name = ds.RemoveAllProcedureParameter();
```

Event

Name	Description
OnFinalize	바인딩 요청 시 데이터 스토어 삭제 직전에 발생합니다.
OnInitialize	바인딩 요청 시 데이터 스토어 초기화 직전에 발생합니다.
OnPostBind	바인딩 실행 후에 발생합니다.
OnPreBind	바인딩 실행 직전에 발생합니다.

OnFinalize

바인딩 요청 시 데이터 스토어 삭제 직전에 발생합니다.

Prototype

void OnFinalize()

OnInitialize

바인딩 요청 시 데이터 스토어 초기화 직전에 발생합니다.

Prototype

void OnInitialize()

Reference

해당 이벤트에서 속성 값을 변경할 수 있습니다.

OnPostBind

바인딩 실행 후에 발생합니다.

Prototype

`void OnPostBind()`

OnPreBind

바인딩 실행 직전에 발생합니다.

Prototype

`void OnPreBind()`

Reference

동적으로 쿼리를 변경하거나 자체 쿼리 실행으로 결과셋 생성이 가능합니다.

디테일셋일 경우 마스터셋의 결과셋 행 수만큼 이벤트가 발생합니다.

OZJSResultSet 객체를 가져올 수 있습니다.

null 또는 OZJSResultSet 이외의 타입을 리턴하며 리턴이 없을 경우 서버의 결과셋 생성 로직을 수행합니다

.

OZDynamicDataSet

- [Property](#)
- [Method](#)
- [Event](#)

Property

Name	Description
ConcurrentFetchSize	데이터 요청 시 한꺼번에 가져올 결과 데이터 셋의 크기
ConcurrentFirstRow	데이터 요청 시 처음 전송할 데이터의 행 수
FlushDataAfterBinding	데이터 셋 바인딩 전에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부
FlushDataBeforeBinding	데이터 셋 바인딩 후에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부

Hidden	데이터 셋 숨김 여부
IgnoreQueryError	쿼리 에러 발생 시 빈 데이터 셋 전송 여부
MaxRowCount	최대 행 수
QueryTimeout	쿼리 실행 제한 시간
SourceDataSetName	원본 셋 이름
StartRowIndex	시작 행 번호

ConcurrentFetchSize

데이터 요청 시 한꺼번에 가져올 결과 데이터 셋의 크기

Prototype

integer ConcurrentFetchSize

Reference

fetch 타입이 Concurrent일 경우에만 해당 속성이 적용됩니다.

Example

```
var ds = This.GetDataSet();
_DEBUG("ConcurrentFetchSize " + ds.ConcurrentFetchSize);
```

ConcurrentFirstRow

데이터 요청 시 처음 전송할 데이터의 행 수

Prototype

integer ConcurrentFirstRow

Reference

fetch 타입이 Concurrent일 경우에만 해당 속성이 적용됩니다.

Example

```
var ds = This.GetDataSet();
_DEBUG("ConcurrentFirstRow " + ds.ConcurrentFirstRow);
```

FlushDataAfterBinding

데이터 셋 바인딩 전에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부

Prototype

bool FlushDataAfterBinding

Example

```
var ds = This.GetDataSet();  
_DEBUG("FlushDataAfterBinding " + ds.FlushDataAfterBinding);
```

FlushDataBeforeBinding

데이터 셋 바인딩 후에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부

Prototype

bool FlushDataBeforeBinding

Example

```
var ds = This.GetDataSet();  
_DEBUG("FlushDataBeforeBinding " + ds.FlushDataBeforeBinding);
```

Hidden

데이터 셋 숨김 여부

Prototype

bool Hidden

Example

```
var ds = This.GetDataSet();  
_DEBUG("Hidden " + ds.Hidden);
```

IgnoreQueryError

쿼리 에러 발생 시 빈 데이터 셋 전송 여부

Prototype

bool IgnoreQueryError

Reference

값을 true로 설정할 경우 쿼리 실행 시 에러가 발생하면 필드 정보만 있고 결과 셋은 없는 빈 데이터 셋을 전송합니다.

Example

```
var ds = This.GetDataSet();  
_DEBUG("IgnoreQueryError " + ds.IgnoreQueryError);
```

See Also

[쿼리 에러 무시](#)

MaxRowCount

최대 행 수

Prototype

integer MaxRowCount

Example

```
var ds = This.GetDataSet();  
_DEBUG("MaxRowCount " + ds.MaxRowCount);
```

QueryTimeout

쿼리 실행 제한 시간

Prototype

integer QueryTimeout

Reference

0 ~ 2147483647 사이의 값으로 설정할 수 있습니다.

특정 쿼리의 실행 속도가 늦어져 세션이 홀딩되고 시스템의 과부하로 인해 다른 쿼리의 응답 속도도 느려지는 문제를 개선하려면 QueryTimeout 속성을 0보다 큰 값으로 적절히 설정하시기 바랍니다.

Example

```
var ds = This.GetDataSet();  
_DEBUG("QueryTimeout " + ds.QueryTimeout);
```

See Also

[쿼리 타임아웃](#)

SourceDataSetName

원본 셋 이름

Prototype

string SourceDataSetName

Example

```
var ds = This.GetDataSet();  
_DEBUG("SourceDataSetName " + ds.SourceDataSetName);
```

StartRowIndex

시작 행 번호

Prototype

integer StartRowIndex

Example

```
var ds = This.GetDataSet();  
_DEBUG("StartRowIndex " + ds.StartRowIndex);
```

Method

Name	Description
GetCalculatedFieldNames	해당 데이터셋에 추가된 모든 수식 필드 이름을 가져옵니다

GetDataFieldNames	해당 데이터셋에 추가된 모든 데이터 필드 이름을 가져옵니다.
GetDataModule	데이터 모듈을 가져옵니다.
GetDataStore	데이터 스토어를 가져옵니다.
GetDetailDataSets	디테일셋 이름을 가져옵니다.
GetField	해당 이름의 필드 객체를 가져옵니다.
GetMasterDataSetName	마스터셋 이름을 가져옵니다.
GetName	데이터셋 이름을 가져옵니다.
IsMasterDataSet	해당 데이터셋이 마스터셋인지 여부를 가져옵니다.

GetCalculatedFieldNames

해당 데이터셋에 추가된 모든 수식 필드 이름을 가져옵니다.

Prototype

array [GetCalculatedFieldNames\(\)](#)

Example

```
var ds = This.GetDataSet();
var CalculatedFieldNames = new Array();
CalculatedFieldNames = ds.GetCalculatedFieldNames();
```

GetDataFieldNames

해당 데이터셋에 추가된 모든 데이터 필드 이름을 가져옵니다.

Prototype

array [GetDataFieldNames\(\)](#)

Example

```
var ds = This.GetDataSet();
var DataFieldNames = new Array();
DataFieldNames = ds.GetDataFieldNames();
```

GetDataModule

데이터 모듈을 가져옵니다.

Prototype

object GetDataModule()

Example

```
var ds = This.GetDataSet();  
var DataModule = ds.GetDataModule();
```

GetDataStore

데이터 스토어를 가져옵니다.

Prototype

object GetDataStore()

Example

```
var ds = This.GetDataSet();  
var DataStore = ds.GetDataStore();
```

GetDetailDataSets

디테일셋 이름을 가져옵니다.

Prototype

array GetDetailDataSets()

Example

```
var ds = This.GetDataSet();  
var DetailDataSets = new Array();  
DetailDataSets = ds.GetDetailDataSets();
```

GetField

해당 이름의 필드 객체를 가져옵니다.

Prototype

object GetField(string FieldName)

Parameters

FieldName	필드 이름
-----------	-------

Reference

데이터 필드 또는 수식 필드 객체를 가져오며, 리턴되는 필드의 종류를 알기 위해서는 IsCalculatedField() 함수를 이용하시기 바랍니다.

Example

```
var ds = This.GetDataSet();  
var Field = ds.GetField("field1");
```

See Also

[IsCalculatedField\(OZCalculatedField\)](#), [IsCalculatedField\(OZDataField\)](#),
[IsCalculatedField\(OZXPathDataField\)](#)

GetMasterDataSetName

마스터셋 이름을 가져옵니다.

Prototype

string GetMasterDataSetName()

Reference

마스터셋이 없을 경우 공백 값을 가져옵니다.

Example

```
var ds = This.GetDataSet();  
var MasterDataSetName = ds.GetMasterDataSetName();
```

GetName

데이터셋 이름을 가져옵니다.

Prototype

string GetName()

Example

```
var ds = This.GetDataSet();  
var Name = ds.GetName();
```

IsMasterDataSet

해당 데이터셋이 마스터셋인지 여부를 가져옵니다.

Prototype

bool IsMasterDataSet()

Example

```
var ds = This.GetDataSet();  
if(ds.IsMasterDataSet() == true) {  
    _DEBUG("Return true");  
} else {  
    _DEBUG("Return false");  
}
```

Event

Name	Description
OnFinalize	바인딩 요청 시 데이터 스토어 삭제 직전에 발생합니다.
OnInitialize	바인딩 요청 시 데이터 스토어 초기화 직전에 발생합니다.

OnFinalize

바인딩 요청 시 데이터 스토어 삭제 직전에 발생합니다.

Prototype

void OnFinalize()

OnInitialize

바인딩 요청 시 데이터 스토어 초기화 직전에 발생합니다.

Prototype

`void OnInitialize()`

Reference

해당 이벤트에서 속성 값을 변경할 수 있습니다.

OZGroupDataSet

- [Property](#)
- [Method](#)
- [Event](#)

Property

Name	Description
Command	실행문
ConcurrentFetchSize	데이터 요청 시 한꺼번에 가져올 결과 데이터 셋의 크기
ConcurrentFirstRow	데이터 요청 시 처음 전송할 데이터의 행 수
FlushDataAfterBinding	데이터 셋 바인딩 전에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부
FlushDataBeforeBinding	데이터 셋 바인딩 후에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부
Hidden	데이터 셋 숨김 여부
MaxRowCount	최대 행 수

Command

실행문

Prototype

`string Command`

Example

```
var ds = This.GetDataSet();  
_DEBUG("Command " + ds.Command);
```

See Also

[실행문](#)

ConcurrentFetchSize

데이터 요청 시 한꺼번에 가져올 결과 데이터 셋의 크기

Prototype

integer ConcurrentFetchSize

Reference

fetch 타입이 Concurrent일 경우에만 해당 속성이 적용됩니다.

Example

```
var ds = This.GetDataSet();  
_DEBUG("ConcurrentFetchSize " + ds.ConcurrentFetchSize);
```

See Also

[Concurrent 전송 크기](#)

ConcurrentFirstRow

데이터 요청 시 처음 전송할 데이터의 행 수

Prototype

integer ConcurrentFirstRow

Reference

fetch 타입이 Concurrent일 경우에만 해당 속성이 적용됩니다.

Example

```
var ds = This.GetDataSet();  
_DEBUG("ConcurrentFirstRow " + ds.ConcurrentFirstRow);
```

See Also

[Concurrent 첫 전송 Row](#)

FlushDataAfterBinding

데이터 셋 바인딩 전에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부

Prototype

bool FlushDataAfterBinding

Example

```
var ds = This.GetDataSet();  
_DEBUG("FlushDataAfterBinding " + ds.FlushDataAfterBinding);
```

See Also

[바인딩 후 데이터 보내기](#)

FlushDataBeforeBinding

데이터 셋 바인딩 후에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부

Prototype

bool FlushDataBeforeBinding

Example

```
var ds = This.GetDataSet();  
_DEBUG("FlushDataBeforeBinding " + ds.FlushDataBeforeBinding);
```

See Also

[바인딩 전 데이터 보내기](#)

Hidden

데이터 셋 숨김 여부

Prototype

bool Hidden

Example

```
var ds = This.GetDataSet();  
_DEBUG("Hidden " + ds.Hidden);
```

See Also

[숨김](#)

MaxRowCount

최대 행 수

Prototype

integer MaxRowCount

Example

```
var ds = This.GetDataSet();  
_DEBUG("MaxRowCount " + ds.MaxRowCount);
```

See Also

[최대행 수](#)

Method

Name	Description
GetDataFieldNames	해당 데이터셋에 추가된 모든 데이터 필드 이름을 가져옵니다.
GetDataModule	데이터 모듈을 가져옵니다.
GetDataStore	데이터 스토어를 가져옵니다.
GetDetailDataSets	디테일셋 이름을 가져옵니다.
GetField	해당 이름의 필드 객체를 가져옵니다.
GetMasterDataSetName	마스터셋 이름을 가져옵니다.

GetName	데이터셋 이름을 가져옵니다.
IsMasterDataSet	해당 데이터셋이 마스터셋인지 여부를 가져옵니다.

GetDataFieldNames

해당 데이터셋에 추가된 모든 데이터 필드 이름을 가져옵니다.

Prototype

array [GetDataFieldNames\(\)](#)

Example

```
var ds = This.GetDataSet();
var DataFieldNames = new Array();
DataFieldNames = ds.GetDataFieldNames();
```

GetDataModule

데이터 모듈을 가져옵니다.

Prototype

object [GetDataModule\(\)](#)

Example

```
var ds = This.GetDataSet();
var DataModule = ds.GetDataModule();
```

GetDataStore

데이터 스토어를 가져옵니다.

Prototype

object [GetDataStore\(\)](#)

Example

```
var ds = This.GetDataSet();
var DataStore = ds.GetDataStore();
```

GetDetailDataSets

디테일셋 이름을 가져옵니다.

Prototype

array GetDetailDataSets()

Example

```
var ds = This.GetDataSet();  
var DetailDataSets = new Array();  
DetailDataSets = ds.GetDetailDataSets();
```

GetField

해당 이름의 필드 객체를 가져옵니다.

Prototype

object GetField(string FieldName)

Parameters

FieldName	필드 이름
-----------	-------

Reference

데이터 필드 또는 수식 필드 객체를 가져오며, 리턴되는 필드의 종류를 알기 위해서는 IsCalculatedField() 함수를 이용하시기 바랍니다.

Example

```
var ds = This.GetDataSet();  
var Field = ds.GetField("field1");
```

See Also

[IsCalculatedField\(OZCalculatedField\)](#), [IsCalculatedField\(OZDataField\)](#),
[IsCalculatedField\(OZXPathDataField\)](#)

GetMasterDataSetName

마스터셋 이름을 가져옵니다.

Prototype

string GetMasterDataSetName()

Reference

마스터셋이 없을 경우 공백 값을 가져옵니다.

Example

```
var ds = This.GetDataSet();  
var MasterDataSetName = ds.GetMasterDataSetName();
```

See Also

[마스터셋](#)

GetName

데이터셋 이름을 가져옵니다.

Prototype

string GetName()

Example

```
var ds = This.GetDataSet();  
var Name = ds.GetName();
```

See Also

[이름](#)

IsMasterDataSet

해당 데이터셋이 마스터셋인지 여부를 가져옵니다.

Prototype

bool IsMasterDataSet()

Example

```
var ds = This.GetDataSet();  
if(ds.IsMasterDataSet() == true) {  
    _DEBUG("Return true");  
} else {  
    _DEBUG("Return false");  
}
```

Event

Name	Description
OnFinalize	바인딩 요청 시 데이터 스토어 삭제 직전에 발생합니다.
OnInitialize	바인딩 요청 시 데이터 스토어 초기화 직전에 발생합니다.

OnFinalize

바인딩 요청 시 데이터 스토어 삭제 직전에 발생합니다.

Prototype

[void OnFinalize\(\)](#)

OnInitialize

바인딩 요청 시 데이터 스토어 초기화 직전에 발생합니다.

Prototype

[void OnInitialize\(\)](#)

Reference

해당 이벤트에서 속성 값을 변경할 수 있습니다.

OZSAPDataSet

- [Property](#)
- [Method](#)

- [Event](#)

Property

Name	Description
ConcurrentFetchSize	데이터 요청 시 한꺼번에 가져올 결과 데이터 셋의 크기
ConcurrentFirstRow	데이터 요청 시 처음 전송할 데이터의 행 수
FlushDataAfterBinding	데이터 셋 바인딩 전에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부
FlushDataBeforeBinding	데이터 셋 바인딩 후에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부
Hidden	데이터 셋 숨김 여부
MappingKey	맵핑 키

ConcurrentFetchSize

데이터 요청 시 한꺼번에 가져올 결과 데이터 셋의 크기

Prototype

integer [ConcurrentFetchSize](#)

Reference

fetch 타입이 Concurrent일 경우에만 해당 속성이 적용됩니다.

Example

```
var ds = This.GetDataSet();
_DEBUG("ConcurrentFetchSize " + ds.ConcurrentFetchSize);
```

See Also

[Concurrent 전송 크기](#)

ConcurrentFirstRow

데이터 요청 시 처음 전송할 데이터의 행 수

Prototype

integer ConcurrentFirstRow

Reference

fetch 타입이 Concurrent일 경우에만 해당 속성이 적용됩니다.

Example

```
var ds = This.GetDataSet();  
_DEBUG("ConcurrentFirstRow " + ds.ConcurrentFirstRow);
```

See Also

[Concurrent 첫 전송 Row](#)

FlushDataAfterBinding

데이터 셋 바인딩 전에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부

Prototype

bool FlushDataAfterBinding

Example

```
var ds = This.GetDataSet();  
_DEBUG("FlushDataAfterBinding " + ds.FlushDataAfterBinding);
```

See Also

[바인딩 후 데이터 보내기](#)

FlushDataBeforeBinding

데이터 셋 바인딩 후에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부

Prototype

bool FlushDataBeforeBinding

Example

```
var ds = This.GetDataSet();  
_DEBUG("FlushDataBeforeBinding " + ds.FlushDataBeforeBinding);
```

See Also

[바인딩 전 데이터 보내기](#)

FunctionName

ABAP 함수 이름

Prototype

string FunctionName

FunctionType

ABAP 함수 타입

Prototype

string FunctionType

Hidden

데이터 셋 숨김 여부

Prototype

bool Hidden

Example

```
var ds = This.GetDataSet();  
_DEBUG("Hidden " + ds.Hidden);
```

See Also

[숨김](#)

MappingKey

맵핑 키

Prototype

string MappingKey

Example

```
var ds = This.GetDataSet();  
_DEBUG("MappingKey " + ds.MappingKey);
```

Method

Name	Description
GetCalculatedFieldNames	해당 데이터셋에 추가된 모든 수식 필드 이름을 가져옵니다
GetDataFieldNames	해당 데이터셋에 추가된 모든 데이터 필드 이름을 가져옵니다.
GetDataModule	데이터 모듈을 가져옵니다.
GetDataStore	데이터 스토어를 가져옵니다.
GetDetailDataSets	디테일셋 이름을 가져옵니다.
GetField	해당 이름의 필드 객체를 가져옵니다.
GetInputParam	지정한 입력 패러미터 값을 가져옵니다.
GetInputParamNames	입력 패러미터 이름을 모두 가져옵니다.
GetName	데이터셋 이름을 가져옵니다.
IsMasterDataSet	해당 데이터셋이 마스터셋인지 여부를 가져옵니다.
SetInputParam	지정한 입력 패러미터의 값을 설정합니다.

GetCalculatedFieldNames

해당 데이터셋에 추가된 모든 수식 필드 이름을 가져옵니다.

Prototype

array GetCalculatedFieldNames()

Example

```
var ds = This.GetDataSet();  
var CalculatedFieldNames = new Array();  
CalculatedFieldNames = ds.GetCalculatedFieldNames();
```

GetDataFieldNames

해당 데이터셋에 추가된 모든 데이터 필드 이름을 가져옵니다.

Prototype

array [GetDataFieldNames\(\)](#)

Example

```
var ds = This.GetDataSet();  
var DataFieldNames = new Array();  
DataFieldNames = ds.GetDataFieldNames();
```

GetDataModule

데이터 모듈을 가져옵니다.

Prototype

object [GetDataModule\(\)](#)

Example

```
var ds = This.GetDataSet();  
var DataModule = ds.GetDataModule();
```

GetDataStore

데이터 스토어를 가져옵니다.

Prototype

object [GetDataStore\(\)](#)

Example

```
var ds = This.GetDataSet();  
var DataStore = ds.GetDataStore();
```

GetDetailDataSets

디테일셋 이름을 가져옵니다.

Prototype

array GetDetailDataSets()

Example

```
var ds = This.GetDataSet();  
var DetailDataSets = new Array();  
DetailDataSets = ds.GetDetailDataSets();
```

GetField

해당 이름의 필드 객체를 가져옵니다.

Prototype

object GetField(string FieldName)

Parameters

FieldName	필드 이름
-----------	-------

Reference

데이터 필드 또는 수식 필드 객체를 가져오며, 리턴되는 필드의 종류를 알기 위해서는 IsCalculatedField() 함수를 이용하시기 바랍니다.

Example

```
var ds = This.GetDataSet();  
var Field = ds.GetField("field1");
```

See Also

[IsCalculatedField\(OZCalculatedField\)](#), [IsCalculatedField\(OZDataField\)](#),
[IsCalculatedField\(OZXPathDataField\)](#)

GetInputParam

지정한 입력 패러미터 값을 가져옵니다.

Prototype

string [GetInputParam\(string InputParamName\)](#)

Parameters

InputParamName	패러미터 이름
----------------	---------

Example

```
var ds = This.GetDataSet();  
var InputParamName = ds.GetInputParam();
```

GetInputParamNames

입력 패러미터 이름을 모두 가져옵니다.

Prototype

array [GetInputParamNames\(\)](#)

GetName

데이터셋 이름을 가져옵니다.

Prototype

string [GetName\(\)](#)

Example

```
var ds = This.GetDataSet();  
var Name = ds.GetName();
```

See Also

[이름](#)

IsMasterDataSet

해당 데이터셋이 마스터셋인지 여부를 가져옵니다.

Prototype

bool IsMasterDataSet()

Example

```
var ds = This.GetDataSet();
if(ds.IsMasterDataSet() == true) {
    _DEBUG("Return true");
} else {
    _DEBUG("Return false");
}
```

SetInputParam

지정한 입력 패러미터의 값을 설정합니다.

Prototype

void SetInputParam(string InputParamName, string InputParamValue)

Parameters

InputParamName	패러미터 이름
InputParamValue	패러미터 값

Example

```
var ds = This.GetDataSet();
ds.SetInputParam("Param1", "Value1");
```

Event

Name	Description
OnFinalize	바인딩 요청 시 데이터 스토어 삭제 직전에 발생합니다.
OnInitialize	바인딩 요청 시 데이터 스토어 초기화 직전에 발생합니다.

OnFinalize

바인딩 요청 시 데이터 스토어 삭제 직전에 발생합니다.

Prototype

`void OnFinalize()`

OnInitialize

바인딩 요청 시 데이터 스토어 초기화 직전에 발생합니다.

Prototype

`void OnInitialize()`

Reference

해당 이벤트에서 속성 값을 변경할 수 있습니다.

OZUserDataSet

- [Property](#)
- [Method](#)
- [Event](#)

Property

Name	Description
Command	실행문
ConcurrentFetchSize	데이터 요청 시 한꺼번에 가져올 결과 데이터 셋의 크기
ConcurrentFirstRow	데이터 요청 시 처음 전송할 데이터의 행 수
FlushDataAfterBinding	데이터 셋 바인딩 전에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부
FlushDataBeforeBinding	데이터 셋 바인딩 후에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부
Hidden	데이터 셋 숨김 여부
UseOZScript	스크립트 사용 여부

Command

실행문

Prototype

string Command

Example

```
var ds = This.GetDataSet();  
_DEBUG("Command " + ds.Command);
```

See Also

[실행문](#)

ConcurrentFetchSize

데이터 요청 시 한꺼번에 가져올 결과 데이터 셋의 크기

Prototype

integer ConcurrentFetchSize

Reference

fetch 타입이 Concurrent일 경우에만 해당 속성이 적용됩니다.

Example

```
var ds = This.GetDataSet();  
_DEBUG("ConcurrentFetchSize " + ds.ConcurrentFetchSize);
```

See Also

[Concurrent 전송 크기](#)

ConcurrentFirstRow

데이터 요청 시 처음 전송할 데이터의 행 수

Prototype

integer ConcurrentFirstRow

Reference

fetch 타입이 Concurrent일 경우에만 해당 속성이 적용됩니다.

Example

```
var ds = This.GetDataSet();  
_DEBUG("ConcurrentFirstRow " + ds.ConcurrentFirstRow);
```

See Also

[Concurrent 첫 전송 Row](#)

FlushDataAfterBinding

데이터 셋 바인딩 전에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부

Prototype

bool FlushDataAfterBinding

Example

```
var ds = This.GetDataSet();  
_DEBUG("FlushDataAfterBinding " + ds.FlushDataAfterBinding);
```

See Also

[바인딩 후 데이터 보내기](#)

FlushDataBeforeBinding

데이터 셋 바인딩 후에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부

Prototype

bool FlushDataBeforeBinding

Example

```
var ds = This.GetDataSet();  
_DEBUG("FlushDataBeforeBinding " + ds.FlushDataBeforeBinding);
```

See Also

[바인딩 전 데이터 보내기](#)

Hidden

데이터 셋 숨김 여부

Prototype

bool Hidden

Example

```
var ds = This.GetDataSet();  
_DEBUG("Hidden " + ds.Hidden);
```

See Also

[숨김](#)

UseOZScript

스크립트 사용 여부

Prototype

bool UseOZScript

Example

```
var ds = This.GetDataSet();  
_DEBUG("UseOZScript " + ds.UseOZScript);
```

See Also

[스크립트 사용](#)

Method

Name	Description
GetCalculatedFieldNames	해당 데이터셋에 추가된 모든 수식 필드 이름을 가져옵니다

GetDataFieldNames	해당 데이터셋에 추가된 모든 데이터 필드 이름을 가져옵니다.
GetDataModule	데이터 모듈을 가져옵니다.
GetDestinationFields	Destination 필드 이름을 가져옵니다.
GetDestinationValue	Destination 필드 값을 가져옵니다.
GetDestinationValues	Destination 필드 값을 가져옵니다.
GetDataStore	데이터 스토어를 가져옵니다.
GetDetailDataSets	디테일셋 이름을 가져옵니다.
GetField	해당 이름의 필드 객체를 가져옵니다.
GetMasterDataSetName	마스터셋 이름을 가져옵니다.
GetName	데이터셋 이름을 가져옵니다.
GetSourceFields	Source 필드 이름을 가져옵니다.
GetSourceValue	Source 필드 값을 가져옵니다.
GetSourceValues	Source 필드 값을 가져옵니다.
IsMasterDataSet	해당 데이터셋이 마스터셋인지 여부를 가져옵니다.

GetCalculatedFieldNames

해당 데이터셋에 추가된 모든 수식 필드 이름을 가져옵니다.

Prototype

array [GetCalculatedFieldNames\(\)](#)

Example

```
var ds = This.GetDataSet();
var CalculatedFieldNames = new Array();
CalculatedFieldNames = ds.GetCalculatedFieldNames();
```

GetDataFieldNames

해당 데이터셋에 추가된 모든 데이터 필드 이름을 가져옵니다.

Prototype

array [GetDataFieldNames\(\)](#)

Example

```
var ds = This.GetDataSet();  
var DataFieldNames = new Array();  
DataFieldNames = ds.GetDataFieldNames();
```

GetDataModule

데이터 모듈을 가져옵니다.

Prototype

object [GetDataModule\(\)](#)

Example

```
var ds = This.GetDataSet();  
var DataModule = ds.GetDataModule();
```

GetDestinationFields

Destination 필드 이름을 가져옵니다.

Prototype

array [GetDestinationFields\(\)](#)

Reference

OnInsert, OnUpdate, OnDelete 이벤트에서 실행한 경우에만 동작하며, 그 외 이벤트에서 실행하였을 경우 null을 리턴합니다.

Example

```
var ds = This.GetDataSet();  
var DestinationFields = new Array();  
DestinationFields = ds.GetDestinationFields();
```

GetDestinationValue

Destination 필드 값을 가져옵니다.

Prototype

```
var GetDestinationValue(string FieldName)
```

Parameters

FieldName	Destination 필드 값을 가져올 필드 이름
-----------	-----------------------------

Reference

OnInsert, OnUpdate, OnDelete 이벤트에서 실행한 경우에만 동작하며, 그 외 이벤트에서 실행하였을 경우 null을 리턴합니다.

리턴되는 필드 값은 필드 타입에 따라 문자, 숫자, bool, Date 객체, [OZBinary](#) 객체로 리턴됩니다.

필드 타입에 따라 리턴되는 값의 타입은 아래 표를 참고하시기 바랍니다.

Name	Description
TINYINT	숫자 값
OZSMALLINT	숫자 값
INTEGER	숫자 값
BIGINT	숫자 값
REAL	숫자 값
FLOAT	숫자 값
DOUBLE	숫자 값
BIT	true, false
CHAR	문자 값
VARCHAR	문자 값
LONGVARCHAR	문자 값
NUMERIC	숫자 값
DECIMAL	숫자 값
DATE	자바 스크립트 Date 객체
TIME	자바 스크립트 Date 객체

TIMESTAMP	자바 스크립트 Date 객체
BINARY	OZBinary 객체
VARBINARY	OZBinary 객체
LONGBINARY	OZBinary 객체
BLOB	OZBinary 객체
CLOB	문자 값

Example

```

var sourceNames = This.GetSourceFields();
PrintArray(This, sourceNames, "Source names ");
PrintArray(This, This.GetSourceValues(), "Source values ");

var destNames = This.GetDestinationFields();
PrintArray(This, destNames, "Destination names ");
PrintArray(This, This.GetDestinationValues(), "Destination values ");

if(destNames != undefined) {
    for(var i = 0; i < destNames.length; i++) {
        var value = This.GetDestinationValue(destNames[i]);
        _DEBUG("Value of '" + destNames[i] + "' is " + value);
    }
}

return "Success";

```

GetDestinationValues

Destination 필드 값을 가져옵니다.

Prototype

array [GetDestinationValues\(\)](#)

Reference

OnInsert, OnUpdate, OnDelete 이벤트에서 실행한 경우에만 동작하며, 그 외 이벤트에서 실행하였을 경우 null을 리턴합니다.

리턴되는 필드 값은 필드 타입에 따라 문자, 숫자, bool, Date 객체, [OZBinary](#) 객체로 리턴됩니다.
필드 타입에 따라 리턴되는 값의 타입은 [GetDestinationValue\(\)](#) 함수의 표를 참고하시기 바랍니다.

Example

```
var ds = This.GetDataSet();  
var DestinationValues = new Array();  
DestinationValues = ds.GetDestinationValues();
```

GetDataStore

데이터 스토어를 가져옵니다.

Prototype

[object](#) [GetDataStore\(\)](#)

Example

```
var ds = This.GetDataSet();  
var DataStore = ds.GetDataStore();
```

GetDetailDataSets

디테일셋 이름을 가져옵니다.

Prototype

[array](#) [GetDetailDataSets\(\)](#)

Example

```
var ds = This.GetDataSet();  
var DetailDataSets = new Array();  
DetailDataSets = ds.GetDetailDataSets();
```

GetField

해당 이름의 필드 객체를 가져옵니다.

Prototype

[object](#) [GetField\(string fieldName\)](#)

Parameters

FieldName	필드 이름
-----------	-------

Reference

데이터 필드 또는 수식 필드 객체를 가져오며, 리턴되는 필드의 종류를 알기 위해서는 IsCalculatedField() 함수를 이용하시기 바랍니다.

Example

```
var ds = This.GetDataSet();  
var Field = ds.GetField("field1");
```

See Also

[IsCalculatedField\(OZCalculatedField\)](#), [IsCalculatedField\(OZDataField\)](#),
[IsCalculatedField\(OZXPathDataField\)](#)

GetMasterDataSetName

마스터셋 이름을 가져옵니다.

Prototype

string GetMasterDataSetName()

Reference

마스터셋이 없을 경우 공백 값을 가져옵니다.

Example

```
var ds = This.GetDataSet();  
var MasterDataSetName = ds.GetMasterDataSetName();
```

See Also

[마스터셋](#)

GetName

데이터셋 이름을 가져옵니다.

Prototype

string GetName()

Example

```
var ds = This.GetDataSet();  
var Name = ds.GetName();
```

See Also

[이름](#)

GetSourceFields

Source 필드 이름을 가져옵니다.

Prototype

array GetSourceFields()

Reference

OnInsert, OnUpdate, OnDelete 이벤트에서 실행한 경우에만 동작하며, 그 외 이벤트에서 실행하였을 경우 null을 리턴합니다.

Example

```
var ds = This.GetDataSet();  
var SourceFields = new Array();  
SourceFields = ds.GetSourceFields();
```

GetSourceValue

Source 필드 값을 가져옵니다.

Prototype

var GetSourceValue(string FieldName)

Parameters

FieldName	Source 필드 값을 가져올 필드 이름
-----------	------------------------

Reference

OnInsert, OnUpdate, OnDelete 이벤트에서 실행한 경우에만 동작하며, 그 외 이벤트에서 실행하였을 경우 null을 리턴합니다.

리턴되는 필드 값은 필드 타입에 따라 문자, 숫자, bool, Date 객체, [OZBinary](#) 객체로 리턴됩니다.

필드 타입에 따라 리턴되는 값의 타입은 아래 표를 참고하시기 바랍니다.

Name	Description
TINYINT	숫자 값
OZSMALLINT	숫자 값
INTEGER	숫자 값
BIGINT	숫자 값
REAL	숫자 값
FLOAT	숫자 값
DOUBLE	숫자 값
BIT	true, false
CHAR	문자 값
VARCHAR	문자 값
LONGVARCHAR	문자 값
NUMERIC	숫자 값
DECIMAL	숫자 값
DATE	자바 스크립트 Date 객체
TIME	자바 스크립트 Date 객체
TIMESTAMP	자바 스크립트 Date 객체
BINARY	OZBinary 객체
VARBINARY	OZBinary 객체
LONGBINARY	OZBinary 객체
BLOB	OZBinary 객체

CLOB	문자 값
------	------

Example

```

var sourceNames = This.GetSourceFields();
PrintArray(This, sourceNames, "Source names ");
PrintArray(This, This.GetSourceValues(), "Source values ");

var destNames = This.GetDestinationFields();
PrintArray(This, destNames, "Destination names ");
PrintArray(This, This.GetDestinationValues(), "Destination values ");

if(sourceNames != undefined) {
    for(var i = 0; i < sourceNames.length; i++) {
        var value = This.GetSourceValue(sourceNames[i]);
        _DEBUG("Value of '" + sourceNames[i] + "' is " + value);
    }
}

return "Success";

```

GetSourceValues

Source 필드 값을 가져옵니다.

Prototype

array [GetSourceValues\(\)](#)

Reference

OnInsert, OnUpdate, OnDelete 이벤트에서 실행한 경우에만 동작하며, 그 외 이벤트에서 실행하였을 경우 null을 리턴합니다.

리턴되는 필드 값은 필드 타입에 따라 문자, 숫자, bool, Date 객체, [OZBinary](#) 객체로 리턴됩니다.

필드 타입에 따라 리턴되는 값의 타입은 [GetSourceValue\(\)](#) 함수의 표를 참고하시기 바랍니다.

Example

```

var ds = This.GetDataSet();
var SourceValues = new Array();
SourceValues = ds.GetSourceValues();

```

IsMasterDataSet

해당 데이터셋이 마스터셋인지 여부를 가져옵니다.

Prototype

bool IsMasterDataSet()

Example

```
var ds = This.GetDataSet();  
if(ds.IsMasterDataSet() == true) {  
    _DEBUG("Return true");  
} else {  
    _DEBUG("Return false");  
}
```

Event

Name	Description
OnFinalize	바인딩 요청 시 데이터 스토어 삭제 직전에 발생합니다.
OnInitialize	바인딩 요청 시 데이터 스토어 초기화 직전에 발생합니다.
OnPostBind	바인딩 실행 후에 발생합니다.
OnPreBind	바인딩 실행 직전에 발생합니다.

OnFinalize

바인딩 요청 시 데이터 스토어 삭제 직전에 발생합니다.

Prototype

void OnFinalize()

OnInitialize

바인딩 요청 시 데이터 스토어 초기화 직전에 발생합니다.

Prototype

void OnInitialize()

Reference

해당 이벤트에서 속성 값을 변경할 수 있습니다.

OnPostBind

바인딩 실행 후에 발생합니다.

Prototype

void OnPostBind()

OnPreBind

바인딩 실행 직전에 발생합니다.

Prototype

void OnPreBind()

Reference

동적으로 쿼리를 변경하거나 자체 쿼리 실행으로 결과셋 생성이 가능합니다.

디테일셋일 경우 마스터셋의 결과셋 행 수만큼 이벤트가 발생합니다.

OZJSResultSet 객체를 가져올 수 있습니다.

null 또는 OZJSResultSet 이외의 타입을 리턴하며 리턴이 없을 경우 서버의 결과셋 생성 로직을 수행합니다

.

OZXMLDataSet

- [Property](#)
- [Method](#)
- [Event](#)

Property

Name	Description
------	-------------

ConcurrentFetchSize	데이터 요청 시 한꺼번에 가져올 결과 데이터 셋의 크기
ConcurrentFirstRow	데이터 요청 시 처음 전송할 데이터의 행 수
ConnectionRetryCount	에러 발생 시 연결을 재시도하는 횟수
DataSetNodeName	데이터셋 노드 이름
DoPost	URL의 패러미터 전송 시 포스트 방식으로 전송할지 여부
EncodingURL	Server URL 인코딩 유무를 설정합니다.
FlushDataAfterBinding	데이터 셋 바인딩 전에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부
FlushDataBeforeBinding	데이터 셋 바인딩 후에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부
Hidden	데이터 셋 숨김 여부
HTTPLogFileName	로그 파일 이름
MaxRowCount	최대 행 수
RecordNodeName	레코드셋 노드 이름
URL	데이터를 가져올 파일 또는 사용자 애플리케이션의 HTTP 주소

ConcurrentFetchSize

데이터 요청 시 한꺼번에 가져올 결과 데이터 셋의 크기

Prototype

integer [ConcurrentFetchSize](#)

Reference

fetch 타입이 Concurrent일 경우에만 해당 속성이 적용됩니다.

Example

```
var ds = This.GetDataSet();
_DEBUG("ConcurrentFetchSize " + ds.ConcurrentFetchSize);
```

See Also

[Concurrent 전송 크기\(파일 스토어 XML 데이터 셋\)](#), [Concurrent 전송 크기\(HTTP 스토어 XML 데이터 셋\)](#)

ConcurrentFirstRow

데이터 요청 시 처음 전송할 데이터의 행 수

Prototype

integer ConcurrentFirstRow

Reference

fetch 타입이 Concurrent일 경우에만 해당 속성이 적용됩니다.

Example

```
var ds = This.GetDataSet();  
_DEBUG("ConcurrentFirstRow " + ds.ConcurrentFirstRow);
```

See Also

[Concurrent 첫 전송 Row\(파일 스토어 XML 데이터 셋\)](#), [Concurrent 첫 전송 Row\(HTTP 스토어 XML 데이터 셋\)](#)

ConnectionRetryCount

에러 발생 시 연결을 재시도하는 횟수

Prototype

integer ConnectionRetryCount

Example

```
var ds = This.GetDataSet();  
_DEBUG("ConnectionRetryCount " + ds.ConnectionRetryCount);
```

DataSetNodeName

데이터셋 노드 이름

Prototype

string DataSetNodeName

Example

```
var ds = This.GetDataSet();  
_DEBUG("DataSetNodeName " + ds.DataSetNodeName);
```

See Also

[데이터 셋 노드 이름\(파일 스토어 XML 데이터 셋\)](#), [데이터 셋 노드 이름\(HTTP 스토어 XML 데이터 셋\)](#)

DoPost

URL의 패러미터 전송 시 포스트 방식으로 전송할지 여부

Prototype

bool DoPost

Reference

부모 스토어가 HTTP 스토어일 경우에만 적용됩니다.

Example

```
var ds = This.GetDataSet();  
_DEBUG("DoPost " + ds.DoPost);
```

EncodingURL

Server URL 인코딩 유무를 설정합니다.

Prototype

bool EncodingURL

Reference

부모 스토어가 HTTP 스토어일 경우에만 적용됩니다.

Example

```
var ds = This.GetDataSet();  
_DEBUG("EncodingURL " + ds.EncodingURL);
```

FlushDataAfterBinding

데이터 셋 바인딩 전에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부

Prototype

bool FlushDataAfterBinding

Example

```
var ds = This.GetDataSet();  
_DEBUG("FlushDataAfterBinding " + ds.FlushDataAfterBinding);
```

See Also

[바인딩 후 데이터 보내기\(파일 스토어 XML 데이터 셋\)](#), [바인딩 후 데이터 보내기\(HTTP 스토어 XML 데이터 셋\)](#)

FlushDataBeforeBinding

데이터 셋 바인딩 후에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부

Prototype

bool FlushDataBeforeBinding

Example

```
var ds = This.GetDataSet();  
_DEBUG("FlushDataBeforeBinding " + ds.FlushDataBeforeBinding);
```

See Also

[바인딩 전 데이터 보내기\(파일 스토어 XML 데이터 셋\)](#), [바인딩 전 데이터 보내기\(HTTP 스토어 XML 데이터 셋\)](#)

Hidden

데이터 셋 숨김 여부

Prototype

bool Hidden

Example

```
var ds = This.GetDataSet();  
_DEBUG("Hidden " + ds.Hidden);
```

See Also

[숨김\(파일 스토어 XML 데이터 셋\)](#), [숨김\(HTTP 스토어 XML 데이터 셋\)](#)

HTTPLogFileName

로그 파일 이름

Prototype

string HTTPLogFileName

Reference

부모 스토어가 HTTP 스토어일 경우에만 적용됩니다.

Example

```
var ds = This.GetDataSet();  
_DEBUG("HTTPLogFileName " + ds.HTTPLogFileName);
```

MaxRowCount

최대 행 수

Prototype

integer MaxRowCount

Example

```
var ds = This.GetDataSet();  
_DEBUG("MaxRowCount " + ds.MaxRowCount);
```

See Also

[최대행 수\(파일 스토어 XML 데이터 셋\)](#), [최대행 수\(HTTP 스토어 XML 데이터 셋\)](#)

RecordNodeName

레코드셋 노드 이름

Prototype

string RecordNodeName

Example

```
var ds = This.GetDataSet();  
_DEBUG("RecordNodeName " + ds.RecordNodeName);
```

See Also

[레코드셋 노드 이름\(파일 스토어 XML 데이터 셋\)](#), [레코드셋 노드 이름\(HTTP 스토어 XML 데이터 셋\)](#)

URL

데이터를 가져올 파일 또는 사용자 애플리케이션의 HTTP 주소

Prototype

string URL

Reference

부모 스토어가 HTTP, 파일 스토어일 경우에만 적용됩니다.

Example

```
var ds = This.GetDataSet();  
_DEBUG("URL " + ds.URL);
```

Method

Name	Description
GetCalculatedFieldNames	해당 데이터셋에 추가된 모든 수식 필드 이름을 가져옵니다
GetDataFieldNames	해당 데이터셋에 추가된 모든 데이터 필드 이름을 가져옵니다.
GetDataModule	데이터 모듈을 가져옵니다.
GetDataStore	데이터 스토어를 가져옵니다.
GetDetailDataSets	디테일셋 이름을 가져옵니다.

GetField	해당 이름의 필드 객체를 가져옵니다.
GetMasterDataSetName	마스터셋 이름을 가져옵니다.
GetName	데이터셋 이름을 가져옵니다.
IsFileStore	데이터 스토어에 파일 스토어인지 여부를 가져옵니다.
IsHTTPStore	데이터 스토어에 HTTP 스토어인지 여부를 가져옵니다.
IsMasterDataSet	해당 데이터셋이 마스터셋인지 여부를 가져옵니다.

GetCalculatedFieldNames

해당 데이터셋에 추가된 모든 수식 필드 이름을 가져옵니다.

Prototype

array [GetCalculatedFieldNames\(\)](#)

Example

```
var ds = This.GetDataSet();
var CalculatedFieldNames = new Array();
CalculatedFieldNames = ds.GetCalculatedFieldNames();
```

GetDataFieldNames

해당 데이터셋에 추가된 모든 데이터 필드 이름을 가져옵니다.

Prototype

array [GetDataFieldNames\(\)](#)

Example

```
var ds = This.GetDataSet();
var DataFieldNames = new Array();
DataFieldNames = ds.GetDataFieldNames();
```

GetDataModule

데이터 모듈을 가져옵니다.

Prototype

object GetDataModule()

Example

```
var ds = This.GetDataSet();  
var DataModule = ds.GetDataModule();
```

GetDataStore

데이터 스토어를 가져옵니다.

Prototype

object GetDataStore()

Example

```
var ds = This.GetDataSet();  
var DataStore = ds.GetDataStore();
```

GetDetailDataSets

디테일셋 이름을 가져옵니다.

Prototype

array GetDetailDataSets()

Exaple

```
var ds = This.GetDataSet();  
var DetailDataSets = new Array();  
DetailDataSets = ds.GetDetailDataSets();
```

GetField

해당 이름의 필드 객체를 가져옵니다.

Prototype

object GetField(string FieldName)

Parameters

FieldName	필드 이름
-----------	-------

Reference

데이터 필드 또는 수식 필드 객체를 가져오며, 리턴되는 필드의 종류를 알기 위해서는 IsCalculatedField() 함수를 이용하시기 바랍니다.

Example

```
var ds = This.GetDataSet();  
var Field = ds.GetField("field1");
```

See Also

[IsCalculatedField\(OZCalculatedField\)](#), [IsCalculatedField\(OZDataField\)](#),
[IsCalculatedField\(OZXPathDataField\)](#)

GetMasterDataSetName

마스터셋 이름을 가져옵니다.

Prototype

string GetMasterDataSetName()

Reference

마스터셋이 없을 경우 공백 값을 가져옵니다.

Example

```
var ds = This.GetDataSet();  
var MasterDataSetName = ds.GetMasterDataSetName();
```

See Also

[마스터셋\(파일 스토어\)](#), [마스터셋\(XML 스토어\)](#)

GetName

데이터셋 이름을 가져옵니다.

Prototype

string GetName()

Example

```
var ds = This.GetDataSet();  
var Name = ds.GetName();
```

See Also

[이름\(파일 스토어\)](#), [이름\(XML 스토어\)](#)

IsFileStore

데이터 스토어가 파일 스토어인지 여부를 가져옵니다.

Prototype

bool IsFileStore()

Example

```
var ds = This.GetDataSet();  
if(ds.IsFileStore() == true) {  
    _DEBUG("Return true");  
} else {  
    _DEBUG("Return false");  
}
```

IsHTTPStore

데이터 스토어가 HTTP 스토어인지 여부를 가져옵니다.

Prototype

bool IsHTTPStore()

Example

```
var ds = This.GetDataSet();  
if(ds.IsHTTPStore() == true) {  
    _DEBUG("Return true");  
}
```

```

} else {
    _DEBUG("Return false");
}

```

IsMasterDataSet

해당 데이터셋이 마스터셋인지 여부를 가져옵니다.

Prototype

bool IsMasterDataSet()

Example

```

var ds = This.GetDataSet();
if(ds.IsMasterDataSet() == true) {
    _DEBUG("Return true");
} else {
    _DEBUG("Return false");
}

```

Event

Name	Description
OnFinalize	바인딩 요청 시 데이터 스토어 삭제 직전에 발생합니다.
OnInitialize	바인딩 요청 시 데이터 스토어 초기화 직전에 발생합니다.

OnFinalize

바인딩 요청 시 데이터 스토어 삭제 직전에 발생합니다.

Prototype

void OnFinalize()

OnInitialize

바인딩 요청 시 데이터 스토어 초기화 직전에 발생합니다.

Prototype

void OnInitialize()

Reference

해당 이벤트에서 속성 값을 변경할 수 있습니다.

OZXPathDataSet

- [Property](#)
- [Method](#)
- [Event](#)

Property

Name	Description
BaseXPath	기본 XPath
BaseXPathCondition	XPath 조건문
ConcurrentFetchSize	데이터 요청 시 한꺼번에 가져올 결과 데이터 셋의 크기
ConcurrentFirstRow	데이터 요청 시 처음 전송할 데이터의 행 수
FlushDataAfterBinding	데이터 셋 바인딩 전에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부
FlushDataBeforeBinding	데이터 셋 바인딩 후에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부
Hidden	데이터 셋 숨김 여부

BaseXPath

기본 XPath

Prototype

string BaseXPath

Example

```
var ds = This.GetDataSet();  
_DEBUG("BaseXPath " + ds.BaseXPath);
```

BaseXPathCondition

XPath 조건문

Prototype

string BaseXPathCondition

Example

```
var ds = This.GetDataSet();  
_DEBUG("BaseXPathCondition " + ds.BaseXPathCondition);
```

ConcurrentFetchSize

데이터 요청 시 한꺼번에 가져올 결과 데이터 셋의 크기

Prototype

integer ConcurrentFetchSize

Reference

fetch 타입이 Concurrent일 경우에만 해당 속성이 적용됩니다.

Example

```
var ds = This.GetDataSet();  
_DEBUG("ConcurrentFetchSize " + ds.ConcurrentFetchSize);
```

See Also

[Concurrent 전송 크기](#)

ConcurrentFirstRow

데이터 요청 시 처음 전송할 데이터의 행 수

Prototype

integer ConcurrentFirstRow

Reference

fetch 타입이 Concurrent일 경우에만 해당 속성이 적용됩니다.

Example

```
var ds = This.GetDataSet();  
_DEBUG("ConcurrentFirstRow " + ds.ConcurrentFirstRow);
```

See Also

[Concurrent 첫 전송 Row](#)

FlushDataAfterBinding

데이터 셋 바인딩 전에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부

Prototype

bool FlushDataAfterBinding

Example

```
var ds = This.GetDataSet();  
_DEBUG("FlushDataAfterBinding " + ds.FlushDataAfterBinding);
```

See Also

[바인딩 후 데이터 보내기](#)

FlushDataBeforeBinding

데이터 셋 바인딩 후에 출력 스트림 버퍼에 있는 데이터를 전송한 후 버퍼를 비울지 여부

Prototype

bool FlushDataBeforeBinding

Example

```
var ds = This.GetDataSet();
_DEBUG("FlushDataBeforeBinding " + ds.FlushDataBeforeBinding);
```

See Also

[바인딩 전 데이터 보내기](#)

Hidden

데이터 셋 숨김 여부

Prototype

bool Hidden

Example

```
var ds = This.GetDataSet();
_DEBUG("Hidden " + ds.Hidden);
```

See Also

[숨김](#)

Method

Name	Description
GetDataFieldNames	해당 데이터셋에 추가된 모든 데이터 필드 이름을 가져옵니다.
GetDataModule	데이터 모듈을 가져옵니다.
GetDataStore	데이터 스토어를 가져옵니다.
GetDetailDataSets	디테일셋 이름을 가져옵니다.
GetField	해당 이름의 필드 객체를 가져옵니다.
GetMasterDataSetName	마스터셋 이름을 가져옵니다.
GetName	데이터셋 이름을 가져옵니다.
IsMasterDataSet	해당 데이터셋이 마스터셋인지 여부를 가져옵니다.

GetDataFieldNames

해당 데이터셋에 추가된 모든 데이터 필드 이름을 가져옵니다.

Prototype

array GetDataFieldNames()

Example

```
var ds = This.GetDataSet();  
var DataFieldNames = new Array();  
DataFieldNames = ds.GetDataFieldNames();
```

GetDataModule

데이터 모듈을 가져옵니다.

Prototype

object GetDataModule()

Example

```
var ds = This.GetDataSet();  
var DataModule = ds.GetDataModule();
```

GetDataStore

데이터 스토어를 가져옵니다.

Prototype

object GetDataStore()

Example

```
var ds = This.GetDataSet();  
var DataStore = ds.GetDataStore();
```

GetDetailDataSets

디테일셋 이름을 가져옵니다.

Prototype

array GetDetailDataSets()

Example

```
var ds = This.GetDataSet();  
var DetailDataSets = new Array();  
DetailDataSets = ds.GetDetailDataSets();
```

GetField

해당 이름의 필드 객체를 가져옵니다.

Prototype

object GetField(string fieldName)

Parameters

fieldName	필드 이름
-----------	-------

Reference

데이터 필드 또는 수식 필드 객체를 가져오며, 리턴되는 필드의 종류를 알기 위해서는 IsCalculatedField() 함수를 이용하시기 바랍니다.

Example

```
var ds = This.GetDataSet();  
var Field = ds.GetField("field1");
```

See Also

[IsCalculatedField\(OZCalculatedField\)](#), [IsCalculatedField\(OZDataField\)](#),
[IsCalculatedField\(OZXPathDataField\)](#)

GetMasterDataSetName

마스터셋 이름을 가져옵니다.

Prototype

string GetMasterDataSetName()

Reference

마스터셋이 없을 경우 공백 값을 가져옵니다.

Example

```
var ds = This.GetDataSet();  
var MasterDataSetName = ds.GetMasterDataSetName();
```

See Also

[마스터셋](#)

GetName

데이터셋 이름을 가져옵니다.

Prototype

string GetName()

Example

```
var ds = This.GetDataSet();  
var Name = ds.GetName();
```

See Also

[이름](#)

IsMasterDataSet

해당 데이터셋이 마스터셋인지 여부를 가져옵니다.

Prototype

bool IsMasterDataSet()

Example

```
var ds = This.GetDataSet();  
if(ds.IsMasterDataSet() == true) {  
    _DEBUG("Return true");  
} else {
```

```
_DEBUG("Return false");  
}
```

Event

Name	Description
OnFinalize	바인딩 요청 시 데이터 스토어 삭제 직전에 발생합니다.
OnInitialize	바인딩 요청 시 데이터 스토어 초기화 직전에 발생합니다.

OnFinalize

바인딩 요청 시 데이터 스토어 삭제 직전에 발생합니다.

Prototype

[void OnFinalize\(\)](#)

OnInitialize

바인딩 요청 시 데이터 스토어 초기화 직전에 발생합니다.

Prototype

[void OnInitialize\(\)](#)

Reference

해당 이벤트에서 속성 값을 변경할 수 있습니다.

DataField

- [OZCalculatedField](#)
- [OZDataField](#)
- [OZXPathDataField](#)

OZCalculatedField

- [Method](#)
- [Event](#)

Method

Name	Description
GetDataModule	데이터 모듈을 가져옵니다.
GetDataSetName	데이터셋 이름을 가져옵니다.
GetDataSetRowIndex	데이터셋의 현재 레코드 커서 인덱스를 가져옵니다.
GetDataSetValue	해당 필드에 대한 데이터 값을 가져옵니다.
GetFieldName	필드 이름을 가져옵니다.
GetFieldType	필드 타입을 가져옵니다.
IsCalculatedField	수식 필드인지 여부를 가져옵니다.
IsDataSetValueNull	해당 필드 값이 null 값인지 여부를 가져옵니다.
IsDecrypt	필드의 복호화 여부를 가져옵니다.
IsNull	데이터 값이 null 값인지 여부를 가져옵니다.
SetFieldData	수식 필드의 값을 설정합니다.

GetDataModule

데이터 모듈을 가져옵니다.

Prototype

object [GetDataModule\(\)](#)

Example

```
var dm = This.GetDataModule();
var ds = dm.GetDataStore("DataStoreName");
var con = ds.GetConnection();
var stmt = con.CreateStatement();
var rs = stmt.ExecuteQuery("select * from customer");
rs.Close();
stmt.Close();
```

GetDataSetName

데이터셋 이름을 가져옵니다.

Prototype

string GetDataSetName()

Reference

사용자 지정 패러미터의 수식 필드인 경우에는 "OZParam"을 리턴합니다.

Example

```
This.SetFieldData(This.GetDataSetName());
```

GetDataSetRowIndex

데이터셋의 현재 레코드 커서 인덱스를 가져옵니다.

Prototype

integer GetDataSetRowIndex()

Reference

리턴되는 인덱스는 One-Based Index입니다.

사용자 지정 패러미터에 추가된 수식 필드인 경우 항상 0을 리턴합니다.

Example

```
This.SetFieldData(This.GetDataSetRowIndex());
```

GetDataSetValue

해당 필드에 대한 데이터 값을 가져옵니다.

Prototype

var GetDataSetValue(string FieldPath)

Parameters

FieldPath	필드 경로 값을 가져올 필드 종류에 따라 다르게 설정
-----------	----------------------------------

Reference

리턴 타입은 필드 타입에 따라 적절히 double, integer, string으로 변환됩니다.

필드 경로에 자기 자신을 설정할 경우 null 값으로 리턴됩니다.

값을 가져올 필드 종류에 따라 필드 경로 설정 방법과 참고사항은 다음과 같습니다.

- **데이터 필드**

"필드 이름" 또는 "데이터셋 이름.필드 이름" 형태로 설정합니다.

데이터셋 이름을 지정할 경우에는 같은 ODI 내에 있는 데이터셋으로 해당 수식 필드가 추가된 데이터셋 또는 해당 수식 필드가 추가된 데이터셋의 마스터셋인 경우에만 지정할 수 있습니다.

- **사용자 지정 패러미터 필드**

"OZParam.필드 이름" 형태로 설정합니다.

필드 경로에 마스터셋의 필드 또는 사용자 지정 패러미터 필드를 설정할 경우 쿼리 디자이너에서는 빈 값으로 표시되나, 리포트 뷰어에서는 정상적으로 표시됩니다.

Example 1

```
//데이터 필드(필드 이름 Price)
This.SetFieldData(This.GetDataSetValue("Price"));
```

Example 2

```
//데이터 필드(마스터셋 이름 Code, 필드 이름 Name)
This.SetFieldData(This.GetDataSetValue("Code.Name"));
```

Example 3

```
//사용자 지정 패러미터 필드(사용자 지정 패러미터 필드 이름 TotalAmount)
This.SetFieldData(This.GetDataSetValue("OZParam.TotalAmount"));
```

GetFieldName

필드 이름을 가져옵니다.

Prototype

string GetFieldName()

Example

```
_DEBUG("FieldName " + This.GetFieldName());
```

GetFieldType

필드 타입을 가져옵니다.

Prototype

`string GetFieldType()`

Example

```
_DEBUG("DataFieldType " + This.GetFieldType());
```

See Also

[DataFieldTypeConst](#)

IsCalculatedField

수식 필드인지 여부를 가져옵니다.

Prototype

`bool IsCalculatedField()`

Example

```
_DEBUG("IsCalculatedField " + This.IsCalculatedField());
```

IsDataSetValueNull

해당 필드 값이 null 값인지 여부를 가져옵니다.

Prototype

`bool IsDataSetValueNull(string FieldPath)`

Parameters

FieldPath	필드 경로 값을 가져올 필드 종류에 따라 다르게 설정
-----------	----------------------------------

Reference

필드 경로에 자기 자신을 설정할 경우 에러가 발생합니다.

값을 가져올 필드 종류에 따라 필드 경로 설정 방법과 참고사항은 다음과 같습니다.

- **데이터 필드**

"필드 이름" 또는 "데이터셋 이름.필드 이름" 형태로 설정합니다.

데이터셋 이름을 지정할 경우에는 같은 ODI 내에 있는 데이터셋으로 해당 수식 필드가 추가된 데이터셋 또는 해당 수식 필드가 추가된 데이터셋의 마스터셋인 경우에만 지정할 수 있습니다.

- **사용자 지정 패러미터 필드**

"OZParam.필드 이름" 형태로 설정합니다.

필드 경로에 마스터셋의 필드 또는 사용자 지정 패러미터 필드를 설정할 경우 쿼리 디자이너에서는 빈 값으로 표시되나, 리포트 뷰어에서는 정상적으로 표시됩니다.

Example 1

```
//데이터 필드(필드 이름 Price)
This.IsDataSetValueNull("Price");
```

Example 2

```
//데이터 필드(마스터셋 이름 Code, 필드 이름 Name)
This.IsDataSetValueNull("Code.Name");
```

Example 3

```
//사용자 지정 패러미터 필드(사용자 지정 패러미터 필드 이름 TotalAmount)
This.IsDataSetValueNull("OZParam.TotalAmount");
```

IsDecrypt

필드의 복호화 여부를 가져옵니다.

Prototype

bool IsDecrypt()

Example

```
_DEBUG("IsDecrypt " + This.IsDecrypt());
```

IsNull

데이터 값이 null 값인지 여부를 가져옵니다.

Prototype

bool IsNull()

Example

```
_DEBUG("IsNull " + This.IsNull());
```

SetFieldData

수식 필드의 값을 설정합니다.

Prototype

void SetFieldData(var FieldValue)

Parameters

FieldValue	설정할 값 수식 필드의 필드 타입에 맞게 설정
------------	------------------------------

Example 1

```
//수식 필드의 필드 타입이 Varchar인 경우  
This.SetFieldData("포시에스");
```

Example 2

```
//수식 필드의 필드 타입이 Integer인 경우  
This.SetFieldData(100);
```

Event

Name	Description
GetFieldData	수식 필드의 값을 사용할 때 발생하는 이벤트입니다.

GetFieldData

수식 필드의 값을 사용할 때 발생하는 이벤트입니다.

Prototype

void GetFieldData()

Reference

수식 필드가 바인딩될 때 데이터셋의 행 별로 한 번씩 발생합니다.

수식 필드의 값은 SetFieldData() 함수를 이용하거나 일반적인 자바 스크립트 함수인 return 함수로 설정하기 바랍니다.

아무런 값을 설정하지 않은 경우 null 값으로 설정됩니다.

Global 함수인 _DEBUG(), _INFO(), _ERROR() 함수와 This 객체를 사용할 수 있으며, _DEBUG(), _INFO(), _ERROR() 함수 사용 시 리포트 뷰어 콘솔의 각 로그 레벨의 탭에 설정한 내용이 표시됩니다.

Example 1

```
//SetFieldData() 함수를 이용, 수식 필드의 필드 타입이 Varchar인 경우
This.SetFieldData("포시에스");
```

Example 2

```
//return 함수를 이용, 수식 필드의 필드 타입이 Varchar인 경우
return "포시에스";
```

OZDataField

- Method

Method

Name	Description
GetFieldName	필드 이름을 가져옵니다.
GetFieldType	필드 타입을 가져옵니다.
GetValue	필드 값을 가져옵니다.
IsCalculatedField	수식 필드인지 여부를 가져옵니다.
IsDecrypt	필드의 복호화 여부를 가져옵니다.
IsNull	데이터 값이 null 값인지 여부를 가져옵니다.

GetFieldName

필드 이름을 가져옵니다.

Prototype

string GetFieldName()

Example

```
_DEBUG("FieldName " + This.GetFieldName());
```

See Also

[이름](#)

GetFieldType

필드 타입을 가져옵니다.

Prototype

string GetFieldType()

Example

```
_DEBUG("DataFieldType " + This.GetFieldType());
```

See Also

[타입](#), [DataFieldTypeConst](#)

GetValue

필드 값을 가져옵니다.

Prototype

var GetValue()

Example

```
_DEBUG("Value " + This.GetValue());
```

IsCalculatedField

수식 필드인지 여부를 가져옵니다.

Prototype

bool IsCalculatedField()

Example

```
_DEBUG("IsCalculatedField " + This.IsCalculatedField());
```

See Also

[DataFieldTypeConst](#)

IsDecrypt

필드의 복호화 여부를 가져옵니다.

Prototype

bool IsDecrypt()

Example

```
_DEBUG("IsDecrypt " + This.IsDecrypt());
```

See Also

[복호화](#)

IsNull

데이터 값이 null 값인지 여부를 가져옵니다.

Prototype

bool IsNull()

Example

```
_DEBUG("IsNull " + This.IsNull());
```

IsNull

- [Property](#)
- [Method](#)

Property

Name	Description
BaseXPathCondition	XPath 조건문
UseOZScript	오즈 스크립트 사용 여부

BaseXPathCondition

XPath 조건문

Prototype

[string](#) [BaseXPathCondition](#)

Example

```
var ds = This.GetDataSet();
_DEBUG("BaseXPathCondition " + ds.BaseXPathCondition);
```

UseOZScript

오즈 스크립트 사용 여부

Prototype

[bool](#) [UseOZScript](#)

Example

```
var ds = This.GetDataSet();
_DEBUG("UseOZScript " + ds.UseOZScript);
```

Method

Name	Description
GetFieldName	필드 이름을 가져옵니다.

GetFieldType	필드 타입을 가져옵니다.
GetXPath	XPath를 가져옵니다.
IsCalculatedField	수식 필드인지 여부를 가져옵니다.
IsDecrypt	필드의 복호화 여부를 가져옵니다.
IsNull	데이터 값이 null 값인지 여부를 가져옵니다.

GetFieldName

필드 [이름](#)을 가져옵니다.

Prototype

[string GetFieldName\(\)](#)

Example

```
_DEBUG("FieldName " + This.GetFieldName());
```

See Also

[이름](#)

GetFieldType

필드 타입을 가져옵니다.

Prototype

[string GetFieldType\(\)](#)

Example

```
_DEBUG("DataFieldType " + This.GetFieldType());
```

See Also

[타입](#), [DataFieldTypeConst](#)

GetXPath

XPath를 가져옵니다.

Prototype

string GetXPath()

Example

```
_DEBUG("XPath " + This.GetXPath());
```

See Also

[XPath](#)

IsCalculatedField

수식 필드인지 여부를 가져옵니다.

Prototype

bool IsCalculatedField()

Example

```
_DEBUG("IsCalculatedField " + This.IsCalculatedField());
```

IsDecrypt

필드의 복호화 여부를 가져옵니다.

Prototype

bool IsDecrypt()

Example

```
_DEBUG("IsDecrypt " + This.IsDecrypt());
```

See Also

[복호화](#)

IsNull

데이터 값이 null 값인지 여부를 가져옵니다.

Prototype

bool IsNull()

Example

```
_DEBUG("IsNull " + This.IsNull());
```

GeneralScript

- [Global](#)
- [OZBinary](#)
- [OZCreator](#)
- [OZDBConnection](#)
- [OZJSPreparedStatement](#)
- [OZJSResultSet](#)
- [OZJSResultSetMetaData](#)
- [OZJSStatement](#)
- [OZMail](#)
- [OZRepository](#)
- [OZProcedureParameter](#)

Global

- [Method](#)

Method

Name	Description
_DEBUG	Debug 레벨 로그
_ERROR	Error 레벨 로그
_INFO	Info 레벨 로그

[_DEBUG](#)

Debug 레벨 로그

Prototype

void _DEBUG(string Message)

Parameters

Message	로그 내용
---------	-------

Example

```
_DEBUG("Debug Message");
```

_ERROR

Error 레벨 로그

Prototype

void _ERROR(string Message)

Parameters

Message	로그 내용
---------	-------

Example

```
_ERROR("Error Message");
```

_INFO

Info 레벨 로그

Prototype

void _INFO(string Message)

Parameters

Message	로그 내용
---------	-------

Example

```
_INFO("Info Message");
```

OZBinary

- [Method](#)

Method

Nameq	Description
Base64decode	Base64로 인코딩된 문자열을 Base64로 디코딩하여 오즈 바이너리 객체로 설정합니다.
Base64encode	오즈 바이너리 객체에 있는 데이터를 Base64로 인코딩한 후 인코딩된 문자열을 가져옵니다.
GetName	오즈 바이너리 객체 이름을 가져옵니다.
ReadFromFile	지정한 파일에 있는 데이터를 오즈 바이너리 객체로 설정합니다.
ReadFromHTTP	지정한 URL의 데이터를 오즈 바이너리 객체로 설정합니다.
SaveToFile	오즈 바이너리 객체에 저장된 데이터를 파일로 저장합니다.
SetName	오즈 바이너리 객체 이름을 설정합니다.
ToString	오즈 바이너리 객체에 저장된 데이터를 문자열로 변환하여 가져옵니다.

Base64decode

Base64로 인코딩된 문자열을 Base64로 디코딩하여 오즈 바이너리 객체로 설정합니다.

Prototype

```
void Base64decode(string Text)
```

Parameters

Text	Base64로 인코딩된 문자열
------	------------------

Example

```
var binary = OZCreator.getBuiltInObject("OZBinary");
//var binary = OZCreator.getBuiltInObject(OZBuiltInObject.BINARY);
```

```
binary.Base64decode("xde9usau");  
var encodetext = binary.Base64encode();
```

Base64encode

오즈 바이너리 객체에 있는 데이터를 Base64로 인코딩한 후 인코딩된 문자열을 가져옵니다.

Prototype

string Base64encode()

Example

```
var binary = OZCreator.getBuiltInObject("OZBinary");  
//var binary = OZCreator.getBuiltInObject(OZBuiltInObject.BINARY);  
binary.Base64decode("xde9usau");  
var encodetext = binary.Base64encode();
```

GetName

오즈 바이너리 객체 이름을 가져옵니다.

Prototype

string GetName()

Example

```
var binary = OZCreator.getBuiltInObject("OZBinary");  
//var binary = OZCreator.getBuiltInObject(OZBuiltInObject.BINARY);  
var name = binary.GetName();
```

ReadFromFile

지정한 파일에 있는 데이터를 오즈 바이너리 객체로 설정합니다.

Prototype

void ReadFromFile(string FileName)

Parameters

FileName	파일 이름
----------	-------

Example

```
var binary = OZCreator.getBuiltInObject("OZBinary");  
//var binary = OZCreator.getBuiltInObject(OZBuiltInObject.BINARY);  
binary.ReadFromFile("C:/sample.odi");
```

ReadFromHTTP

지정한 URL의 데이터를 오즈 바이너리 객체로 설정합니다.

Prototype

void ReadFromHTTP(string URL)

Parameters

URL	URL
-----	-----

Example

```
var binary = OZCreator.getBuiltInObject("OZBinary");  
//var binary = OZCreator.getBuiltInObject(OZBuiltInObject.BINARY);  
binary.ReadFromFile("http://127.0.0.1/sample.odi");
```

SaveToFile

오즈 바이너리 객체에 저장된 데이터를 파일로 저장합니다.

Prototype

void SaveToFile(string FileName)

Parameters

FileName	파일 이름
----------	-------

Example

```
var binary = OZCreator.getBuiltInObject("OZBinary");  
//var binary = OZCreator.getBuiltInObject(OZBuiltInObject.BINARY);  
binary.SaveToFile("C:/sample.odi");
```

SetName

오즈 바이너리 객체 이름을 설정합니다.

Prototype

void SetName(string Name)

Parameters

Name	오즈 바이너리 객체 이름
------	---------------

Example

```
var binary = OZCreator.getBuiltInObject("OZBinary");  
//var binary = OZCreator.getBuiltInObject(OZBuiltInObject.BINARY);  
binary.SetName("BINARY");
```

ToString

오즈 바이너리 객체에 저장된 데이터를 문자열로 변환하여 가져옵니다.

Prototype

string ToString()

Example

```
var binary = OZCreator.getBuiltInObject("OZBinary");  
//var binary = OZCreator.getBuiltInObject(OZBuiltInObject.BINARY);  
var Str = binary.ToString();
```

OZCreator

- [Method](#)

Method

Name	Description
GetBuiltInObject	오즈 빌트인 객체를 가져옵니다.

GetBuiltInObject

오즈 빌트인 객체를 가져옵니다.

Prototype

object GetBuiltInObject(string BuiltInObjectName)

object GetBuiltInObject(object BuiltInObject)

Parameters

BuiltInObjectName	오즈 빌트인 객체 이름
BuiltInObject	오즈 빌트인 객체

Example 1

```
var binary = OZCreator.getBuiltInObject("OZBinary");
var name = binary.GetName();
```

Example 2

```
var binary = OZCreator.getBuiltInObject(OZBuiltInObject.BINARY);
var name = binary.GetName();
```

OZDBConnection

- [Method](#)

Method

Name	Description
AddItemValue	데이터베이스 연결 속성을 추가합니다.
Close	커넥션을 종료합니다.
Commit	커밋을 실행합니다.
Connect	커넥션을 생성합니다.
CreateStatement	쿼리 실행을 위한 실행문을 가져옵니다.
GetAutoCommit	커넥션의 자동 커밋 여부를 가져옵니다.

IsClosed	커넥션이 데이터베이스에 연결된 상태인지 여부를 가져옵니다.
PrepareStatement	지정한 쿼리문으로 OZPrepareStatement 객체를 생성한 후 생성된 OZPrepareStatement 객체를 가져옵니다.
Rollback	롤백합니다.
SetAlias	오즈 서버 커넥션 풀의 앨리어스 이름을 설정합니다.
SetAutoCommit	현재 커넥션의 자동 커밋 여부를 설정합니다.
SetVendor	DBMS 벤더의 이름을 설정합니다.

AddItemValue

데이터베이스 연결 속성을 추가합니다.

Prototype

void AddItemValue(string ItemKey, string ItemValue)

Parameters

ItemKey	아이템 키
ItemValue	아이템 값

Reference

dbconfig.xml에 정의된 벤더 아이템의 key와 value는 모두 추가하여야 하며, 이 때 user와 password 키는 반드시 추가하여야 합니다.

Example

```
function DataBaseConnectionSample(){
    var con = OZCreator.GetBuiltInObject(OZBuiltInObject.DBCONNECTION);
    con.SetVendor("mssql");
    con.AddItemValue("serverAddress","127.0.0.1");
    con.AddItemValue("portNo","1433");
    con.AddItemValue("dbName", "oz");
    con.AddItemValue("user", "sa");
    con.AddItemValue("password", "forcs");
    var isOK = con.Connect(false);
    _DEBUG("isOK="+isOK);
}
```

```
con.SetAutoCommit(false);
var stmt;
try {
    stmt = con.CreateStatement();
    var query = "insert into table1 (field1) values('value1')";
    stmt.ExecuteUpdate(query);
    con.Commit();
} catch(e) {
    con.Rollback();
    _ERROR("error="+e);
} finally {
    stmt.Close();
    con.Close();
}
_DEBUG("OK");
}
```

Close

커넥션을 종료합니다.

Prototype

void Close()

Example

```
function DataBaseConnectionSample(){
    var con = OZCreator.GetBuiltInObject(OZBuiltInObject.DBCONNECTION);
    con.SetVendor("mssql");
    con.AddItemValue("serverAddress","127.0.0.1");
    con.AddItemValue("portNo","1433");
    con.AddItemValue("dbName", "oz");
    con.AddItemValue("user", "sa");
    con.AddItemValue("password", "forcs");
    var isOK = con.Connect(false);
    _DEBUG("isOK="+isOK);
    con.SetAutoCommit(false);
    var stmt;
    try {
        stmt = con.CreateStatement();
```

```
        var query = "insert into table1 (field1) values('value1')";
        stmt.ExecuteUpdate(query);
        con.Commit();
    } catch(e) {
        con.Rollback();
        _ERROR("error="+e);
    } finally {
        stmt.Close();
        con.Close();
    }
    _DEBUG("OK");
}
```

Commit

커밋을 실행합니다.

Prototype

void **Commit()**

Example

```
function DataBaseConnectionSample(){
    var con = OZCreator.GetBuiltInObject(OZBuiltInObject.DBCONNECTION);
    con.SetVendor("mssql");
    con.AddItemValue("serverAddress","127.0.0.1");
    con.AddItemValue("portNo","1433");
    con.AddItemValue("dbName", "oz");
    con.AddItemValue("user", "sa");
    con.AddItemValue("password", "forcs");
    var isOK = con.Connect(false);
    _DEBUG("isOK="+isOK);
    con.SetAutoCommit(false);
    var stmt;
    try {
        stmt = con.CreateStatement();
        var query = "insert into table1 (field1) values('value1')";
        stmt.ExecuteUpdate(query);
        con.Commit();
    } catch(e) {
```

```

        con.Rollback();
        _ERROR("error="+e);
    } finally {
        stmt.Close();
        con.Close();
    }
    _DEBUG("OK");
}

```

Connect

커넥션을 생성합니다.

Prototype

bool Connect(bool FromServer)

Parameters

FromServer	커넥션을 생성할 서버가 오즈 서버인지 여부
------------	-------------------------

Reference

FromServer 값을 true로 설정하면 오즈 서버의 커넥션 풀로부터 커넥션을 생성하고, false로 설정하면 DBMS에 접속하여 커넥션을 생성합니다.

Example

```

function DataBaseConnectionSample(){
    var con = OZCreator.GetBuiltInObject(OZBuiltInObject.DBCONNECTION);
    con.SetVendor("mssql");
    con.AddItemValue("serverAddress","127.0.0.1");
    con.AddItemValue("portNo","1433");
    con.AddItemValue("dbName", "oz");
    con.AddItemValue("user", "sa");
    con.AddItemValue("password", "forcs");
    var isOK = con.Connect(false);
    _DEBUG("isOK="+isOK);
    con.SetAutoCommit(false);
    var stmt;
    try {
        stmt = con.CreateStatement();
    }
}

```

```

    var query = "insert into table1 (field1) values('value1')";
    stmt.ExecuteUpdate(query);
    con.Commit();
} catch(e) {
    con.Rollback();
    _ERROR("error="+e);
} finally {
    stmt.Close();
    con.Close();
}
_DEBUG("OK");
}

```

CreateStatement

쿼리 실행을 위한 실행문을 가져옵니다.

Prototype

object CreateStatement()

public OZJSStatement CreateStatement(integer resultSetType, integer
resultSetConcurrency, [integer resultSetHoldability]) throws OZSQLException

Parameters

resultSetType	결과셋 타입 OZJSResultSetConst.TYPE_FORWARD_ONLY, OZJSResultSetConst.TYPE_SCROLL_INSENSITIVE, OZJSResultSetConst.TYPE_SCROLL_SENSITIVE 중 하나의 값으로 설정
resultSetConcurrency	결과셋 Concurrency OZJSResultSetConst.CONCUR_READ_ONLY, OZJSResultSetConst.CONC UR_UPDATABLE 중 하나의 값으로 설정
resultSetHoldability	결과셋 Holdability OZJSResultSetConst.HOLD_CURSORS_OVER_COMMIT, OZJSResultSetConst.CLOSE_CURSORS_AT_COMMIT 중 하나의 값으로 설 정

Reference

그룹 관리자 정보를 가져온 경우 true가 리턴되고, 실패한 경우 false가 리턴됩니다.

Example 1

```
function DataBaseConnectionSample(){
    var con = OZCreator.GetBuiltInObject(OZBuiltInObject.DBCONNECTION);
    con.SetVendor("mssql");
    con.AddItemValue("serverAddress","127.0.0.1");
    con.AddItemValue("portNo","1433");
    con.AddItemValue("dbName", "oz");
    con.AddItemValue("user", "sa");
    con.AddItemValue("password", "forcs");
    var isOK = con.Connect(false);
    _DEBUG("isOK="+isOK);
    con.SetAutoCommit(false);
    var stmt;
    try {
        stmt = con.CreateStatement();
        var query = "insert into table1 (field1) values('value1')";
        stmt.ExecuteUpdate(query);
        con.Commit();
    } catch(e) {
        con.Rollback();
        _ERROR("error="+e);
    } finally {
        stmt.Close();
        con.Close();
    }
    _DEBUG("OK");
}
```

Example 2

```
function prepareStatementSample(){
    var con = OZCreator.GetBuiltInObject(OZBuiltInObject.DBCONNECTION);
    con.SetVendor("mssql2008");
    con.AddItemValue("serverAddress","192.168.100.172");
    con.AddItemValue("portNo","1433");
    con.AddItemValue("dbName", "pubs");
    con.AddItemValue("user", "sa");
    con.AddItemValue("password", "1234");
}
```

```
var isOK = con.Connect(false);
_DEBUG("isOK="+isOK);
con.SetAutoCommit(false);
var stmt;
try {
    var query = "select * from sales";
    stmt = con.PrepareStatement(query, OZJSResultSetConst.TYPE_SCROLL_SENSITIVE,
OZJSResultSetConst.CONCUR_UPDATABLE, OZJSResultSetConst.CLOSE_CURSORS_AT_COMMIT);
    var rs = stmt.executeQuery();
    var meta = rs.GetMetaData();
    for(var i=0; i<meta.GetColumnCount(); i++) {
        _DEBUG(i+'= column Name='+meta.GetColumnName(i+1)+'
type='+meta.GetColumnType(i+1));
    }
    while(rs.Next()) {
        var a = "";
        for(var i=0; i<meta.GetColumnCount(); i++) {
            a = a + rs.GetObject(meta.GetColumnName(i+1)) + ", ";
        }
        _DEBUG(a);
    }
    rs.BeforeFirst();
    while(rs.Next()) {
        var a = "";
        for(var i=0; i<meta.GetColumnCount(); i++) {
            a = a + rs.GetObject(meta.GetColumnName(i+1)) + ", ";
        }
        _DEBUG(a);
    }
    rs.Close();
} catch(e) {
    _ERROR("error="+e);
} finally {
    //stmt.Close();
    //con.Close();
}
_DEBUG("OK");
}
```

See Also

GetAutoCommit

커넥션의 자동 커밋 여부를 가져옵니다.

Prototype

bool GetAutoCommit()

Example

```
function DataBaseConnectionSample(){
    var con = OZCreator.GetBuiltInObject(OZBuiltInObject.DBCONNECTION);
    con.SetVendor("mssql");
    con.AddItemValue("serverAddress","127.0.0.1");
    con.AddItemValue("portNo","1433");
    con.AddItemValue("dbName", "oz");
    con.AddItemValue("user", "sa");
    con.AddItemValue("password", "forcs");
    var isOK = con.Connect(false);
    _DEBUG("isOK="+isOK);
    con.SetAutoCommit(false);
    var AutoCommit = con.GetAutoCommit();
    _DEBUG("AutoCommit="+AutoCommit);
    var stmt;
    try {
        stmt = con.CreateStatement();
        var query = "insert into table1 (field1) values('value1')";
        stmt.ExecuteUpdate(query);
        con.Commit();
    } catch(e) {
        con.Rollback();
        _ERROR("error="+e);
    } finally {
        stmt.Close();
        con.Close();
    }
    _DEBUG("OK");
}
```

IsClosed

커넥션이 데이터베이스에 연결된 상태인지 여부를 가져옵니다.

Prototype

bool IsClosed()

Example

```
function DataBaseConnectionSample(){
    var con = OZCreator.GetBuiltInObject(OZBuiltInObject.DBCONNECTION);
    con.SetVendor("mssql");
    con.AddItemValue("serverAddress","127.0.0.1");
    con.AddItemValue("portNo","1433");
    con.AddItemValue("dbName", "oz");
    con.AddItemValue("user", "sa");
    con.AddItemValue("password", "forcs");
    var isOK = con.Connect(false);
    _DEBUG("isOK="+isOK);
    var IsClosed = con.IsClosed();
    _DEBUG("IsClosed="+IsClosed);
    con.SetAutoCommit(false);
    var stmt;
    try {
        stmt = con.CreateStatement();
        var query = "insert into table1 (field1) values('value1')";
        stmt.ExecuteUpdate(query);
        con.Commit();
    } catch(e) {
        con.Rollback();
        _ERROR("error="+e);
    } finally {
        stmt.Close();
        con.Close();
    }
    _DEBUG("OK");
}
```

PrepareStatement

지정한 쿼리문으로 OZPreparedStatement 객체를 생성한 후 생성된 OZPreparedStatement 객체를 가져옵니다.

Prototype

object PreparedStatement(string QueryString)

public OZJSPreparedStatement PreparedStatement(string QueryString, integer resultSetType, integer resultSetConcurrency, [integer resultSetHoldability]) throws OZSQLException

Parameters

QueryString	쿼리문
resultSetType	로컬 파일 전체 경로 또는 로컬 파일 전체 경로를 가진 배열 OZJResultSetConst.TYPE_FORWARD_ONLY, OZJResultSetConst.TYPE_SCROLL_INSENSITIVE, OZJResultSetConst.TYPE_SCROLL_SENSITIVE 중 하나의 값으로 설정
resultSetConcurrency	결과셋 Concurrency OZJResultSetConst.CONCUR_READ_ONLY, OZJResultSetConst.CONCUR_UPDATABLE 중 하나의 값으로 설정
resultSetHoldability	결과셋 Holdability OZJResultSetConst.HOLD_CURSORS_OVER_COMMIT, OZJResultSetConst.CLOSE_CURSORS_AT_COMMIT 중 하나의 값으로 설정

Reference

그룹 관리자 정보를 가져온 경우 true가 리턴되고, 실패한 경우 false가 리턴됩니다.

Example 1

```
function preparedStatementSample(){
    var con = OZCreator.GetBuiltInObject(OZBuiltInObject.DBCONNECTION);
    con.SetAlias("ozcar");
    con.Connect(true);
    var pstmt = con.PrepareStatement("select * from car where Maker = ? ");
    pstmt.SetParameter(1, "HYUNDAI");
    var rs = pstmt.ExecuteQuery();
}
```

```
var meta = rs.GetMetaData();
for(var i=0; i<meta.GetColumnCount(); i++) {
    _DEBUG(i+"= column Name="+meta.GetColumnName(i+1)+"
type="+meta.GetColumnType(i+1));
}
while(rs.Next()) {
    var data = "";
    for(var i=0; i<meta.GetColumnCount(); i++) {
        data = data + rs.GetString(i+1) + ",    ";
    }
    _DEBUG(data);
}
rs.Close();
pstmt.Close();
}
```

Example 2

```
function prepareStatementSample(){
    var con = OZCreator.GetBuiltInObject(OZBuiltInObject.DBCONNECTION);
    con.SetVendor("mssql2008");
    con.AddItemValue("serverAddress","192.168.100.172");
    con.AddItemValue("portNo","1433");
    con.AddItemValue("dbName", "pubs");
    con.AddItemValue("user", "sa");
    con.AddItemValue("password", "1234");
    var isOK = con.Connect(false);
    _DEBUG("isOK="+isOK);
    con.SetAutoCommit(false);
    var stmt;
    try {
        var query = "select * from sales";
        stmt = con.PrepareStatement(query, OZJSResultSetConst.TYPE_SCROLL_SENSITIVE,
OZJSResultSetConst.CONCUR_UPDATABLE, OZJSResultSetConst.CLOSE_CURSORS_AT_COMMIT);
        var rs = stmt.ExecuteQuery();
        var meta = rs.GetMetaData();
        for(var i=0; i<meta.GetColumnCount(); i++) {
            _DEBUG(i+'= column Name='+meta.GetColumnName(i+1)+'
type='+meta.GetColumnType(i+1));
        }
        while(rs.Next()) {
```

```
        var a = "";
        for(var i=0; i<meta.GetColumnCount(); i++) {
            a = a + rs.GetObject(meta.GetColumnName(i+1)) + ", ";
        }
        _DEBUG(a);
    }
    rs.BeforeFirst();
    while(rs.Next()) {
        var a = "";
        for(var i=0; i<meta.GetColumnCount(); i++) {
            a = a + rs.GetObject(meta.GetColumnName(i+1)) + ", ";
        }
        _DEBUG(a);
    }
    rs.Close();
} catch(e) {
    _ERROR("error="+e);
} finally {
    //stmt.Close();
    //con.Close();
}
_DEBUG("OK");
}
```

See Also

[OZPrepareStatement](#), [OZJSResultSetConst](#)

Rollback

롤백합니다.

Prototype

void Rollback()

Example

```
function DataBaseConnectionSample() {
    var con = OZCreator.GetBuiltInObject(OZBuiltInObject.DBCONNECTION);
    con.SetVendor("mssql");
    con.AddItemValue("serverAddress","127.0.0.1");
}
```

```
con.AddItemValue("portNo", "1433");
con.AddItemValue("dbName", "oz");
con.AddItemValue("user", "sa");
con.AddItemValue("password", "forcs");
var isOK = con.Connect(false);
_DEBUG("isOK="+isOK);
con.SetAutoCommit(false);
var stmt;
try {
    stmt = con.CreateStatement();
    var query = "insert into table1 (field1) values('value1')";
    stmt.ExecuteUpdate(query);
    con.Commit();
} catch(e) {
    con.Rollback();
    _ERROR("error="+e);
} finally {
    stmt.Close();
    con.Close();
}
_DEBUG("OK");
}
```

SetAlias

오즈 서버 커넥션 풀의 앨리어스 이름을 설정합니다.

Prototype

void SetAlias(string AliasName)

Parameters

AliasName	앨리어스 이름
-----------	---------

Reference

오즈 서버로부터 커넥션을 가져올 때 사용합니다.

Example

```
function prepareStatementSample(){
```

```

var con = OZCreator.GetBuiltInObject(OZBuiltInObject.DBCONNECTION);
con.SetAlias("ozcar");
con.Connect(true);
var pstmt = con.PrepareStatement("select * from car where Maker = ? ");
pstmt.SetParameter(1, "HYUNDAI");
var rs = pstmt.ExecuteQuery();
var meta = rs.GetMetaData();
for(var i=0; i<meta.GetColumnCount(); i++) {
    _DEBUG(i+"= column Name="+meta.GetColumnName(i+1)+"type="+meta.GetColumnType(i+1));
}
while(rs.Next()) {
    var data = "";
    for(var i=0; i<meta.GetColumnCount(); i++) {
        data = data + rs.GetString(i+1) + ",    ";
    }
    _DEBUG(data);
}
rs.Close();
pstmt.Close();
}

```

SetAutoCommit

현재 커넥션의 자동 커밋 여부를 설정합니다.

Prototype

void SetAutoCommit(bool AutoCommit)

Parameters

AutoCommit	자동 커밋 여부
------------	----------

Example

```

function DataBaseConnectionSample(){
    var con = OZCreator.GetBuiltInObject(OZBuiltInObject.DBCONNECTION);
    con.SetVendor("mssql");
    con.AddItemValue("serverAddress","127.0.0.1");
    con.AddItemValue("portNo","1433");
    con.AddItemValue("dbName", "oz");
}

```

```

con.AddItemValue("user", "sa");
con.AddItemValue("password", "forcs");
var isOK = con.Connect(false);
_DEBUG("isOK="+isOK);
con.SetAutoCommit(false);
var stmt;
try {
    stmt = con.CreateStatement();
    var query = "insert into table1 (field1) values('value1')";
    stmt.ExecuteUpdate(query);
    con.Commit();
} catch(e) {
    con.Rollback();
    _ERROR("error="+e);
} finally {
    stmt.Close();
    con.Close();
}
_DEBUG("OK");
}

```

SetVendor

DBMS 벤더의 이름을 설정합니다.

Prototype

void SetVendor(string VendorName)

Parameters

VendorName	벤더 이름
------------	-------

Example

```

function DataBaseConnectionSample(){
    var con = OZCreator.GetBuiltInObject(OZBuiltInObject.DBCONNECTION);
    con.SetVendor("mssql");
    con.AddItemValue("serverAddress","127.0.0.1");
    con.AddItemValue("portNo","1433");
    con.AddItemValue("dbName", "oz");
}

```

```

con.AddItemValue("user", "sa");
con.AddItemValue("password", "forcs");
var isOK = con.Connect(false);
_DEBUG("isOK="+isOK);
con.SetAutoCommit(false);
var stmt;
try {
    stmt = con.CreateStatement();
    var query = "insert into table1 (field1) values('value1')";
    stmt.ExecuteUpdate(query);
    con.Commit();
} catch(e) {
    con.Rollback();
    _ERROR("error="+e);
} finally {
    stmt.Close();
    con.Close();
}
_DEBUG("OK");
}

```

OZJSPreparedStatement

- [Method](#)

Method

Name	Description
ClearParameters	패러미터 정보를 삭제합니다.
Close	OZJSPreparedStatement 객체를 종료합니다.
ExecuteQuery	쿼리문을 실행하고, 쿼리를 실행한 결과셋으로 OZJSResultSet 객체를 가져옵니다.
ExecuteUpdate	쿼리문을 실행하고, 테이블에서 변경된 행 수를 가져옵니다.
SetFetchRow	JDBC 전송 ROW 측, 결과 셋 반환 시 한꺼번에 가져올 수 있는 행 수를 설정합니다.
SetMaxRows	최대 행 수를 설정합니다.
SetParameter	패러미터 값을 설정합니다.

SetQueryTimeout	쿼리 타임아웃 시간을 설정합니다.
---------------------------------	--------------------

ClearParameters

패러미터 정보를 삭제합니다.

Prototype

void ClearParameters()

Example

```
function OZJSPreparedStatementSample(){
    var con = OZCreator.GetBuiltInObject(OZBuiltInObject.DBCONNECTION);
    con.SetAlias("ozcar");
    con.Connect(true);
    var pstmt = con.PrepareStatement("select * from car where Maker = ? ");
    pstmt.SetParameter(1, "HYUNDAI");
    var rs = pstmt.ExecuteQuery();
    var meta = rs.GetMetaData();
    for(var i=0; i<meta.GetColumnCount(); i++) {
        _DEBUG(i+"= column Name="+meta.GetColumnName(i+1)+"type="+meta.GetColumnType(i+1));
    }
    while(rs.Next()) {
        var data = "";
        for(var i=0; i<meta.GetColumnCount(); i++) {
            data = data + rs.GetString(i+1) + ", ";
        }
        _DEBUG(data);
    }
    pstmt.ClearParameters();
    rs.Close();
    pstmt.Close();
}
```

Close

OZJSPreparedStatement 객체를 종료합니다.

Prototype

void Close()

Example

```
function OZJSPreparedStatementSample(){
    var con = OZCreator.GetBuiltInObject(OZBuiltInObject.DBCONNECTION);
    con.SetAlias("ozcar");
    con.Connect(true);
    var pstmt = con.PrepareStatement("select * from car where Maker = ? ");
    pstmt.SetParameter(1, "HYUNDAI");
    var rs = pstmt.ExecuteQuery();
    var meta = rs.GetMetaData();
    for(var i=0; i<meta.GetColumnCount(); i++) {
        _DEBUG(i+"= column Name="+meta.GetColumnName(i+1)+"type="+meta.GetColumnType(i+1));
    }
    while(rs.Next()) {
        var data = "";
        for(var i=0; i<meta.GetColumnCount(); i++) {
            data = data + rs.GetString(i+1) + ", ";
        }
        _DEBUG(data);
    }
    rs.Close();
    pstmt.Close();
}
```

ExecuteQuery

쿼리문을 실행하고, 쿼리를 실행한 결과셋으로 OZJSResultSet 객체를 가져옵니다.

Prototype

object ExecuteQuery()

Example

```
function OZJSPreparedStatementSample(){
    var con = OZCreator.GetBuiltInObject(OZBuiltInObject.DBCONNECTION);
    con.SetAlias("ozcar");
    con.Connect(true);
    var pstmt = con.PrepareStatement("select * from car where Maker = ? ");
    pstmt.SetParameter(1, "HYUNDAI");
```

```
var rs = pstmt.executeQuery();
var meta = rs.GetMetaData();
for(var i=0; i<meta.GetColumnCount(); i++) {
    _DEBUG(i+"= column Name="+meta.GetColumnName(i+1)+"type="+meta.GetColumnType(i+1));
}
while(rs.Next()) {
    var data = "";
    for(var i=0; i<meta.GetColumnCount(); i++) {
        data = data + rs.GetString(i+1) + ",    ";
    }
    _DEBUG(data);
}
rs.Close();
pstmt.Close();
}
```

See Also

[OZJSResultSet](#)

ExecuteUpdate

쿼리문을 실행하고, 테이블에서 변경된 행 수를 가져옵니다.

Prototype

integer ExecuteUpdate()

Example

```
function OZJSPreparedStatementSample(){
    var con = OZCreator.GetBuiltInObject(OZBuiltInObject.DBCONNECTION);
    con.SetAlias("ozcar");
    con.Connect(true);
    var pstmt = con.PrepareStatement("select * from car where Maker = ? ");
    pstmt.SetParameter(1, "HYUNDAI");
    var RowCnt = pstmt.ExecuteUpdate();
    _DEBUG(RowCnt);
    pstmt.Close();
}
```

SetFetchRow

JDBC 전송 ROW 측, 결과 셋 반환 시 한꺼번에 가져올 수 있는 행 수를 설정합니다.

Prototype

void SetFetchRow(integer Fetchrows)

Parameters

Fetchrows	전송할 행 수
-----------	---------

Example

```
function OZJSPreparedStatementSample() {
    var con = OZCreator.GetBuiltInObject(OZBuiltInObject.DBCONNECTION);
    con.SetVendor("odbc");
    con.AddItemValue("dsn","ozedu");
    con.Connect(false);
    var pstmt = con.PrepareStatement("select * from Product");
    try {
        pstmt.SetFetchRow (9);
    } catch(Exception) {}
    var rs = pstmt.ExecuteQuery();
    var meta = rs.GetMetaData();
    for(var i=0; i<meta.GetColumnCount(); i++) {
        _DEBUG(i+"= preparedstatement column Name="+meta.GetColumnName(i+1)
+"type="+meta.GetColumnType(i+1));
    }
    while(rs.Next()) {
        var data = "";
        for(var i=0; i<meta.GetColumnCount(); i++) {
            data = data + rs.GetString(i+1) + ",    ";
        }
        _DEBUG(data);
    }
    rs.Close();
    pstmt.Close();
    con.Close();
}
```

SetMaxRows

최대 행 수를 설정합니다.

Prototype

void SetMaxRows(integer MaxRowCount)

Parameters

MaxRowCount	최대 행 수
-------------	--------

Example

```
function OZJSPreparedStatementSample(){
    var con = OZCreator.GetBuiltInObject(OZBuiltInObject.DBCONNECTION);
    con.SetAlias("ozcar");
    con.Connect(true);
    var pstmt = con.PrepareStatement("select * from car where Maker = ? ");
    pstmt.SetMaxRows(2);
    pstmt.SetParameter(1, "HYUNDAI");
    var rs = pstmt.ExecuteQuery();
    var meta = rs.GetMetaData();
    for(var i=0; i<meta.GetColumnCount(); i++) {
        _DEBUG(i+"= column Name="+meta.GetColumnName(i+1)+"type="+meta.GetColumnType(i+1));
    }
    while(rs.Next()) {
        var data = "";
        for(var i=0; i<meta.GetColumnCount(); i++) {
            data = data + rs.GetString(i+1) + ",    ";
        }
        _DEBUG(data);
    }
    rs.Close();
    pstmt.Close();
}
```

SetParameter

패러미터 값을 설정합니다.

Prototype

void SetParameter(integer ParameterIndex, object ValueObject)

void SetParameter(string ParameterIndex, object ValueObject)**Parameters**

ParameterIndex	패러미터 인덱스 (One-Based Index)
ValueObject	값을 설정할 객체

Reference

오즈 서버 타입이 자바인 경우 ParameterIndex를 integer 타입으로 선언하고, 닷넷인 경우 string 타입으로 선언하여 사용합니다.

Example 1

```
function OZJSPreparedStatementSample(){
    //OZ Server Type = Java
    var con = OZCreator.GetBuiltInObject(OZBuiltInObject.DBCONNECTION);
    con.SetAlias("ozcar");
    con.Connect(true);
    var pstmt = con.PrepareStatement("select * from car where Maker = ? ");
    pstmt.SetMaxRows(2);
    pstmt.SetParameter(1, "HYUNDAI");
    var rs = pstmt.ExecuteQuery();
    var meta = rs.GetMetaData();
    for(var i=0; i<meta.GetColumnCount(); i++) {
        _DEBUG(i+"= column Name="+meta.GetColumnName(i+1)+"type="+meta.GetColumnType(i+1));
    }
    while(rs.Next()) {
        var data = "";
        for(var i=0; i<meta.GetColumnCount(); i++) {
            data = data + rs.GetString(i+1) + ",  ";
        }
        _DEBUG(data);
    }
    rs.Close();
    pstmt.Close();
}
```

Example 2

```
function OZJSPreparedStatementSample(){
```

```
//OZ Server Type = .Net
var con = OZCreator.GetBuiltInObject(OZBuiltInObject.DBCONNECTION);
con.SetAlias("ozcar");
con.Connect(true);
var pstmt = con.PrepareStatement("select * from car where Maker = @1");
pstmt.SetMaxRows(2);
pstmt.SetParameter("@1", "HYUNDAI");
var rs = pstmt.ExecuteQuery();
var meta = rs.GetMetaData();
for(var i=0; i<meta.GetColumnCount(); i++) {
    _DEBUG(i+"= column Name="+meta.GetColumnName(i+1)+"type="+meta.GetColumnType(i+1));
}
while(rs.Next()) {
    var data = "";
    for(var i=0; i<meta.GetColumnCount(); i++) {
        data = data + rs.GetString(i+1) + ",    ";
    }
    _DEBUG(data);
}
rs.Close();
pstmt.Close();
}
```

SetQueryTimeout

쿼리 타임아웃 시간을 설정합니다.

Prototype

void SetQueryTimeout(integer QueryTimeout)

Parameters

QueryTimeout	타임 아웃 시간 (단위 초)
--------------	-----------------

Example

```
function OZJSPreparedStatementSample(){
    var con = OZCreator.GetBuiltInObject(OZBuiltInObject.DBCONNECTION);
    con.SetAlias("ozcar");
    con.Connect(true);
```

```

var pstmt = con.PrepareStatement("select * from car where Maker = ? ");
pstmt.SetQueryTimeout(1000);
pstmt.SetMaxRows(2);
pstmt.SetParameter(1, "HYUNDAI");
var rs = pstmt.executeQuery();
var meta = rs.GetMetaData();
for(var i=0; i<meta.GetColumnCount(); i++) {
    _DEBUG(i+"= column Name="+meta.GetColumnName(i+1)+"type="+meta.GetColumnType(i+1));
}
while(rs.Next()) {
    var data = "";
    for(var i=0; i<meta.GetColumnCount(); i++) {
        data = data + rs.GetString(i+1) + ",    ";
    }
    _DEBUG(data);
}
rs.Close();
pstmt.Close();
}

```

OZJSResultSet

- [Method](#)

Method

Name	Description
Close	OZJSResultSet 객체를 종료합니다.
GetObject	필드 객체를 가져옵니다.
GetString	데이터 값을 가져옵니다.
GetMetaData	ResultSetMetaData 자바 객체를 가져옵니다.
Next	결과셋의 커서를 다음 행으로 이동시킵니다.

Close

OZJSResultSet 객체를 종료합니다.

Prototype

void Close()

Example

```
function OZJSResultSetSample(){
    var con = OZCreator.GetBuiltInObject(OZBuiltInObject.DBCONNECTION);
    con.SetAlias("ozcar");
    con.Connect(true);
    var stmt = con.CreateStatement();
    var rs = stmt.ExecuteQuery("select * from part");
    var meta = rs.GetMetaData();
    for(var i=0; i<meta.GetColumnCount(); i++) {
        _DEBUG(i+"= column Name="+meta.GetColumnName(i+1)+" type="+meta.GetColumnType(i+1));
    }
    while(rs.Next()) {
        var data = "";
        for(var i=0; i<meta.GetColumnCount(); i++) {
            data = data + rs.GetString(i+1) + ",";
        }
        _DEBUG(data);
    }
    rs.Close();
    stmt.Close();
}
```

GetObject

필드 객체를 가져옵니다.

Prototype

object GetObject(string FieldName)

object GetObject(integer FieldIndex)

Parameters

FieldName	필드 이름
FieldIndex	필드 인덱스

Reference

필드 인덱스를 지정하여 필드 객체를 가져올 경우 필드 타입에 따라 리턴되는 자바 스크립트 객체는 다음 표와 같으며, 자바스크립트 객체는 아래 표를 참고하시기 바랍니다.

JAVA SQL FieldType	Java Script Object
TINYINT	integer
OZSMALLINT	integer
INTEGER	integer
BIGINT	Number
REAL	Number
FLOAT	Number
DOUBLE	Number
BIT	Boolean
CHAR	String
VARCHAR	String
LONGVARCHAR	String
NUMERIC	Number
DECIMAL	Number
DATE	Date
TIME	Date
TIMESTAMP	Date
BINARY	OZBinary
VARBINARY	OZBinary
LONGBINARY	OZBinary
BLOB	OZBinary
CLOB	String

Example 1

```
function OZJSResultSetSample(){
    var con = OZCreator.GetBuiltInObject(OZBuiltInObject.DBCONNECTION);
    con.SetAlias("ozcar");
    con.Connect(true);
    var stmt = con.CreateStatement();
    var rs = stmt.ExecuteQuery("select * from car");
    while(rs.Next()) {
        var data = "";
        data = data + rs.GetObject("Field1");
        _DEBUG(data);
    }
    rs.Close();
    stmt.Close();
}
```

Example 2

```
function OZJSResultSetSample(){
    var con = OZCreator.GetBuiltInObject(OZBuiltInObject.DBCONNECTION);
    con.SetVendor("mssql");
    con.AddItemValue("serverAddress","127.0.0.1");
    con.AddItemValue("portNo","1433");
    con.AddItemValue("dbName", "oz");
    con.AddItemValue("user", "sa");
    con.AddItemValue("password", "forcs");
    var isOK = con.Connect(false);
    _DEBUG("isOK="+isOK);
    var stmt;
    var rs;
    try {
        stmt = con.CreateStatement();
        var query = "select * from car";
        rs = stmt.ExecuteQuery(query);
        var meta = rs.GetMetaData();
        for(var i=0; i<meta.GetColumnCount(); i++) {
            _DEBUG(i+'= column Name='+meta.GetColumnName(i+1)+" type="+meta.GetColumnType(i+1));
        }
        while(rs.Next()) {
            for(var i=0; i<meta.GetColumnCount(); i++) {
```

```
switch(meta.GetColumnType(i+1)){
    case 1: // java.sql.Types.CHAR:
    case 12: //java.sql.Types.VARCHAR:
    case -1: // java.sql.Types.LONGVARCHAR:
    case 2005: //java.sql.Types.CLOB
        var svalue = rs.GetObject(i+1);
        _DEBUG("String="+svalue);
        break;
    case -6: //java.sql.Types.TINYINT:
    case 5: // java.sql.Types.SMALLINT:
    case 4: //java.sql.Types.INTEGER:
        var ivalue = rs.GetObject(i+1);
        _DEBUG("integer="+ivalue);
        break;
    case -5: //java.sql.Types.BIGINT:
    case 7: //java.sql.Types.REAL:
    case 6: //java.sql.Types.FLOAT:
    case 8: //java.sql.Types.DOUBLE:
    case 2: //java.sql.Types.NUMERIC:
    case 3: //java.sql.Types.DECIMAL:
        var value4 = rs.GetObject(i+1);
        _DEBUG("Number="+value4);
        break;
    case -7: //java.sql.Types.BIT:
        var boolv = rs.GetObject(i+1);
        _DEBUG("Boolean="+boolv.toString());
        break;
    case 91: //java.sql.Types.DATE:
    case 92: //java.sql.Types.TIME:
    case 93: //java.sql.Types.TIMESTAMP:
        var datev = rs.GetObject(i+1);
        _DEBUG("Date="+datev.toGMTString());
        break;
    case -2: //java.sql.Types.BINARY:
    case -3: //java.sql.Types.VARBINARY:
    case -4: //java.sql.Types.LONGVARBINARY:
    case 2004: //java.sql.Types.BLOB
        var ozb = rs.GetObject(i+1);
        if (ozb == null) {
            _DEBUG("OZBinary is null.");
        }
    }
```

```

        }else {
            ozb.SaveToFile("c:/test.jpg");
            _DEBUG("OZBinary="+ozb);
        }
        break;
    default:
        var s = rs.GetObject(i+1);
        _DEBUG("String==="+svalue);
        break;
    }
}
}
} catch(e) {
    _ERROR("error="+e);
} finally {
    rs.Close();
    stmt.Close();
    con.Close();
}
_DEBUG("OK");
}

```

GetString

데이터 값을 가져옵니다.

Prototype

string GetString(string fieldName)

string GetString(integer FieldIndex)

Parameters

FieldName	필드 이름
FieldIndex	필드 인덱스

Example 1

```

function OZJSResultSetSample(){
    var con = OZCreator.GetBuiltInObject(OZBuiltInObject.DBCONNECTION);
    con.SetAlias("ozcar");
}

```

```
con.Connect(true);
var stmt = con.CreateStatement();
var rs = stmt.ExecuteQuery("select * from car");
while(rs.Next()) {
    var data = "";
    data = data + rs.GetString("Field1");
    _DEBUG(data);
}
rs.Close();
stmt.Close();
}
```

Example 2

```
function OZJSResultSetSample(){
    var con = OZCreator.GetBuiltInObject(OZBuiltInObject.DBCONNECTION);
    con.SetAlias("ozcar");
    con.Connect(true);
    var stmt = con.CreateStatement();
    var rs = stmt.ExecuteQuery("select * from car");
    var meta = rs.GetMetaData();
    for(var i=0; i<meta.GetColumnCount(); i++) {
        _DEBUG(i+"= column Name="+meta.GetColumnName(i+1)+" type="+meta.GetColumnType(i+1));
    }
    while(rs.Next()) {
        var data = "";
        for(var i=0; i<meta.GetColumnCount(); i++) {
            data = data + rs.GetString(i+1) + ",";
        }
        _DEBUG(data);
    }
    rs.Close();
    stmt.Close();
}
```

GetMetaData

ResultSetMetaData 자바 객체를 가져옵니다.

Prototype

object GetMetaData()

Reference

자바 객체에 대한 설명은 썬 개발자 네트워크 홈페이지에서 자바 API 관련 부분을 참고하시기 바랍니다.

Example

```
function OZJSResultSetSample(){
    var con = OZCreator.GetBuiltInObject(OZBuiltInObject.DBCONNECTION);
    con.SetAlias("ozcar");
    con.Connect(true);
    var stmt = con.CreateStatement();
    var rs = stmt.ExecuteQuery("select * from part");
    var meta = rs.GetMetaData();
    for(var i=0; i<meta.GetColumnCount(); i++) {
        _DEBUG(i+"= column Name="+meta.GetColumnName(i+1)+"
type="+meta.GetColumnType(i+1));
    }
    while(rs.Next()) {
        var data = "";
        for(var i=0; i<meta.GetColumnCount(); i++) {
            data = data + rs.GetString(i+1) + ",";
        }
        _DEBUG(data);
    }
    rs.Close();
    stmt.Close();
}
```

Next

결과셋의 커서를 다음 행으로 이동시킵니다.

Prototype

bool Next()

Example

```
function OZJSResultSetSample(){
    var con = OZCreator.GetBuiltInObject(OZBuiltInObject.DBCONNECTION);
```

```

con.SetAlias("ozcar");
con.Connect(true);
var stmt = con.CreateStatement();
var rs = stmt.ExecuteQuery("select * from part");
var meta = rs.GetMetaData();
for(var i=0; i<meta.GetColumnCount(); i++) {
    _DEBUG(i+"= column Name="+meta.GetColumnName(i+1)+"
type="+meta.GetColumnType(i+1));
}
while(rs.Next()) {
    var data = "";
    for(var i=0; i<meta.GetColumnCount(); i++) {
        data = data + rs.GetString(i+1) + ",";
    }
    _DEBUG(data);
}
rs.Close();
stmt.Close();
}

```

OZJSResultSetMetaData

- [Method](#)

Method

Name	Description
GetColumnCount	ResultSet의 필드 개수를 가져옵니다.
GetColumnName	필드 이름을 가져옵니다.
GetColumnType	필드 타입을 가져옵니다.

GetColumnCount

ResultSet의 필드 개수를 가져옵니다.

Prototype

integer [GetColumnCount](#)(integer FieldIndex)

Parameters

FieldIndex	필드 인덱스 (One-Based Index)
------------	--------------------------

Example

```
function OZJSResultSetMetaData(){
  var con = OZCreator.GetBuiltInObject(OZBuiltInObject.DBCONNECTION);
  con.SetAlias("ozcar");
  con.Connect(true);
  var stmt = con.createStatement();
  var rs = stmt.executeQuery("select * from car");
  var meta = rs.GetMetaData();
  _DEBUG("GetColumnCount: "+meta.GetColumnCount());
  rs.Close();
  stmt.Close();
}
```

GetColumnName

필드 이름을 가져옵니다.

Prototype

integer GetColumnName(integer FieldIndex)

Parameters

FieldIndex	필드 인덱스 (One-Based Index)
------------	--------------------------

Example

```
function OZJSResultSetMetaData(){
  var con = OZCreator.GetBuiltInObject(OZBuiltInObject.DBCONNECTION);
  con.SetAlias("ozcar");
  con.Connect(true);
  var stmt = con.createStatement();
  var rs = stmt.executeQuery("select * from car");
  var meta = rs.GetMetaData();
  _DEBUG("GetColumnName: "+meta.GetColumnName(1));
  rs.Close();
}
```

```
stmt.Close();  
}
```

GetColumnType

필드 타입을 가져옵니다.

Prototype

string GetColumnType(**integer** FieldIndex)

Parameters

FieldIndex	필드 인덱스 (One-Based Index)
------------	--------------------------

Example

```
function OZJSResultSetMetaData(){  
    var con = OZCreator.GetBuiltInObject(OZBuiltInObject.DBCONNECTION);  
    con.SetAlias("ozcar");  
    con.Connect(true);  
    var stmt = con.CreateStatement();  
    var rs = stmt.ExecuteQuery("select * from car");  
    var meta = rs.GetMetaData();  
    _DEBUG("GetColumnType: "+meta.GetColumnType(1));  
    rs.Close();  
    stmt.Close();  
}
```

See Also

[DataFieldTypeConst](#)

OZJSStatement

- [Method](#)

Method

Name	Description
------	-------------

Close	커넥션을 종료합니다.
ExecuteQuery	쿼리문을 실행하고, 쿼리를 실행한 결과셋으로 OZJSResultSet 객체를 가져옵니다.
ExecuteUpdate	쿼리문을 실행하고, 테이블에서 변경된 행 수를 가져옵니다.
SetFetchRow	JDBC 전송 ROW 측, 결과 셋 반환 시 한꺼번에 가져올 수 있는 행 수를 설정합니다.
SetMaxRows	최대 행 수를 설정합니다.
SetQueryTimeout	쿼리 타임아웃 시간을 설정합니다.

Close

커넥션을 종료합니다.

Prototype

void Close()

Example

```
function OZJSStatementSample(){
    var con = OZCreator.GetBuiltInObject(OZBuiltInObject.DBCONNECTION);
    con.SetAlias("ozcar");
    con.Connect(true);
    var stmt = con.createStatement();
    var rs = stmt.executeQuery("select * from part");
    stmt.setQueryTimeout(1000);
    stmt.setMaxRows(2);
    var meta = rs.getMetaData();
    for(var i=0; i<meta.getColumnCount(); i++) {
        _DEBUG(i+"= column Name="+meta.getColumnName(i+1)+"
type="+meta.getColumnType(i+1));
    }
    while(rs.next()) {
        var data = "";
        for(var i=0; i<meta.getColumnCount(); i++) {
            data = data + rs.getString(i+1) + ", ";
        }
        _DEBUG(data);
    }
    rs.close();
}
```

```
stmt.Close();
}
```

ExecuteQuery

쿼리문을 실행하고, 쿼리를 실행한 결과셋으로 OZJSResultSet 객체를 가져옵니다.

Prototype

object ExecuteQuery(string QueryString)

Parameters

QueryString	쿼리문
-------------	-----

Example

```
function OZJSStatementSample(){
    var con = OZCreator.GetBuiltInObject(OZBuiltInObject.DBCONNECTION);
    con.SetAlias("ozcar");
    con.Connect(true);
    var stmt = con.CreateStatement();
    var rs = stmt.ExecuteQuery("select * from part");
    stmt.SetQueryTimeout(1000);
    stmt.SetMaxRows(2);
    var meta = rs.GetMetaData();
    for(var i=0; i<meta.GetColumnCount(); i++) {
        _DEBUG(i+"= column Name="+meta.GetColumnName(i+1)+"
type="+meta.GetColumnType(i+1));
    }
    while(rs.Next()) {
        var data = "";
        for(var i=0; i<meta.GetColumnCount(); i++) {
            data = data + rs.GetString(i+1) + ", ";
        }
        _DEBUG(data);
    }
    rs.Close();
    stmt.Close();
}
```

See Also

[OZResultSet](#)

ExecuteUpdate

쿼리문을 실행하고, 테이블에서 변경된 행 수를 가져옵니다.

Prototype

integer ExecuteUpdate(string QueryString)

Parameters

QueryString	쿼리문
-------------	-----

Example

```
function DataBaseConnectionSample() {
    var con = OZCreator.GetBuiltInObject(OZBuiltInObject.DBCONNECTION);
    con.SetVendor("mssql");
    con.AddItemValue("serverAddress","127.0.0.1");
    con.AddItemValue("portNo","1433");
    con.AddItemValue("dbName", "oz");
    con.AddItemValue("user", "sa");
    con.AddItemValue("password", "forcs");
    var isOK = con.Connect(false);
    _DEBUG("isOK="+isOK);
    con.SetAutoCommit(false);
    var stmt;
    try {
        stmt = con.CreateStatement();
        var query = "insert into table1 (field1) values('value1')";
        stmt.ExecuteUpdate(query);
        con.Commit();
    } catch(e) {
        con.Rollback();
        _ERROR("error="+e);
    } finally {
        stmt.Close();
        con.Close();
    }
}
```

```

    }
    _DEBUG("OK");
}

```

SetFetchRow

JDBC 전송 ROW 즉, 결과 셋 반환 시 한꺼번에 가져올 수 있는 행 수를 설정합니다.

Prototype

void SetFetchRow(integer Fetchrows)

Parameters

Fetchrows	전송할 행 수
-----------	---------

Example

```

function OZJSSstatementSample() {
    var con = OZCreator.GetBuiltInObject(OZBuiltInObject.DBCONNECTION);
    con.SetVendor("odbc");
    con.AddItemValue("dsn","ozedu");
    con.Connect(false);
    var stmt = con.CreateStatement();
    try {
        stmt.SetFetchRow (2);
    } catch(Exception) {}
    var rs = stmt.ExecuteQuery("select * from Product");
    var meta = rs.GetMetaData();
    for(var i=0; i<meta.GetColumnCount(); i++) {
        _DEBUG(i+"= statement column Name="+meta.GetColumnName(i+1)
+"type="+meta.GetColumnType(i+1));
    }
    while(rs.Next()) {
        var data = "";
        for(var i=0; i<meta.GetColumnCount(); i++) {
            data = data + rs.GetString(i+1) + ",    ";
        }
        _DEBUG(data);
    }
    rs.Close();
}

```

```
stmt.Close();
con.Close();
}
```

SetMaxRows

최대 행 수를 설정합니다.

Prototype

void SetMaxRows(integer MaxRowCount)

Parameters

MaxRowCount	최대 행 수
-------------	--------

Example

```
function OZJSStatementSample(){
    var con = OZCreator.GetBuiltInObject(OZBuiltInObject.DBCONNECTION);
    con.SetAlias("ozcar");
    con.Connect(true);
    var stmt = con.CreateStatement();
    var rs = stmt.ExecuteQuery("select * from part");
    stmt.SetQueryTimeout(1000);
    stmt.SetMaxRows(2);
    var meta = rs.GetMetaData();
    for(var i=0; i<meta.GetColumnCount(); i++) {
        _DEBUG(i+"= column Name="+meta.GetColumnName(i+1)+"
type="+meta.GetColumnType(i+1));
    }
    while(rs.Next()) {
        var data = "";
        for(var i=0; i<meta.GetColumnCount(); i++) {
            data = data + rs.GetString(i+1) + ", ";
        }
        _DEBUG(data);
    }
    rs.Close();
    stmt.Close();
}
```

SetQueryTimeout

쿼리 타임아웃 시간을 설정합니다.

Prototype

void SetQueryTimeout(integer TimeoutValue)

Parameters

TimeoutValue	타임 아웃 시간(단위 초)
--------------	----------------

Example

```
function OZJSStatementSample(){
    var con = OZCreator.GetBuiltInObject(OZBuiltInObject.DBCONNECTION);
    con.SetAlias("ozcar");
    con.Connect(true);
    var stmt = con.CreateStatement();
    var rs = stmt.ExecuteQuery("select * from part");
    stmt.SetQueryTimeout(1000);
    stmt.SetMaxRows(2);
    var meta = rs.GetMetaData();
    for(var i=0; i<meta.GetColumnCount(); i++) {
        _DEBUG(i+"= column Name="+meta.GetColumnName(i+1)+"
type="+meta.GetColumnType(i+1));
    }
    while(rs.Next()) {
        var data = "";
        for(var i=0; i<meta.GetColumnCount(); i++) {
            data = data + rs.GetString(i+1) + ", ";
        }
        _DEBUG(data);
    }
    rs.Close();
    stmt.Close();
}
```

OZMail

- [Method](#)

Method

Name	Description
AddAttachmentPart	메일에 파일을 첨부합니다.
EnableSSL	SSL 사용 여부를 설정합니다.
PrefixSubjectMessage	메일 제목 앞에 사용할 접두어를 설정합니다.
Send	비동기 방식으로 메일을 전송합니다.
SendRetryCount	메일 전송 실패 시 재전송 횟수를 설정합니다.
SendRetryPeriodTime	메일 전송 실패 시 재전송 주기를 설정합니다.
SendSync	동기 방식으로 메일을 전송합니다.
SetAlias	오즈 서버 mail.properties의 앨리어스 이름을 설정합니다.
SetContext	메일 내용을 설정합니다.
SetFrom	발신자 메일 주소를 설정합니다.
SetRecipients	수신자 메일 주소를 설정합니다.
SetSmtpID	SMTP 인증이 필요한 메일 서버를 이용할 때 인증할 사용자 ID를 설정합니다.
SetSmtpPort	SMTP 인증이 필요한 메일 서버를 이용할 때 인증할 SMTP 서버 포트 번호를 설정합니다.
SetSmtpPW	SMTP 인증이 필요한 메일 서버를 이용할 때 인증할 사용자 ID의 패스워드를 설정합니다.
SetSmtpServer	SMTP 인증이 필요한 메일 서버를 이용할 때 SMTP 서버 URL을 설정합니다.
SetSubject	메일 제목을 설정합니다.

AddAttachmentPart

메일에 파일을 첨부합니다.

Prototype

```
void AddAttachmentPart(string FileFullPath, [string AttachFileName])
```

```
void AddAttachmentPart(object File, string AttachFileName)
```

Parameters

FileFullPath	첨부할 파일 전체 경로
AttachFileName	첨부 파일에 표시할 이름 값을 설정하지 않을 경우 FileFullPath에서 설정한 파일 이름이 적용됨
File	첨부할 파일 오즈 바이너리 객체로 설정하여야 함

Example

```
function mailTestAlias(){
    try {
        var mail = OZCreator.GetBuiltInObject(OZBuiltInObject.MAIL);
        mail.SetAlias("ozmail");
        mail.SetRecipients(RecipientTypeConst.TO, "oz@forcs.com");
        mail.SetRecipients(RecipientTypeConst.CC, "forcs@forcs.com");
        mail.SetSubject("Mail Subject");
        mail.SetContext("Mail Context");
        mail.AddAttachmentPart("d:/sample.xls");
        mail.Send();
        mail.Release();
    } catch(e) {
        _ERROR(e);
    }
}
```

See Also

[OZBinary](#)

EnableSSL

SSL 사용 여부를 설정합니다.

Prototype

void EnableSSL(bool IsSSL)

Parameters

IsSSL	SSL 사용 여부
-------	-----------

Reference

연동되는 서버가 자바 서버일 경우 jre1.4.2_12 버전 이상에서만 동작합니다.

Example

```
function mailTestAlias(){
    try {
        var mail = OZCreator.GetBuiltInObject(OZBuiltInObject.MAIL);
        mail.SetAlias("ozmail");
        mail.SetRecipients(RecipientTypeConst.TO, "oz@forcs.com");
        mail.SetRecipients(RecipientTypeConst.CC, "forcs@forcs.com");
        mail.SetSubject("Mail Subject");
        mail.SetContext("Mail Context");
        mail.AddAttachmentPart("d:/sample.xls");
        mail.Send();
        mail.Release();
    } catch(e) {
        _ERROR(e);
    }
}
```

PrefixSubjectMessage

메일 제목 앞에 사용할 접두어를 설정합니다.

Prototype

void PrefixSubjectMessage(string PrefixSubject)

Parameters

PrefixSubject	메일 제목 접두어
---------------	-----------

Example

```
function mailTest() {
    try {
        var mail = OZCreator.GetBuiltInObject(OZBuiltInObject.MAIL);
        mail.SetSmtpServer("mail.forcs.com");
```

```
mail.SetSmtppPort(25);
mail.SetSmtppID("oz@forcs.com");
mail.SetSmtppPW("UserPassword");
mail.SetFrom("oz@forcs.com","Forcs");
mail.SendRetryCount(1);
mail.SendRetryPeriodTime(10);
mail.PrefixSubjectMessage("[Forcs]");
mail.SetRecipients(RecipientTypeConst.T0, "user1@forcs.com, user2@forcs.com");
mail.SetRecipients(RecipientTypeConst.BCC, "user3@forcs.com");
mail.SetSubject("Mail Subject");
mail.SetContext("Mail Context");
mail.AddAttachmentPart("d:/sample.xls");
mail.Send();
} catch(e) {
    _ERROR(e);
}
}
```

Send

비동기 방식으로 메일을 전송합니다.

Prototype

void Send()

Example

```
function mailTestAlias(){
    try {
        var mail = OZCreator.GetBuiltInObject(OZBuiltInObject.MAIL);
        mail.SetAlias("ozmail");
        mail.SetRecipients(RecipientTypeConst.T0, "oz@forcs.com");
        mail.SetRecipients(RecipientTypeConst.CC, "forcs@forcs.com");
        mail.SetSubject("Mail Subject");
        mail.SetContext("Mail Context");
        mail.AddAttachmentPart("d:/sample.xls");
        mail.Send();
        mail.Release();
    } catch(e) {
        _ERROR(e);
    }
}
```

```
}
}
```

SendRetryCount

메일 전송 실패 시 재전송 횟수를 설정합니다.

Prototype

void SendRetryCount(integer RetryCount)

Parameters

RetryCount	재전송 횟수
------------	--------

Example

```
function mailTest() {
    try {
        var mail = OZCreator.GetBuiltInObject(OZBuiltInObject.MAIL);
        mail.SetSmtpServer("mail.forcs.com");
        mail.SetSmtpPort(25);
        mail.SetSmtpID("oz@forcs.com");
        mail.SetSmtpPW("UserPassword");
        mail.SetFrom("oz@forcs.com", "Forcs");
        mail.SendRetryCount(1);
        mail.SendRetryPeriodTime(10);
        mail.PrefixSubjectMessage("[Forcs]");
        mail.SetRecipients(RecipientTypeConst.T0, "user1@forcs.com, user2@forcs.com");
        mail.SetRecipients(RecipientTypeConst.BCC, "user3@forcs.com");
        mail.SetSubject("Mail Subject");
        mail.SetContext("Mail Context");
        mail.AddAttachmentPart("d:/sample.xls");
        mail.Send();
    } catch(e) {
        _ERROR(e);
    }
}
```

SendRetryPeriodTimea

메일 전송 실패 시 재전송 주기를 설정합니다.

Prototype

void SendRetryPeriodTime(integer RetryPeriodTime)

Parameters

RetryPeriodTime	재전송 주기 (단위 초)
-----------------	---------------

Example

```
function mailTest() {  
    try {  
        var mail = OZCreator.GetBuiltInObject(OZBuiltInObject.MAIL);  
        mail.SetSmtpServer("mail.forcs.com");  
        mail.SetSmtpPort(25);  
        mail.SetSmtpID("oz@forcs.com");  
        mail.SetSmtpPW("UserPassword");  
        mail.SetFrom("oz@forcs.com", "Forcs");  
        mail.SendRetryCount(1);  
        mail.SendRetryPeriodTime(10);  
        mail.PrefixSubjectMessage("[Forcs]");  
        mail.SetRecipients(RecipientTypeConst.T0, "user1@forcs.com, user2@forcs.com");  
        mail.SetRecipients(RecipientTypeConst.BCC, "user3@forcs.com");  
        mail.SetSubject("Mail Subject");  
        mail.SetContext("Mail Context");  
        mail.AddAttachmentPart("d:/sample.xls");  
        mail.Send();  
    } catch(e) {  
        _ERROR(e);  
    }  
}
```

SendSync

동기 방식으로 메일을 전송합니다.

Prototype

bool SendSync()

Reference

메일 전송 성공 여부를 리턴합니다.

Example

```
function mailTestAlias(){
    try {
        var mail = OZCreator.GetBuiltInObject(OZBuiltInObject.MAIL);
        mail.SetAlias("ozmail");
        mail.SetRecipients(RecipientTypeConst.TO, "oz@forcs.com");
        mail.SetRecipients(RecipientTypeConst.CC, "forcs@forcs.com");
        mail.SetSubject("Mail Subject");
        mail.SetContext("Mail Context");
        mail.AddAttachmentPart("d:/sample.xls");
        var isResult = mail.SendSync();
        _DEBUG("isOK="+isResult);
        mail.Release();
    } catch(e) {
        _ERROR(e);
    }
}
```

SetAlias

오즈 서버 mail.properties의 앨리어스 이름을 설정합니다.

Prototype

void SetAlias(string AliasName)

Parameters

AliasName	앨리어스 이름
-----------	---------

Example

```
function mailTestAlias(){
    try {
        var mail = OZCreator.GetBuiltInObject(OZBuiltInObject.MAIL);
        mail.SetAlias("ozmail");
        mail.SetRecipients(RecipientTypeConst.TO, "oz@forcs.com");
```

```

        mail.SetRecipients(RecipientTypeConst.CC, "forcs@forcs.com");
        mail.SetSubject("Mail Subject");
        mail.SetContext("Mail Context");
        mail.AddAttachmentPart("d:/sample.xls");
        mail.Send();
        mail.Release();
    } catch(e) {
        _ERROR(e);
    }
}

```

SetContext

메일 내용을 설정합니다.

Prototype

void SetContext(string Context, [bool IsHTML])

Parameters

Context	메일 내용
IsHTML	메시지 형식을 HTML로 설정할지 여부 값을 설정하지 않을 경우 False로 적용됨

Reference

IsHTML을 False로 설정할 경우 메시지 형식이 일반 텍스트로 설정됩니다.

Example

```

function mailTestAlias(){
    try {
        var mail = OZCreator.GetBuiltInObject(OZBuiltInObject.MAIL);
        mail.SetAlias("ozmail");
        mail.SetRecipients(RecipientTypeConst.TO, "oz@forcs.com");
        mail.SetRecipients(RecipientTypeConst.CC, "forcs@forcs.com");
        mail.SetSubject("Mail Subject");
        mail.SetContext("Mail Context");
        mail.AddAttachmentPart("d:/sample.xls");
        mail.Send();
    }
}

```

```

        mail.Release();
    } catch(e) {
        _ERROR(e);
    }
}

```

SetFrom

발신자 메일 주소를 설정합니다.

Prototype

```
void SetFrom(string MailAddress, [string UserName])
```

Parameters

MailAddress	발신자 메일 주소
UserName	발신자 이름 값을 설정하지 않을 경우 MailAddress로 적용됨

Example

```

function mailTest() {
    try {
        var mail = OZCreator.GetBuiltInObject(OZBuiltInObject.MAIL);
        mail.SetSmtpServer("mail.forcs.com");
        mail.SetSmtpPort(25);
        mail.SetSmtpID("oz@forcs.com");
        mail.SetSmtpPW("UserPassword");
        mail.SetFrom("oz@forcs.com", "Forcs");
        mail.SendRetryCount(1);
        mail.SendRetryPeriodTime(10);
        mail.PrefixSubjectMessage("[Forcs]");
        mail.SetRecipients(RecipientTypeConst.T0, "user1@forcs.com, user2@forcs.com");
        mail.SetRecipients(RecipientTypeConst.BCC, "user3@forcs.com");
        mail.SetSubject("Mail Subject");
        mail.SetContext("Mail Context");
        mail.AddAttachmentPart("d:/sample.xls");
        mail.Send();
    } catch(e) {
        _ERROR(e);
    }
}

```

```
}
}
```

SetRecipients

수신자 메일 주소를 설정합니다.

Prototype

void SetRecipients([integer RecipientType], string MailAddress)

Parameters

RecipientType	수신자 타입. RecipientTypeConst 참고 값을 설정하지 않을 경우 "RecipientTypeConst.TO"로 적용됨
MailAddress	수신자 메일 주소

Example

```
function mailTest() {
    try {
        var mail = OZCreator.GetBuiltInObject(OZBuiltInObject.MAIL);
        mail.SetSmtpServer("mail.forcs.com");
        mail.SetSmtpPort(25);
        mail.SetSmtpID("oz@forcs.com");
        mail.SetSmtpPW("UserPassword");
        mail.SetFrom("oz@forcs.com", "Forcs");
        mail.SendRetryCount(1);
        mail.SendRetryPeriodTime(10);
        mail.PrefixSubjectMessage("[Forcs]");
        mail.SetRecipients(RecipientTypeConst.TO, "user1@forcs.com, user2@forcs.com");
        mail.SetRecipients(RecipientTypeConst.BCC, "user3@forcs.com");
        mail.SetSubject("Mail Subject");
        mail.SetContext("Mail Context");
        mail.AddAttachmentPart("d:/sample.xls");
        mail.Send();
    } catch(e) {
        _ERROR(e);
    }
}
```

See Also

[RecipientTypeConst](#)

SetSmtpID

SMTP 인증이 필요한 메일 서버를 이용할 때 인증할 사용자 ID를 설정합니다.

Prototype

void SetSmtpID(string SMTPUserID)

Parameters

SMTPUserID	SMTP 사용자 ID
------------	-------------

Example

```
function mailTest() {
    try {
        var mail = OZCreator.GetBuiltInObject(OZBuiltInObject.MAIL);
        mail.SetSmtpServer("mail.forcs.com");
        mail.SetSmtpPort(25);
        mail.SetSmtpID("oz@forcs.com");
        mail.SetSmtpPW("UserPassword");
        mail.SetFrom("oz@forcs.com", "Forcs");
        mail.SendRetryCount(1);
        mail.SendRetryPeriodTime(10);
        mail.PrefixSubjectMessage("[Forcs]");
        mail.SetRecipients(RecipientTypeConst.TO, "user1@forcs.com, user2@forcs.com");
        mail.SetRecipients(RecipientTypeConst.BCC, "user3@forcs.com");
        mail.SetSubject("Mail Subject");
        mail.SetContext("Mail Context");
        mail.AddAttachmentPart("d:/sample.xls");
        mail.Send();
    } catch(e) {
        _ERROR(e);
    }
}
```

SetSmtpPort

SMTP 인증이 필요한 메일 서버를 이용할 때 인증할 SMTP 서버 포트 번호를 설정합니다.

Prototype

void SetSmtpport(integer SMTPPort)

Parameters

SMTPPort	SMTP 서버 포트 번호
----------	---------------

Example

```
function mailTest() {  
    try {  
        var mail = OZCreator.GetBuiltInObject(OZBuiltInObject.MAIL);  
        mail.SetSmtppServer("mail.forcs.com");  
        mail.SetSmtppPort(25);  
        mail.SetSmtppID("oz@forcs.com");  
        mail.SetSmtppPW("UserPassword");  
        mail.SetFrom("oz@forcs.com", "Forcs");  
        mail.SendRetryCount(1);  
        mail.SendRetryPeriodTime(10);  
        mail.PrefixSubjectMessage("[Forcs]");  
        mail.SetRecipients(RecipientTypeConst.TO, "user1@forcs.com, user2@forcs.com");  
        mail.SetRecipients(RecipientTypeConst.BCC, "user3@forcs.com");  
        mail.SetSubject("Mail Subject");  
        mail.SetContext("Mail Context");  
        mail.AddAttachmentPart("d:/sample.xls");  
        mail.Send();  
    } catch(e) {  
        _ERROR(e);  
    }  
}
```

SetSmtppPW

SMTP 인증이 필요한 메일 서버를 이용할 때 인증할 사용자 ID의 패스워드를 설정합니다.

Prototype

void SetSmtppPW(string SMTPUserPassword)

Parameters

SMTPUserPassword	SMTP 사용자 ID의 패스워드
------------------	-------------------

Example

```
function mailTest() {  
    try {  
        var mail = OZCreator.GetBuiltInObject(OZBuiltInObject.MAIL);  
        mail.SetSmtpServer("mail.forcs.com");  
        mail.SetSmtpPort(25);  
        mail.SetSmtpID("oz@forcs.com");  
        mail.SetSmtpPW("UserPassword");  
        mail.SetFrom("oz@forcs.com", "Forcs");  
        mail.SendRetryCount(1);  
        mail.SendRetryPeriodTime(10);  
        mail.PrefixSubjectMessage("[Forcs]");  
        mail.SetRecipients(RecipientTypeConst.T0, "user1@forcs.com, user2@forcs.com");  
        mail.SetRecipients(RecipientTypeConst.BCC, "user3@forcs.com");  
        mail.SetSubject("Mail Subject");  
        mail.SetContext("Mail Context");  
        mail.AddAttachmentPart("d:/sample.xls");  
        mail.Send();  
    } catch(e) {  
        _ERROR(e);  
    }  
}
```

SetSmtpServer

SMTP 인증이 필요한 메일 서버를 이용할 때 SMTP 서버 URL을 설정합니다.

Prototype

void SetSmtpServer(string SMTPServerURL)

Parameters

SMTPServerURL	SMTP 서버 URL
---------------	-------------

Example

```
function mailTest() {
    try {
        var mail = OZCreator.GetBuiltInObject(OZBuiltInObject.MAIL);
        mail.SetSmtpServer("mail.forcs.com");
        mail.SetSmtpPort(25);
        mail.SetSmtpID("oz@forcs.com");
        mail.SetSmtpPW("UserPassword");
        mail.SetFrom("oz@forcs.com", "Forcs");
        mail.SendRetryCount(1);
        mail.SendRetryPeriodTime(10);
        mail.PrefixSubjectMessage("[Forcs]");
        mail.SetRecipients(RecipientTypeConst.TO, "user1@forcs.com, user2@forcs.com");
        mail.SetRecipients(RecipientTypeConst.BCC, "user3@forcs.com");
        mail.SetSubject("Mail Subject");
        mail.SetContext("Mail Context");
        mail.AddAttachmentPart("d:/sample.xls");
        mail.Send();
    } catch(e) {
        _ERROR(e);
    }
}
```

SetSubject

메일 제목을 설정합니다.

Prototype

void SetSubject(string Subject)

Parameters

Subject	메일 제목
---------	-------

Example

```
function mailTestAlias(){
    try {
        var mail = OZCreator.GetBuiltInObject(OZBuiltInObject.MAIL);
```

```

        mail.SetAlias("ozmail");
        mail.SetRecipients(RecipientTypeConst.TO, "oz@forcs.com");
        mail.SetRecipients(RecipientTypeConst.CC, "forcs@forcs.com");
        mail.SetSubject("Mail Subject");
        mail.SetContext("Mail Context");
        mail.AddAttachmentPart("d:/sample.xls");
        mail.Send();
        mail.Release();
    } catch(e) {
        _ERROR(e);
    }
}

```

OZRepository

- [Method](#)

Method

Name	Description
AddCategory	하위 카테고리를 추가합니다.
AddGroupAdmin	그룹 관리자를 추가합니다.
CheckIn	아이템을 체크인합니다.
CheckOut	아이템을 체크아웃합니다.
CheckUserPassword	사용자의 비밀번호가 맞는지 확인합니다.
ClearHistoryItem	아이템의 히스토리 중 특정 버전의 히스토리를 삭제합니다.
CreateGroup	새로운 그룹을 생성합니다.
CreateUser	새로운 사용자를 추가합니다.
DeleteCategory	카테고리를 삭제합니다.
DeleteGroup	그룹을 삭제합니다.
DeleteItem	지정한 아이템을 삭제합니다.
DeleteItems	지정한 아이템을 삭제합니다.

DeleteUser	사용자를 삭제합니다.
DownloadFile	아이템을 파일 또는 메모리로 다운로드합니다.
DownloadFileByVersion	특정 버전의 아이템을 다운로드합니다.
GetAccessibleTopNodePath	리파지토리 최상위 노드의 전체 경로를 가져옵니다.
GetCategoryListInCategory	카테고리에 포함된 모든 하위 카테고리의 이름과 전체 경로를 가져옵니다.
GetCheckOutFolder	아이템의 체크아웃 폴더를 가져옵니다.
GetCheckOutUserName	아이템의 체크아웃 사용자 이름을 가져옵니다.
GetErrorMessageByCode	에러 코드에 해당하는 에러 메시지를 가져옵니다.
GetGroupAdminListInGroup	그룹 관리자 정보를 가져옵니다.
GetGroupAuthToCategory	카테고리에 대한 그룹의 권한을 가져옵니다.
GetGroupAuthToItem	아이템에 대한 그룹의 권한을 가져옵니다.
GetGroupListAuthToCategory	카테고리에 대해 특정 권한 이상의 권한을 가진 모든 그룹 정보를 가져옵니다.
GetGroupListAuthToItem	아이템에 대해 특정 권한 이상의 권한을 가진 모든 그룹 정보를 가져옵니다.
GetGroupListInGroup	모든 하위 그룹의 정보를 가져옵니다.
GetGroupPathOfUser	사용자의 그룹 경로를 가져옵니다.
GetHistoryItemList	아이템의 히스토리 정보를 가져옵니다.
GetItemListInCategory	카테고리에 포함된 모든 아이템 이름을 가져옵니다.
GetLastDetailErrorMessage	마지막으로 발생한 에러의 에러 메시지를 가져옵니다.
GetLastErrorCode	마지막으로 발생한 에러의 에러 코드를 가져옵니다.
GetUserAuthToCategory	카테고리에 대한 사용자의 권한 정보를 가져옵니다.
GetUserAuthToItem	아이템에 대한 지정한 사용자의 권한 정보를 가져옵니다.
GetUserListAuthToCategory	카테고리에 대한 사용자의 권한이 특정 권한 이상인 모든 사용자 정보를 가져옵니다.

GetUserListAuthToItem	아이템에 대한 사용자의 권한이 특정 권한 이상인 모든 사용자 정보를 가져옵니다.
GetUserListInGroup	그룹에 추가된 모든 사용자 정보를 가져옵니다.
IsFileExist	파일의 존재 여부를 가져옵니다.
Login	스크립트가 실행 중인 서버의 리파지토리에 로그인합니다.
LoginToServer	지정한 서버에 로그인합니다.
Logout	서버에서 로그아웃합니다.
RemoveGroupAdmin	그룹 관리자를 해제합니다.
RenameCategory	카테고리 이름을 변경합니다.
RenameItem	아이템 이름을 변경합니다.
RollbackItem	아이템을 롤백합니다.
SearchItems	아이템을 검색합니다.
UndoCheckOut	아이템을 체크아웃 취소합니다.
UpdateCategoryGroupAuth	카테고리에 대한 그룹 권한을 변경합니다.
UpdateCategoryUserAuth	카테고리에 대한 사용자 권한을 변경합니다.
UpdateGroupName	그룹 이름을 변경합니다.
UpdateItemGroupAuth	아이템에 대한 그룹의 권한을 변경합니다.
UpdateItemUserAuth	아이템에 대한 사용자의 권한을 변경합니다.
UpdateUserDescription	사용자 설명을 변경합니다.
UpdateUserName	사용자 이름을 변경합니다.
UpdateUserPassword	사용자 패스워드를 변경합니다.
UploadFile	아이템을 업로드합니다.

AddCategory

하위 카테고리를 추가합니다.

Prototype

bool AddCategory(string CategoryFullPath, string NewCategoryName, string Comment)

Parameters

CategoryFullPath	카테고리 전체 경로
NewCategoryName	카테고리 이름
Comment	주석

Reference

하위 카테고리가 추가된 경우 true가 리턴되고, 실패한 경우 false가 리턴됩니다.

Example

```
function AddCategory(){
    var rep = OZCreator.GetBuiltInObject(OZBuiltInObject.REPOSITORY);
    var login = rep.Login("admin","admin");
    if (login) {
        var isOK = rep.AddCategory("/", "abc", "comment");
        _DEBUG("AddCategory="+isOK);
        rep.Logout();
    }
}
```

AddGroupAdmin

그룹 관리자를 추가합니다.

Prototype

bool AddGroupAdmin(string GroupPath, string AdminUserName)

Parameters

GroupPath	그룹 경로
AdminUserName	그룹 관리자 이름

Reference

그룹 관리자가 추가된 경우 true가 리턴되고, 실패한 경우 false가 리턴됩니다.

Example

```
function AddGroupAdmin(){
    var rep = OZCreator.GetBuiltInObject(OZBuiltInObject.REPOSITORY);
    var login = rep.Login("admin","admin");
    if (login) {
        var groupPath = "/";
        var adminUsername = "User1";
        var result = rep.AddGroupAdmin(groupPath, adminUsername);
        if (result == true) {
            _DEBUG("result="+result);
        } else {
            _DEBUG("result="+result+"["+rep.GetLastErrorCode()+"]
"+rep.GetErrorMessageByCode(rep.GetLastErrorCode()));
        }
        rep.Logout();
    }
}
```

CheckIn

아이템을 체크인합니다.

Prototype

bool CheckIn(string ItemLocalFullPath, string ItemServerFullPath, [bool KeepCheckout], string Comment)

bool CheckIn(array ItemLocalFullPaths, array ItemServerFullPaths, array ErrorCodes, [array KeepCheckouts], string Comment)

bool CheckIn(object Item, string ItemServerFullPath, [bool KeepCheckout], string Comment)

bool CheckIn(array Items, array ItemServerFullPaths, array ErrorCodes, [array KeepCheckouts], string Comment)

Parameters

ItemLocalFullPath	로컬 파일 전체 경로
ItemServerFullPath	서버 전체 경로

KeepCheckout	체크아웃 상태 유지 여부
Comment	주석
ItemLocalFullPaths	로컬 파일 전체 경로를 가진 배열
ItemServerFullPaths	서버 전체 경로를 가진 배열
ErrorCodes	에러 코드를 가져올 배열
KeepCheckouts	체크아웃 상태 유지 여부를 가진 배열
Item	아이템 오즈 바이너리 객체로 설정하여야 함
Items	아이템을 가진 배열 오즈 바이너리 객체를 가진 배열로 설정하여야 함

Reference

체크인이 성공한 경우 true가 리턴되고, 실패한 경우 false가 리턴됩니다.

에러가 발생한 아이템의 인덱스에 에러 코드 값을 리턴하며, 성공한 경우에는 에러 코드 값에 0을 리턴합니다.

Example 1

```
function checkIn(){
    var rep = OZCreator.GetBuiltInObject(OZBuiltInObject.REPOSITORY);
    var login = rep.Login("admin","admin");
    if (login) {
        var itemLocalFullPath = "c:/parameter_test.odi";
        var itemServerFullPath = "/abc/parameter_test.odi";
        var result = rep.CheckIn(itemLocalFullPath, itemServerFullPath, "comment");
        if (result == true) {
            _DEBUG("result="+result);
        } else {
            _DEBUG("result="+result+"["+rep.GetLastErrorCode()+"]
"+rep.GetErrorMessageByCode(rep.GetLastErrorCode()));
        }
        rep.Logout();
    }
}
```

Example 2

```
function checkIn(){
    var rep = OZCreator.GetBuiltInObject(OZBuiltInObject.REPOSITORY);
    var login = rep.Login("admin","admin");
    if (login) {
        var itemlocalFullPaths = new Array();
        itemlocalFullPaths[0] = "c:/parameter_test.odi";
        itemlocalFullPaths[1] = "c:/parameter_test.ozr";
        var serverFullPaths = new Array();
        serverFullPaths[0] = "/abc/parameter_test.odi";
        serverFullPaths[1] = "/abc/parameter_test.ozr";
        var errorCodes = new Array();
        var keepCheckouts = new Array();
        keepCheckouts[0] = true;
        keepCheckouts[1] = true;
        var result = rep.CheckIn(itemlocalFullPaths, serverFullPaths, errorCodes,
keepCheckouts, "comment");
        if (result == true) {
            _DEBUG("result="+result);
        } else {
            for (i=0; i < errorCodes.length; i++) {
                _DEBUG("i="+i+" ["+errorCodes[i]+"
"+rep.GetErrorMessageByCode(errorCodes[i]));
            }
        }
        rep.Logout();
    }
}
```

Example 3

```
function checkIn(){
    var rep = OZCreator.GetBuiltInObject(OZBuiltInObject.REPOSITORY);
    var login = rep.Login("admin","admin");
    if (login) {
        var item = OZCreator.GetBuiltInObject(OZBuiltInObject.BINARY);
        item.ReadFromFile("c:/parameter_test.odi");
        var itemServerFullPath = "/abc/parameter_test.odi";
        var keepCheckout = true;
        var result = rep.CheckIn(item, itemServerFullPath, keepCheckout, "comment");
    }
```

```
        if (result == true) {
            _DEBUG("result="+result);
        } else {
            _DEBUG("result="+result+"["+rep.GetLastErrorCode()+"]
"+rep.GetErrorMessageByCode(rep.GetLastErrorCode()));
        }
        rep.Logout();
    }
}
```

Example 4

```
function checkIn(){
    var rep = OZCreator.GetBuiltInObject(OZBuiltInObject.REPOSITORY);
    var login = rep.Login("admin","admin");
    if (login) {
        var items = new Array();
        items[0] = OZCreator.GetBuiltInObject(OZBuiltInObject.BINARY);
        items[0].ReadFromFile("c:/parameter_test.odi");
        items[1] = OZCreator.GetBuiltInObject(OZBuiltInObject.BINARY);
        items[1].ReadFromFile("c:/parameter_test.ozr");
        var itemServerFullPaths = new Array();
        itemServerFullPaths[0] = "/data/parameter_test.odi";
        itemServerFullPaths[1] = "/report/parameter_test.ozr";
        var errorCodes = new Array();
        var keepCheckout = new Array();
        keepCheckout[0] = true;
        keepCheckout[1] = true;
        var result = rep.CheckIn(items, itemServerFullPaths, errorCodes, keepCheckout,
"comment");
        if (result == true) {
            _DEBUG("result="+result);
        } else {
            for (i=0; i < errorCodes.length; i++) {
                _DEBUG("i="+i+" ["+errorCodes[i]+"]
"+rep.GetErrorMessageByCode(errorCodes[i]));
            }
        }
        rep.Logout();
    }
}
```

See Also

[OZBinary](#)

CheckOut

아이템을 체크아웃합니다.

Prototype

```
bool CheckOut(string ItemServerFullPath, string ItemLocalFullPath, [bool Download],  
[string Comment])
```

```
bool CheckOut(array ItemServerFullPaths, array ItemLocalFullPaths, array ErrorCodes,  
[array Downloads], [array Comment])
```

```
bool CheckOut(string ItemServerFullPath, object Item, [string Comment])
```

```
bool CheckOut(array ItemServerFullPaths, array Items, array ErrorCodes, [array  
Comment])
```

Parameters

ItemServerFullPath	서버 전체 경로
ItemLocalFullPath	로컬 파일 전체 경로
Download	로컬에 다운로드 여부
Comment	주석 값을 설정하지 않으면 공백으로 동작됨
ItemServerFullPaths	서버 전체 경로를 가진 배열
ItemLocalFullPaths	로컬 파일 전체 경로를 가진 배열
ErrorCodes	에러 코드를 가져올 배열
Downloads	로컬에 다운로드 여부를 가진 배열
Item	아이템 오즈 바이너리 객체로 설정하여야 함
Items	아이템을 가진 배열 오즈 바이너리 객체를 가진 배열로 설정하여야 함

Reference

체크아웃이 성공한 경우 true가 리턴되고, 실패한 경우 false가 리턴됩니다.

에러가 발생한 아이템의 인덱스에 에러 코드 값을 리턴하며, 성공한 경우에는 에러 코드 값에 0을 리턴합니다.

Example 1

```
function checkOut(){
    try {
        var rep = OZCreator.GetBuiltInObject(OZBuiltInObject.REPOSITORY);
        var login = rep.Login("admin","admin");
        if (login) {
            var itemServerFullPath = "/parameter_test.odi";
            var itemLocalFullPath = "c:/parameter_test.odi";
            var download = true;
            var result = rep.CheckOut(itemServerFullPath, itemLocalFullPath, download);
            if (result == true) {
                _DEBUG("result="+result);
            } else {
                _DEBUG("result="+result+"["+rep.GetLastErrorCode()+"]
"+rep.GetErrorMessageByCode(rep.GetLastErrorCode()));
            }
            rep.Logout();
        }
    }catch(e){
        _ERROR(e);
    }
}
```

Example 2

```
function checkOut(){
    try {
        var rep = OZCreator.GetBuiltInObject(OZBuiltInObject.REPOSITORY);
        var login = rep.Login("admin","admin");
        if (login) {
            var itemServerFullPaths = new Array();
            itemServerFullPaths[0] = "/parameter_test.odi";
            itemServerFullPaths[1] = "/parameter_test.ozr";
            var itemLocalFullPaths = new Array();
```

```
        itemLocalFullPaths[0] = OZCreator.GetBuiltInObject(OZBuiltInObject.BINARY);
        itemLocalFullPaths[1] = OZCreator.GetBuiltInObject(OZBuiltInObject.BINARY);
        var errorCodes = new Array();
        var downloads = new Array();
        downloads[0] = true;
        downloads[1] = true;
        var result = rep.CheckOut(itemServerFullPaths, itemLocalFullPaths, errorCodes,
downloads);
        itemLocalFullPaths[0].SaveToFile("c:/parameter_test_checkout.odi");
        itemLocalFullPaths[1].SaveToFile("c:/parameter_test_checkout.ozr");
        if (result == true) {
            _DEBUG("result="+result);
        } else {
            for (i=0; i < errorCodes.length; i++) {
                _DEBUG("checkOut1 i="+i+" ["+errorCodes[i]+""]
"+rep.GetErrorMessageByCode(errorCodes[i]));
            }
        }
        rep.Logout();
    }
} catch(e){
    _ERROR(e);
}
}
```

Example 3

```
function checkOut(){
    var rep = OZCreator.GetBuiltInObject(OZBuiltInObject.REPOSITORY);
    var login = rep.Login("admin","admin");
    if (login) {
        var itemServerFullPath = "/parameter_test.odi";
        var Item = OZCreator.GetBuiltInObject(OZBuiltInObject.BINARY);
        var download = true;
        var result = rep.CheckOut(itemServerFullPath, Item, download);
        Item.SaveToFile("c:/parameter_test_checkout5.odi");
        if (result == true) {
            _DEBUG("result="+result);
        } else {
            _DEBUG("result="+result+"["+rep.GetLastErrorCode()+"]
"+rep.GetErrorMessageByCode(rep.GetLastErrorCode()));
        }
    }
}
```

```
    }  
    rep.Logout();  
  }  
}
```

Example 4

```
function checkOut(){  
  try {  
    var rep = OZCreator.GetBuiltInObject(OZBuiltInObject.REPOSITORY);  
    var login = rep.Login("admin","admin");  
    if (login) {  
      var itemServerFullPaths = new Array();  
      itemServerFullPaths[0] = "/parameter_test.odi";  
      itemServerFullPaths[1] = "/parameter_test.ozr";  
      var items = new Array();  
      items[0] = OZCreator.GetBuiltInObject(OZBuiltInObject.BINARY);  
      items[1] = OZCreator.GetBuiltInObject(OZBuiltInObject.BINARY);  
      var errorCodes = new Array();  
      var download = new Array();  
      download[0] = true;  
      download[1] = true;  
      var result = rep.CheckOut(itemServerFullPaths, items, errorCodes, download);  
      items[0].SaveToFile("c:/parameter_test_checkout.odi");  
      items[1].SaveToFile("c:/parameter_test_checkout.ozr");  
      if (result == true) {  
        _DEBUG("result="+result);  
      } else {  
        for (i=0; i < errorCodes.length; i++) {  
          _DEBUG("checkOut1 i="+i+" ["+errorCodes[i]+""]  
"+rep.GetErrorMessageByCode(errorCodes[i]));  
        }  
      }  
      rep.Logout();  
    }  
  }catch(e){  
    _ERROR(e);  
  }  
}
```

See Also

CheckUserPassword

사용자의 패스워드가 맞는지 확인합니다.

Prototype

bool CheckUserPassword(string UserName, string Password)

Parameters

UserName	사용자 이름
Password	패스워드

Reference

패스워드가 맞는 경우 true가 리턴되고, 맞지 않는 경우 false가 리턴됩니다.

Example

```
unction CheckUserPassword() {  
    var rep = OZCreator.GetBuiltInObject(OZBuiltInObject.REPOSITORY);  
    var login = rep.Login("admin","admin");  
    if (login) {  
        var username = "admin";  
        var password = "admin";  
        var result = rep.CheckUserPassword(username, password);  
        if (result == true) {  
            _DEBUG("check user password result="+result);  
        } else {  
            _DEBUG("check user password  result="+result+"["+rep.GetLastErrorCode()+"]  
"+rep.GetErrorMessageByCode(rep.GetLastErrorCode()));  
        }  
        rep.Logout();  
    }  
}
```

ClearHistoryItem

아이템의 히스토리 중 특정 버전의 히스토리를 삭제합니다.

Prototype

bool ClearHistoryItem(string ItemFullPath, integer ItemVersion)

Parameters

ItemFullPath	아이템 전체 경로
ItemVersion	아이템 버전

Reference

히스토리가 삭제된 경우 true가 리턴되고, 실패한 경우 false가 리턴됩니다.

Example

```
function ClearHistoryItem(){
    var rep = OZCreator.GetBuiltInObject(OZBuiltInObject.REPOSITORY);
    var login = rep.Login("admin","admin");
    if (login) {
        var itemFullPath = "/parameter_test.odi";
        var itemVersion = 1;
        var result = rep.ClearHistoryItem(itemFullPath, itemVersion);
        if (result == true) {
            _DEBUG("ClearHistoryItem result="+result);
        } else {
            _DEBUG("ClearHistoryItem result="+result+" ["+rep.GetLastErrorCode()+"]
"+rep.GetErrorMessageByCode(rep.GetLastErrorCode()));
        }
        rep.Logout();
    }
}
```

CreateGroup

새로운 그룹을 생성합니다.

Prototype

bool CreateGroup(string ParentGroupPath, string NewGroupName, [string Description])

Parameters

ParentGroupPath	상위 그룹 전체 경로
NewGroupName	그룹 이름
Description	그룹 설명

Reference

그룹이 생성된 경우 true가 리턴되고, 실패한 경우 false가 리턴됩니다.

Example 1

```
function CreateGroup(){
    var rep = OZCreator.GetBuiltInObject(OZBuiltInObject.REPOSITORY);
    var login = rep.Login("admin","admin");
    if (login) {
        var parentGroupPath = "/";
        var newGroupName = "newgroup";
        var descriptions = "newgroup desc";
        var result = rep.CreateGroup(parentGroupPath, newGroupName, descriptions);
        if (result == true) {
            _DEBUG("result="+result);
        } else {
            _DEBUG("result="+result+"["+rep.GetLastErrorCode()+"]
"+rep.GetErrorMessageByCode(rep.GetLastErrorCode()));
        }
        rep.Logout();
    }
}
```

Example 2

```
function CreateGroup(){
    var rep = OZCreator.GetBuiltInObject(OZBuiltInObject.REPOSITORY);
    var login = rep.Login("admin","admin");
    if (login) {
        var parentGroupPath = "/";
        var newGroupName = "newgroup";
        var result = rep.CreateGroup(parentGroupPath, newGroupName);
        if (result == true) {
```

```
        _DEBUG("result="+result);
    } else {
        _DEBUG("result="+result+"["+rep.GetLastErrorCode()+"]
"+rep.GetErrorMessageByCode(rep.GetLastErrorCode()));
    }
    rep.Logout();
}
}
```

CreateUser

새로운 사용자를 추가합니다.

Prototype

bool CreateUser(string GroupPath, string UserName, string Password, [string Description])

Parameters

GroupPath	그룹 경로
UserName	사용자 이름
Password	사용자 패스워드
Description	사용자 설명

Reference

사용자가 추가된 경우 true가 리턴되고, 실패한 경우 false가 리턴됩니다.

Example 1

```
function CreateUser() {
    var rep = OZCreator.GetBuiltInObject(OZBuiltInObject.REPOSITORY);
    var login = rep.Login("admin","admin");
    if (login) {
        var groupPath = "/";
        var username = "user1";
        var password = "1234";
        var description = "user1 desc";
        var result = rep.CreateUser(groupPath, username, password, description);
    }
}
```

```

        if (result == true) {
            _DEBUG("result="+result);
        } else {
            _DEBUG("result="+result+" ["+rep.GetLastErrorCode()+"]
"+rep.GetErrorMessageByCode(rep.GetLastErrorCode()));
        }
        rep.Logout();
    }
}

```

Example 2

```

function CreateUser(){
    var rep = OZCreator.GetBuiltInObject(OZBuiltInObject.REPOSITORY);
    var login = rep.Login("admin","admin");
    if (login) {
        var groupPath = "/";
        var username = "user1";
        var password = "1234";
        var result = rep.CreateUser(groupPath, username, password);
        if (result == true) {
            _DEBUG("result="+result);
        } else {
            _DEBUG("result="+result+" ["+rep.GetLastErrorCode()+"]
"+rep.GetErrorMessageByCode(rep.GetLastErrorCode()));
        }
        rep.Logout();
    }
}

```

DeleteCategory

카테고리를 삭제합니다.

Prototype

bool DeleteCategory(string CategoryFullPath, string Comment)

Parameters

CategoryFullPath	카테고리 전체 경로
------------------	------------

Comment	주석
---------	----

Reference

카테고리가 삭제된 경우 true가 리턴되고, 실패한 경우 false가 리턴됩니다.

Example

```
function DeleteCategory(){
    var rep = OZCreator.GetBuiltInObject(OZBuiltInObject.REPOSITORY);
    var login = rep.Login("admin","admin");
    if (login) {
        var categoryFullPath = "/abc";
        var result = rep.DeleteCategory(categoryFullPath,"comment");
        if (result == true) {
            _DEBUG("result="+result);
        } else {
            _DEBUG("result="+result+" ["+rep.GetLastErrorCode()+"]
"+rep.GetErrorMessageByCode(rep.GetLastErrorCode()));
        }
        rep.Logout();
    }
}
```

DeleteGroup

그룹을 삭제합니다.

Prototype

bool DeleteGroup(string GroupPath)

Parameters

GroupPath	그룹 전체 경로
-----------	----------

Reference

그룹이 삭제된 경우 true가 리턴되고, 실패한 경우 false가 리턴됩니다.

Example

```
function DeleteGroup(){
```

```

var rep = OZCreator.GetBuiltInObject(OZBuiltInObject.REPOSITORY);
var login = rep.Login("admin","admin");
if (login) {
    var groupPath = "/newgroup";
    var result = rep.DeleteGroup(groupPath);
    if (result == true) {
        _DEBUG("result="+result);
    } else {
        _DEBUG("result="+result+" ["+rep.GetLastErrorCode()+"]
"+rep.GetErrorMessageByCode(rep.GetLastErrorCode()));
    }
    rep.Logout();
}
}

```

DeleteItem

지정한 아이템을 삭제합니다.

Prototype

bool DeleteItem(string ItemFullPath, string Comment)

Parameters

ItemFullPath	아이템 전체 경로
Comment	주석

Reference

아이템이 삭제된 경우 true가 리턴되고, 실패한 경우 false가 리턴됩니다.

Example

```

function DeleteItem(){
    var rep = OZCreator.GetBuiltInObject(OZBuiltInObject.REPOSITORY);
    var login = rep.Login("admin","admin");
    if (login) {
        var itemFullPath = "/data/parameter_test.odi";
        var result = rep.DeleteItem(itemFullPath,"comment");
        if (result == true) {

```

```

        _DEBUG("result="+result);
    } else {
        _DEBUG("result="+result+" ["+rep.GetLastErrorCode()+"]
"+rep.GetErrorMessageByCode(rep.GetLastErrorCode()));
    }
    rep.Logout();
}
}

```

DeleteItems

지정한 아이템을 삭제합니다.

Prototype

bool DeleteItems(array ItemFullPaths, array ErrorCodes, string Comment)

Parameters

ItemFullPaths	아이템 전체 경로를 가진 배열
ErrorCodes	에러 코드를 가져올 배열
Comment	주석

Reference

아이템이 삭제된 경우 true가 리턴되고, 실패한 경우 false가 리턴됩니다.

에러가 발생한 아이템의 인덱스에 에러 코드 값을 리턴하며, 성공한 경우에는 에러 코드 값에 0을 리턴합니다.

Example

```

function DeleteItems(){
    var rep = OZCreator.GetBuiltInObject(OZBuiltInObject.REPOSITORY);
    var login = rep.Login("admin","admin");
    if (login) {
        var itemFullPaths = new Array();
        itemFullPaths[0] = "/data/parameter_test.odi";
        itemFullPaths[1] = "/report/parameter_test.ozr";
        var errorCodes = new Array();
        var result = rep.DeleteItems(itemFullPaths, errorCodes,"comment");
        if (result == true) {

```

```
        _DEBUG("result="+result);
    } else {
        for (i=0; i < errorCodes.length; i++) {
            _DEBUG("DeleteItems i="+i+" ["+errorCodes[i]+""]
"+rep.GetErrorMessageByCode(errorCodes[i]));
        }
    }
    rep.Logout();
}
}
```

DeleteUser

사용자를 삭제합니다.

Prototype

bool DeleteUser(string UserName)

Parameters

UserName	사용자 이름
----------	--------

Reference

사용자가 삭제된 경우 true가 리턴되고, 실패한 경우 false가 리턴됩니다.

Example

```
function DeleteUser(){
    var rep = OZCreator.GetBuiltInObject(OZBuiltInObject.REPOSITORY);
    var login = rep.Login("admin","admin");
    if (login) {
        var username = "user1";
        var result = rep.DeleteUser(username);
        if (result == true) {
            _DEBUG("result="+result);
        } else {
            _DEBUG("result="+result+"["+rep.GetLastErrorCode()+"]
"+rep.GetErrorMessageByCode(rep.GetLastErrorCode()));
        }
        rep.Logout();
    }
}
```

```
}
}
```

DownloadFile

아이템을 파일 또는 메모리로 다운로드합니다.

Prototype

bool DownloadFile(string ItemFullPath, string CreateItemFullName, [bool ReadOnly])

bool DownloadFile(array ItemFullPaths, array CreateItemFullNames, array ErrorCodes, [array ReadOnlys])

bool DownloadFile(string ItemFullPath, object CreateItem)

bool DownloadFile(array ItemFullPaths, array CreateItems, array ErrorCodes)

Parameters

ItemFullPath	아이템 전체 경로
CreateItemFullName	아이템 이름
ReadOnly	읽기 전용 여부 값을 설정하지 않을 경우 false로 적용됨
ItemFullPaths	아이템 전체 경로를 가진 배열
CreateItemFullNames	아이템 이름을 가진 배열
ErrorCodes	에러 코드를 가져올 배열
ReadOnlys	읽기 전용 여부를 가진 배열 값을 설정하지 않을 경우 false로 적용됨
CreateItem	메모리에 다운로드된 아이템 오즈 바이너리 객체로 설정하여야 함
CreateItems	메모리에 다운로드된 아이템을 가져올 배열 오즈 바이너리 객체를 가진 배열로 설정하여야 함

Reference

아이템이 다운로드된 경우 true가 리턴되고, 실패한 경우 false가 리턴됩니다.

로컬에 해당 파일이 이미 존재한 경우 아이템 다운로드 시 덮어쓰기합니다.

에러가 발생한 아이템의 인덱스에 에러 코드 값을 리턴하며, 성공한 경우에는 에러 코드 값에 0을 리턴합니다.

Example 1

```
function downloadFile(){
    var rep = OZCreator.GetBuiltInObject(OZBuiltInObject.REPOSITORY);
    var login = rep.Login("admin","admin");
    if (login) {
        var itemFullPath = "/parameter_test.odi";
        var createItemFullName = "c:/parameter_test.odi";
        var bReadOnly = true;
        var result = rep.DownloadFile(itemFullPath, createItemFullName, bReadOnly);
        if (result == true) {
            _DEBUG("result="+result);
        } else {
            _DEBUG("result="+result+"["+rep.GetLastErrorCode()+"]
"+rep.GetErrorMessageByCode(rep.GetLastErrorCode()));
        }
        rep.Logout();
    }
}
```

Example 2

```
function downloadFile(){
    var rep = OZCreator.GetBuiltInObject(OZBuiltInObject.REPOSITORY);
    var login = rep.Login("admin","admin");
    if (login) {
        var itemFullPaths = new Array();
        itemFullPaths[0] = "/parameter_test.odi";
        itemFullPaths[1] = "/parameter_test.ozr";
        var createItemFullNames = new Array();
        createItemFullNames[0] = "c:/parameter_test.odi";
        createItemFullNames[1] = "c:/parameter_test.ozr";
        var errorCodes = new Array();
        var bReadOnlys = new Array();
        bReadOnlys[0] = false;
        bReadOnlys[1] = false;
        var result = rep.DownloadFile(itemFullPaths, createItemFullNames, errorCodes,
bReadOnlys);
    }
```

```
        if (result == true) {
            _DEBUG("result="+result);
        } else {
            for (i=0; i < errorCodes.length; i++) {
                _DEBUG("DeleteItems i="+i+"["+errorCodes[i]+""]
"+rep.GetErrorMessageByCode(errorCodes[i]));
            }
        }
        rep.Logout();
    }
}
```

Example 3

```
function downloadFile(){
    var rep = OZCreator.GetBuiltInObject(OZBuiltInObject.REPOSITORY);
    var login = rep.Login("admin","admin");
    if (login) {
        var itemFullPath = "/parameter_test.odi";
        var createItem = OZCreator.GetBuiltInObject(OZBuiltInObject.BINARY);
        var result = rep.DownloadFile(itemFullPath, createItem);
        createItem.SaveToFile("c:/parameter_test_ozbinary.odi");
        if (result == true) {
            _DEBUG("result="+result);
        } else {
            _DEBUG("result="+result+"["+rep.GetLastErrorCode()+"]
"+rep.GetErrorMessageByCode(rep.GetLastErrorCode()));
        }
        rep.Logout();
    }
}
```

Example 4

```
function downloadFile(){
    var rep = OZCreator.GetBuiltInObject(OZBuiltInObject.REPOSITORY);
    var login = rep.Login("admin","admin");
    if (login) {
        var itemFullPaths = new Array();
        itemFullPaths[0] = "/parameter_test.odi";
        itemFullPaths[1] = "/parameter_test.ozr";
        var createItems = new Array();
```

```

        createItems[0] = OZCreator.GetBuiltInObject(OZBuiltInObject.BINARY);
        createItems[1] = OZCreator.GetBuiltInObject(OZBuiltInObject.BINARY);
        var errorCodes = new Array();
        var result = rep.DownloadFile(itemFullPaths, createItems, errorCodes);
        if (result == true) {
            _DEBUG("result="+result);
        } else {
            for (i=0; i < errorCodes.length; i++) {
                _DEBUG("DeleteItems i="+i+"["+errorCodes[i]+""]
"+rep.GetErrorMessageByCode(errorCodes[i]));
            }
        }
        rep.Logout();
    }
}

```

See Also

[OZBinary](#)

DownloadFileByVersion

특정 버전의 아이템을 다운로드합니다.

Prototype

bool DownloadFileByVersion(string ItemServerFullPath, integer HistoryItemVersion, string ItemLocalFull)

bool DownloadFileByVersion(string ItemServerFullPath, integer HistoryItemVersion, object CreateItem)

Parameters

ItemServerFullPath	서버 전체 경로
HistoryItemVersion	히스토리 아이템 버전
ItemLocalFullPath	아이템을 다운로드 받을 로컬 경로
CreateItem	메모리에 다운로드된 아이템 오즈 바이너리 객체로 설정하여야 함

Reference

아이템을 다운로드 받은 경우 true가 리턴되고, 실패한 경우 false가 리턴됩니다.

Example

```
var repository = OZCreator.GetBuiltInObject(OZBuiltInObject.REPOSITORY);
var Binary = OZCreator.GetBuiltInObject(OZBuiltInObject.BINARY);
var successToLogin = repository.Login('admin', 'admin');
if(successToLogin)
    _DEBUG("Success to log in to the repository. ");
var ItemID = "/Car.odi";
var paths = new Array();
var versions = new Array();
var dates = new Array();
var checkinUsers = new Array();
var comments = new Array();
var getHistoryEntries = repository.GetHistoryItemList(ItemID, Paths, Versions, Dates,
CheckinUsers, Comments);
if(getHistoryEntries)
    _DEBUG("Success to get history entries of " + ItemID);
var size = Paths.length;
for(var i = 0; i < size - 1; i++) {
    _DEBUG("Path " + Paths[i] + ", Version " + Versions[i] + ", Date " + Dates[i] + ", Checkin
user " + CheckinUsers[i] + ", Comment " + Comments[i]);
    repository.DownloadFileByVersion(ItemID, i, "d:/temp/History/" + ItemID + "_" + i);
    repository.DownloadFileByVersion(ItemID, i, Binary);
    Binary.SaveToFile("d:/temp/History/" + ItemID + "_binary_" + i);
}
var successToLogout = repository.Logout();
if(successToLogout)
    _DEBUG("Success to log out from the repository. ");
```

See Also

[OZBinary](#)

GetAccessibleTopNodePath

리파지토리 최상위 노드의 전체 경로를 가져옵니다.

Prototype

string GetAccessibleTopNodePath()

Example

```
function getCategoryListInCategory(){
    var rep = OZCreator.GetBuiltInObject(OZBuiltInObject.REPOSITORY);
    var login = rep.Login("admin","admin");
    if (login) {
        var cfullpath = "/";
        var childCategoryNames = new Array();
        var childCategoryFullPaths = new Array();
        _DEBUG("Start to check category list");
        var result = rep.GetCategoryListInCategory(cfullpath, childCategoryNames,
childCategoryFullPaths);
        if (result == true) {
            for (i=0; i <childCategoryNames.length; i++) {
                _DEBUG("result="+result+" childCategoryNames["+i+"]="+childCategoryNames[i]+"
childCategoryFullPaths ["+i+"]="+childCategoryFullPaths[i] );
            }
        } else {
            _DEBUG("result="+result+"["+rep.GetLastErrorCode()+"]
"+rep.GetErrorMessageByCode(rep.GetLastErrorCode()));
        }
        rep.Logout();
    }
}
```

GetCategoryListInCategory

카테고리에 포함된 모든 하위 카테고리의 이름과 전체 경로를 가져옵니다.

Prototype

bool GetCategoryListInCategory(string CategoryFullPath, array ChildCategoryNames, array ChildCategoryFullPaths)

Parameters

CategoryFullPath	카테고리 전체 경로
------------------	------------

ChildCategoryNames	하위 카테고리 이름을 가져올 배열
ChildCategoryFullPaths	하위 카테고리 전체 경로를 가져올 배열

Reference

하위 카테고리 정보를 가져온 경우 true가 리턴되고, 실패한 경우 false가 리턴됩니다.

Example

```
function getCategoryListInCategory(){
    var rep = OZCreator.GetBuiltInObject(OZBuiltInObject.REPOSITORY);
    var login = rep.Login("admin","admin");
    if (login) {
        var cfullpath = "/";
        var childCategoryNames = new Array();
        var childCategoryFullPaths = new Array();
        _DEBUG("Start to check category list");
        var result = rep.GetCategoryListInCategory(cfullpath, childCategoryNames,
childCategoryFullPaths);
        if (result == true) {
            for (i=0; i <childCategoryNames.length; i++) {
                _DEBUG("result="+result+" childCategoryNames["+i+"]="+childCategoryNames[i]+"
childCategoryFullPaths ["+i+"]="+childCategoryFullPaths[i] );
            }
        } else {
            _DEBUG("result="+result+"["+rep.GetLastErrorCode()+"]
"+rep.GetErrorMessageByCode(rep.GetLastErrorCode()));
        }
        rep.Logout();
    }
}
```

GetCheckOutFolder

아이템의 체크아웃 폴더를 가져옵니다.

Prototype

string GetCheckOutFolder(string ItemFullPath)

Parameters

ItemFullPath	아이템 전체 경로
--------------	-----------

Example

```
function getCheckOutFolder(){
    var rep = OZCreator.GetBuiltInObject(OZBuiltInObject.REPOSITORY);
    var login = rep.Login("admin","admin");
    if (login) {
        var itemFullPath = "/parameter_test.odi";
        var result = rep.GetCheckOutFolder(itemFullPath);
        _DEBUG("result="+result);
        rep.Logout();
    }
}
```

GetCheckOutUserName

아이템의 체크아웃 사용자 이름을 가져옵니다.

Prototype

string GetCheckOutUserName(string ItemFullPath)

Parameters

ItemFullPath	아이템 전체 경로
--------------	-----------

Example

```
function getCheckOutUserName(){
    var rep = OZCreator.GetBuiltInObject(OZBuiltInObject.REPOSITORY);
    var login = rep.Login("admin","admin");
    if (login) {
        var itemFullPath = "/parameter_test.odi";
        var result = rep.GetCheckOutUserName(itemFullPath);
        _DEBUG("result="+result);
        rep.Logout();
    }
}
```

GetErrorMessageByCode

에러 코드에 해당하는 에러 메시지를 가져옵니다.

Prototype

string GetErrorMessageByCode(long ErrorCode)

Parameters

ErrorCode	에러 코드
-----------	-------

Example

```
function getErrorMessageByCode(){
    var rep = OZCreator.GetBuiltInObject(OZBuiltInObject.REPOSITORY);
    var login = rep.Login("admin","admin");
    if (login) {
        var groupPath = "/";
        var groupAdminNames = new Array();
        var result = rep.GetGroupAdminListInGroup(groupPath, groupAdminNames);
        if (result == true) {
            for (i=0; i < groupAdminNames.length; i++) {
                _DEBUG("result="+result+"    groupAdminNames["+i+"]="+groupAdminNames[i]);
            }
        } else {
            _DEBUG("result="+result+"["+rep.GetLastErrorCode()+"]
"+rep.GetErrorMessageByCode(rep.GetLastErrorCode()));
        }
        rep.Logout();
    }
}
```

GetGroupAdminListInGroup

그룹 관리자 정보를 가져옵니다.

Prototype

bool GetGroupAdminListInGroup(string GroupPath, array GroupAdminNames)

Parameters

GroupPath	그룹 경로
GroupAdminNames	그룹 관리자 이름을 가져올 배열

Reference

그룹 관리자 정보를 가져온 경우 true가 리턴되고, 실패한 경우 false가 리턴됩니다.

Example

```
function getGroupAdminListInGroup(){
    var rep = OZCreator.GetBuiltInObject(OZBuiltInObject.REPOSITORY);
    var login = rep.Login("admin","admin");
    if (login) {
        var groupPath = "/";
        var groupAdminNames = new Array();
        var result = rep.GetGroupAdminListInGroup(groupPath, groupAdminNames);
        if (result == true) {
            for (i=0; i < groupAdminNames.length; i++) {
                _DEBUG("result="+result+"    groupAdminNames["+i+"]="+groupAdminNames[i]);
            }
        } else {
            _DEBUG("result="+result+"["+rep.GetLastErrorCode()+"]
"+rep.GetErrorMessageByCode(rep.GetLastErrorCode()));
        }
        rep.Logout();
    }
}
```

GetGroupAuthToCategory

카테고리에 대한 그룹의 권한을 가져옵니다.

Prototype

string GetGroupAuthToCategory(string CategoryFullPath, string GroupPath)

Parameters

CategoryFullPath	카테고리 전체 경로
------------------	------------

GroupPath	그룹 경로
-----------	-------

Example

```
function getGroupAuthToCategory(){
    var rep = OZCreator.GetBuiltInObject(OZBuiltInObject.REPOSITORY);
    var login = rep.Login("admin","admin");
    if (login) {
        var categoryFullPath = "/";
        var groupPath = "/";
        var result = rep.GetGroupAuthToCategory(categoryFullPath, groupPath);
        _DEBUG("result="+result);
        rep.Logout();
    }
}
```

GetGroupAuthToItem

아이템에 대한 그룹의 권한을 가져옵니다.

Prototype

string GetGroupAuthToItem(string ItemFullPath, string GroupPath)

Parameters

ItemFullPath	아이템 전체 경로
GroupPath	그룹 경로

Example

```
function getGroupAuthToItem(){
    var rep = OZCreator.GetBuiltInObject(OZBuiltInObject.REPOSITORY);
    var login = rep.Login("admin","admin");
    if (login) {
        var itemFullPath = "/parameter_test.odi";
        var groupPath = "/";
        var result = rep.GetGroupAuthToItem(itemFullPath, groupPath);
        _DEBUG("result="+result);
        rep.Logout();
    }
}
```

}

GetGroupListAuthToCategory

카테고리에 대해 특정 권한 이상의 권한을 가진 모든 그룹 정보를 가져옵니다.

Prototype

bool GetGroupListAuthToCategory(string CategoryFullPath, string Permission, array GroupNames, array GroupFullPaths)

Parameters

CategoryFullPath	카테고리 전체 경로
Permission	권한 정보 None, View, Read, ReadWrite 중 하나의 값으로 설정
GroupNames	그룹 이름을 가져올 배열
GroupFullPaths	그룹 전체 경로를 가져올 배열

Reference

그룹 정보를 가져온 경우 true가 리턴되고, 실패한 경우 false가 리턴됩니다.

Example

```
function getGroupListAuthToCategory(){
    var rep = OZCreator.GetBuiltInObject(OZBuiltInObject.REPOSITORY);
    var login = rep.Login("admin","admin");
    if (login) {
        var categoryFullPath = "/";
        var permission = "ReadWrite";
        //None, View, Read, ReadWrite
        var groupNames = new Array();
        var groupFullPaths = new Array();
        var result = rep.GetGroupListAuthToCategory(categoryFullPath, permission, groupNames,
groupFullPaths);
        if (result == true) {
            for (i=0; i < groupNames.length; i++) {
                _DEBUG("result="+result+" groupNames["+i+"]="+groupNames[i]+"
groupFullPaths["+i+"]="+groupFullPaths[i]);
            }
        }
    }
}
```

```

    }
  } else {
    _DEBUG("result="+result+" ["+rep.GetLastErrorCode()+"]
"+rep.GetErrorMessageByCode(rep.GetLastErrorCode()));
  }
  rep.Logout();
}
}
}

```

GetGroupListAuthToItem

아이템에 대해 특정 권한 이상의 권한을 가진 모든 그룹 정보를 가져옵니다.

Prototype

bool GetGroupListAuthToItem(string ItemFullPath, string Permission, array GroupNames, array GroupFullPaths)

Parameters

ItemFullPath	아이템 전체 경로
Permission	권한 정보 None, View, Read, ReadWrite 중 하나의 값으로 설정
GroupNames	그룹 이름을 가져올 배열
GroupFullPaths	그룹 전체 경로를 가져올 배열

Reference

그룹 정보를 가져온 경우 true가 리턴되고, 실패한 경우 false가 리턴됩니다.

Example

```

function getGroupListAuthToItem(){
  var rep = OZCreator.GetBuiltInObject(OZBuiltInObject.REPOSITORY);
  var login = rep.Login("admin","admin");
  if (login) {
    var itemFullPath = "/parameter_test.odi";
    var permission = "ReadWrite";
    //None, View, Read, ReadWrite
    var groupNames = new Array();
  }
}

```

```

        var groupFullPaths = new Array();
        var result = rep.GetGroupListAuthToItem(itemFullPath, permission, groupNames,
groupFullPaths);
        if (result == true) {
            for (i=0; i < groupNames.length; i++) {
                _DEBUG("result="+result+" groupNames["+i+"]="+groupNames[i]+"
groupFullPaths["+i+"]="+groupFullPaths[i]);
            }
        } else {
            _DEBUG("result="+result+" ["+rep.GetLastErrorCode()+"]
"+rep.GetErrorMessageByCode(rep.GetLastErrorCode()));
        }
        rep.Logout();
    }
}

```

GetGroupListInGroup

모든 하위 그룹의 정보를 가져옵니다.

Prototype

bool GetGroupListInGroup(string GroupPath, array GroupNames, array GroupFullPaths)

Parameters

GroupPath	그룹 경로
GroupNames	그룹 이름을 가져올 배열
GroupFullPaths	그룹 전체 경로를 가져올 배열

Reference

하위 그룹 정보를 가져온 경우 true가 리턴되고, 실패한 경우 false가 리턴됩니다.
그룹 경로가 루트인 경우 "/"로 설정합니다.

Example

```

function getGroupListInGroup(){
    var rep = OZCreator.GetBuiltInObject(OZBuiltInObject.REPOSITORY);
    var login = rep.Login("admin","admin");
    if (login) {

```

```

        var groupPath = "/";
        var groupNames = new Array();
        var groupFullPaths = new Array();
        var result = rep.GetGroupListInGroup(groupPath, groupNames, groupFullPaths);
        if (result == true) {
            for (i=0; i < groupNames.length; i++) {
                _DEBUG("result="+result+" groupNames["+i+"]="+groupNames[i]+"
groupFullPaths["+i+"]="+groupFullPaths[i]);
            }
        } else {
            _DEBUG("result="+result+" ["+rep.GetLastErrorCode()+"]
"+rep.GetErrorMessageByCode(rep.GetLastErrorCode()));
        }
        rep.Logout();
    }
}

```

GetGroupPathOfUser

사용자의 그룹 경로를 가져옵니다.

Prototype

string GetGroupPathOfUser(string UserName)

Parameters

UserName	사용자 이름
----------	--------

Example

```

function getGroupPathOfUser(){
    var rep = OZCreator.GetBuiltInObject(OZBuiltInObject.REPOSITORY);
    var login = rep.Login("admin","admin");
    if (login) {
        var username = "admin";
        var result = rep.GetGroupPathOfUser(username);
        _DEBUG("result="+result);
        rep.Logout();
    }
}

```

GetHistoryItemList

아이템의 히스토리 정보를 가져옵니다.

Prototype

bool GetHistoryItemList(string ItemFullPath, array HistoryItemPaths, array HistoryItemVersions, array HistoryItemDates, array HistoryItemCheckInUser, [array HistoryItemComments])

Parameters

ItemFullPath	아이템 전체 경로
HistoryItemPaths	히스토리 아이템의 경로를 가져올 배열
HistoryItemVersions	히스토리 아이템의 버전을 가져올 배열
HistoryItemDates	히스토리 아이템의 날짜를 가져올 배열
HistoryItemCheckInUser	히스토리 아이템의 체크인 사용자를 가져올 배열
HistoryItemComments	히스토리 아이템의 주석을 가져올 배열

Reference

히스토리 정보를 가져온 경우 true가 리턴되고, 실패한 경우 false가 리턴됩니다.

Example

```
function getHistoryItemList(){
    var rep = OZCreator.GetBuiltInObject(OZBuiltInObject.REPOSITORY);
    var login = rep.Login("admin","admin");
    if (login) {
        var itemFullPath = "/parameter_test.odi";
        var historyItemPaths = new Array();
        var historyItemVersions = new Array();
        var historyItemDates = new Array();
        var historyItemCheckInUser = new Array();
        var result = rep.GetHistoryItemList(itemFullPath, historyItemPaths, historyItemVersions,
historyItemDates, historyItemCheckInUser);
        if (result == true) {
            for (i=0; i < historyItemPaths.length; i++) {
```

```

        _DEBUG("result="+result+"historyItemPaths["+i+"]="+historyItemPaths[i]
+"historyItemVersions["+i+"]="+historyItemVersions[i]+"    historyItemDates["+i+"]
="+historyItemDates[i]+"historyItemCheckInUser["+i+"]="+historyItemCheckInUser[i] );
    }
} else {
    _DEBUG("result="+result+"["+rep.GetLastErrorCode()+"]
"+rep.GetErrorMessageByCode(rep.GetLastErrorCode()));
}
rep.Logout();
}
}

```

GetItemListInCategory

카테고리에 포함된 모든 아이템 이름을 가져옵니다.

Prototype

bool GetItemListInCategory(string CategoryFullPath, array ChildItemNames)

Parameters

CategoryFullPath	카테고리 전체 경로
ChildItemNames	아이템 이름을 가져올 배열

Reference

아이템 이름을 가져온 경우 true가 리턴되고, 실패한 경우 false가 리턴됩니다.

Example

```

function getItemListInCategory(){
    var rep = OZCreator.GetBuiltInObject(OZBuiltInObject.REPOSITORY);
    var login = rep.Login("admin","admin");
    if (login) {
        var cfullpath = "/";
        var childItemNames = new Array();
        var result = rep.GetItemListInCategory(cfullpath, childItemNames);
        if (result == true) {
            for (i=0; i < childItemNames.length; i++) {
                _DEBUG("result="+result+"childItemNames["+i+"]="+childItemNames[i]);
            }
        }
    }
}

```

```
    }  
  } else {  
    _DEBUG("result="+result+"["+rep.GetLastErrorCode()+"]  
"+rep.GetErrorMessageByCode(rep.GetLastErrorCode()));  
  }  
  rep.Logout();  
}  
}
```

GetLastDetailErrorMessage

마지막으로 발생한 에러의 에러 메시지를 가져옵니다.

Prototype

string GetLastDetailErrorMessage()

Reference

마지막으로 수행한 동작에 에러가 없을 경우 공백 문자를 리턴합니다.

Example

```
function getLastDetailErrorMessage(){  
  var rep = OZCreator.GetBuiltInObject(OZBuiltInObject.REPOSITORY);  
  var login = rep.Login("admin","admin");  
  if (login) {  
    var cfullpath = "/";  
    var childItemNames = new Array();  
    var result = rep.GetItemListInCategory(cfullpath, childItemNames);  
    if (result == true) {  
      for (i=0; i < childItemNames.length; i++) {  
        _DEBUG("result="+result+"      childItemNames["+i+"]="+childItemNames[i]);  
      }  
    } else {  
      _DEBUG("result="+result+", "+rep.GetLastDetailErrorMessage());  
    }  
    rep.Logout();  
  }  
}
```

GetLastErrorCode

마지막으로 발생한 에러의 에러 코드를 가져옵니다.

Prototype

long GetLastErrorCode()

Reference

마지막으로 수행한 동작에 에러가 없을 경우 0을 리턴합니다.

Example

```
function getLastErrorCode(){
    var rep = OZCreator.GetBuiltInObject(OZBuiltInObject.REPOSITORY);
    var login = rep.Login("admin","admin");
    if (login) {
        var cfullpath = "/";
        var childItemNames = new Array();
        var result = rep.GetItemListInCategory(cfullpath, childItemNames);
        if (result == true) {
            for (i=0; i < childItemNames.length; i++) {
                _DEBUG("result="+result+"      childItemNames["+i+"]="+childItemNames[i]);
            }
        } else {
            _DEBUG("result="+result+"["+rep.GetLastErrorCode()+"]
"+rep.GetErrorMessageByCode(rep.GetLastErrorCode()));
        }
        rep.Logout();
    }
}
```

GetUserAuthToCategory

카테고리에 대한 사용자의 권한 정보를 가져옵니다.

Prototype

string GetUserAuthToCategory(string CategoryFullPath, string UserName)

Parameters

CategoryFullPath	카테고리 전체 경로
------------------	------------

UserName	사용자 이름
----------	--------

Example

```
function getUserAuthToCategory(){
    var rep = OZCreator.GetBuiltInObject(OZBuiltInObject.REPOSITORY);
    var login = rep.Login("admin","admin");
    if (login) {
        var categoryFullPath = "/";
        var username = "admin";
        var result = rep.GetUserAuthToCategory(categoryFullPath, username);
        _DEBUG("result="+result);
        rep.Logout();
    }
}
```

GetUserAuthToItem

아이템에 대한 지정한 사용자의 권한 정보를 가져옵니다.

Prototype

string GetUserAuthToItem(string ItemFullPath, string UserName)

Parameters

ItemFullPath	아이템 전체 경로
UserName	사용자 이름

Example

```
function getUserAuthToItem(){
    var rep = OZCreator.GetBuiltInObject(OZBuiltInObject.REPOSITORY);
    var login = rep.Login("admin","admin");
    if (login) {
        var itemFullPath = "/parameter_test.odi";
        var username = "admin";
        var result = rep.GetUserAuthToItem(itemFullPath, username);
        _DEBUG("result="+result);
        rep.Logout();
    }
}
```

}

GetUserListAuthToCategory

카테고리에 대한 사용자의 권한이 특정 권한 이상인 모든 사용자 정보를 가져옵니다.

Prototype

bool GetUserListAuthToCategory(string CategoryFullPath, string Permission, array UserNames, array Passwords, array Descriptions)

Parameters

CategoryFullPath	카테고리 전체 경로
Permission	권한 정보 None, View, Read, ReadWrite 중 하나의 값으로 설정
UserNames	사용자 이름을 가져올 배열
Passwords	사용자 패스워드를 가져올 배열
Descriptions	설명을 가져올 배열

Reference

사용자 정보를 가져온 경우 true가 리턴되고, 실패한 경우 false가 리턴됩니다.

사용자 패스워드는 항상 공백으로 리턴됩니다.

Example

```
function getUserListAuthToCategory(){
    var rep = OZCreator.GetBuiltInObject(OZBuiltInObject.REPOSITORY);
    var login = rep.Login("admin","admin");
    if (login) {
        var categoryFullPath = "/";
        var permission = "ReadWrite";
        //None, View, Read, ReadWrite
        var usernames = new Array();
        var passwords = new Array();
        var descriptions = new Array();
        var result = rep.GetUserListAuthToCategory(categoryFullPath, permission, usernames,
passwords, descriptions);
```

```

        if (result == true) {
            for (i=0; i < usernames.length; i++) {
                _DEBUG("result="+result+" usernames["+i+"]="+usernames[i]+" passwords["+i+"]
="+passwords[i]+" descriptions["+i+"]="+descriptions[i]);
            }
        } else {
            _DEBUG("result="+result+"["+rep.GetLastErrorCode()+"]
"+rep.GetErrorMessageByCode(rep.GetLastErrorCode()));
        }
        rep.Logout();
    }
}

```

GetUserListAuthToItem

아이템에 대한 사용자의 권한이 특정 권한 이상인 모든 사용자 정보를 가져옵니다.

Prototype

bool GetUserListAuthToItem(string ItemFullPath, string Permission, array UserNames, array Passwords, array Descriptions)

Parameters

ItemFullPath	아이템 전체 경로
Permission	권한 정보 None, View, Read, ReadWrite 중 하나의 값으로 설정
UserNames	사용자 이름을 가져올 배열
Passwords	사용자 패스워드를 가져올 배열
Descriptions	설명을 가져올 배열

Reference

사용자 정보를 가져온 경우 true가 리턴되고, 실패한 경우 false가 리턴됩니다.

사용자 패스워드는 항상 공백으로 리턴됩니다.

Example

```
function getUserListAuthToItem(){
```

```

var rep = OZCreator.GetBuiltInObject(OZBuiltInObject.REPOSITORY);
var login = rep.Login("admin","admin");
if (login) {
    var itemFullPath = "/parameter_test.odi";
    var permission = "ReadWrite";
    //None, View, Read, ReadWrite
    var usernames = new Array();
    var passwords = new Array();
    var descriptions = new Array();
    var result = rep.GetUserListAuthToItem(itemFullPath, permission, usernames, passwords,
descriptions);
    if (result == true) {
        for (i=0; i < usernames.length; i++) {
            _DEBUG("result="+result+" usernames["+i+"]="+usernames[i]+ " passwords["+i+"]
="+passwords[i]+ " descriptions["+i+"]="+descriptions[i]);
        }
    } else {
        _DEBUG("result="+result+" ["+rep.GetLastErrorCode()+"]
"+rep.GetErrorMessageByCode(rep.GetLastErrorCode()));
    }
    rep.Logout();
}
}

```

GetUserListInGroup

그룹에 추가된 모든 사용자 정보를 가져옵니다.

Prototype

bool GetUserListInGroup(string GroupPath, array UserNames, array Passwords, array Descriptions)

Parameters

GroupPath	그룹 경로
UserNames	사용자 이름을 가져올 배열
Passwords	사용자 패스워드를 가져올 배열
Descriptions	설명을 가져올 배열

Reference

사용자 정보를 가져온 경우 true가 리턴되고, 실패한 경우 false가 리턴됩니다.

사용자 패스워드는 항상 공백으로 리턴됩니다.

Example

```
function getUserListInGroup(){
    var rep = OZCreator.GetBuiltInObject(OZBuiltInObject.REPOSITORY);
    var login = rep.Login("admin","admin");
    if (login) {
        var groupPath = "/";
        var usernames = new Array();
        var passwords = new Array();
        var descriptions = new Array();
        var result = rep.GetUserListInGroup(groupPath, usernames, passwords, descriptions);
        if (result == true) {
            for (i=0; i < usernames.length; i++) {
                _DEBUG("result="+result+" usernames["+i+"]="+usernames[i]+" passwords["+i+"]"
                    +passwords[i]+" descriptions["+i+"]="+descriptions[i]);
            }
        } else {
            _DEBUG("result="+result+"["+rep.GetLastErrorCode()+"]"
                +rep.GetErrorMessageByCode(rep.GetLastErrorCode()));
        }
        rep.Logout();
    }
}
```

IsFileExist

파일의 존재 여부를 가져옵니다.

Prototype

bool IsFileExist(string FullPath)

Parameters

FullPath	파일 경로
----------	-------

Example

```
function isFileExist(){
    var rep = OZCreator.GetBuiltInObject(OZBuiltInObject.REPOSITORY);
    var login = rep.Login("admin","admin");
    if (login) {
        var fullpath = "/";
        var result = rep.IsFileExist(fullpath);
        _DEBUG("result="+result);
        rep.Logout();
    }
}
```

Login

스크립트가 실행 중인 서버의 리파지토리에 로그인합니다.

Prototype

bool Login(string User, string Password)

Parameters

User	사용자 ID
Password	사용자 패스워드

Reference

로그인이 성공한 경우 true가 리턴되고, 실패한 경우 false가 리턴됩니다.

Example

```
function login(){
    var rep = OZCreator.GetBuiltInObject(OZBuiltInObject.REPOSITORY);
    var login = rep.Login("admin","admin");
    _DEBUG("is login="+login);
    rep.Logout();
}
```

LoginToServer

지정한 서버에 로그인합니다.

Prototype

bool LoginToServer(string URL, string User, string Password)

Parameters

URL	오즈 서버 URL
User	사용자 ID
Password	사용자 패스워드

Reference

로그인이 성공한 경우 true가 리턴되고, 실패한 경우 false가 리턴됩니다.

Example

```
function loginToServerURL(){
    _DEBUG("OZ Repository Test start. ASP.NET/servlet");
    var rep = OZCreator.GetBuiltInObject(OZBuiltInObject.REPOSITORY);
    var login = rep.LoginToServer("http://127.0.0.1/oz/server", "admin","admin");
    _DEBUG("is login="+login);
    rep.Logout();
}
```

Logout

서버에서 로그아웃합니다.

Prototype

bool Logout()

Reference

로그아웃이 성공한 경우 true가 리턴되고, 실패한 경우 false가 리턴됩니다.

Example

```
function logout(){
    var rep = OZCreator.GetBuiltInObject(OZBuiltInObject.REPOSITORY);
```

```

    var login = rep.Login("admin","admin");
    _DEBUG("is login="+login);
    var logout = rep.Logout();
    _DEBUG("is logout="+logout);
}

```

RemoveGroupAdmin

그룹 관리자를 해제합니다.

Prototype

bool RemoveGroupAdmin(string GroupPath, string AdminUserName)

Parameters

GroupPath	그룹 경로
AdminUserName	그룹 관리자 권한을 해제할 사용자 이름

Reference

그룹 관리자가 해제된 경우 true가 리턴되고, 실패한 경우 false가 리턴됩니다.

Example

```

function removeGroupAdmin(){
    var rep = OZCreator.GetBuiltInObject(OZBuiltInObject.REPOSITORY);
    var login = rep.Login("admin","admin");
    if (login) {
        var groupPath = "/";
        var adminUsername = "user1";
        var result = rep.RemoveGroupAdmin(groupPath, adminUsername);
        if (result == true) {
            _DEBUG("result="+result);
        } else {
            _DEBUG("result="+result+" ["+rep.GetLastErrorCode()+"]
"+rep.GetErrorMessageByCode(rep.GetLastErrorCode()));
        }
        rep.Logout();
    }
}

```

RenameCategory

카테고리 이름을 변경합니다.

Prototype

bool RenameCategory(string CategoryFullPath, string NewCategoryName, string Comment)

Parameters

CategoryFullPath	카테고리 전체 경로
NewCategoryName	카테고리 이름
Comment	주석

Reference

카테고리 이름이 변경된 경우 true가 리턴되고, 실패한 경우 false가 리턴됩니다.

Example

```
function renameCategory(){
    var rep = OZCreator.GetBuiltInObject(OZBuiltInObject.REPOSITORY);
    var login = rep.Login("admin","admin");
    if (login) {
        var categoryFullPath = "/data";
        var newCategoryName = "newData";
        var result = rep.RenameCategory(categoryFullPath, newCategoryName,"comment");
        if (result == true) {
            _DEBUG("result="+result);
        } else {
            _DEBUG("result="+result+" ["+rep.GetLastErrorCode()+"]
"+rep.GetErrorMessageByCode(rep.GetLastErrorCode()));
        }
        rep.Logout();
    }
}
```

RenameItem

아이템 이름을 변경합니다.

Prototype

bool RenameItem(string ItemFullPath, string NewItemName, string Comment)

Parameters

ItemFullPath	아이템 전체 경로
NewItemName	아이템 이름
Comment	주석

Reference

아이템 이름이 변경된 경우 true가 리턴되고, 실패한 경우 false가 리턴됩니다.

Example

```
function renameItem(){
    var rep = OZCreator.GetBuiltInObject(OZBuiltInObject.REPOSITORY);
    var login = rep.Login("admin","admin");
    if (login) {
        var itemFullPath = "/parameter_test.odi";
        var newItemName = "parameter_test_new.odi";
        var itemFullPath2 = "/parameter_test_new.odi";
        var newItemName2 = "parameter_test.odi";
        var result = rep.RenameItem(itemFullPath, newItemName,"comment");
        if (result == true) {
            _DEBUG("result="+result);
        } else {
            _DEBUG("result="+result+" ["+rep.GetLastErrorCode()+"]
"+rep.GetErrorMessageByCode(rep.GetLastErrorCode()));
        }
        result = rep.RenameItem(itemFullPath2, newItemName2,"comment");
        if (result == true) {
            _DEBUG("result="+result);
        } else {
            _DEBUG("result="+result+" ["+rep.GetLastErrorCode()+"]
"+rep.GetErrorMessageByCode(rep.GetLastErrorCode()));
        }
    }
    rep.Logout();
}
```

```
}
}
```

RollbackItem

아이템을 롤백합니다.

Prototype

bool RollbackItem(string ItemFullPath, integer ItemVersion, string Comment)

Parameters

ItemFullPath	아이템 전체 경로
ItemVersion	아이템 버전
Comment	주석

Reference

롤백된 경우 true가 리턴되고, 실패한 경우 false가 리턴됩니다.

Example

```
function renameItem(){
    var rep = OZCreator.GetBuiltInObject(OZBuiltInObject.REPOSITORY);
    var login = rep.Login("admin","admin");
    if (login) {
        var itemFullPath = "/parameter_test.odi";
        var newItemName = "parameter_test_new.odi";
        var itemFullPath2 = "/parameter_test_new.odi";
        var newItemName2 = "parameter_test.odi";
        var result = rep.RenameItem(itemFullPath, newItemName,"comment");
        if (result == true) {
            _DEBUG("result="+result);
        } else {
            _DEBUG("result="+result+" ["+rep.GetLastErrorCode()+"]
"+rep.GetErrorMessageByCode(rep.GetLastErrorCode()));
        }
        result = rep.RenameItem(itemFullPath2, newItemName2,"comment");
        if (result == true) {
            _DEBUG("result="+result);
        }
    }
}
```

```

    } else {
        _DEBUG("result="+result+" ["+rep.GetLastErrorCode()+"]
"+rep.GetErrorMessageByCode(rep.GetLastErrorCode()));
    }
    rep.Logout();
}
}

```

SearchItems

아이템을 검색합니다.

Prototype

```
bool SearchItems(string CategoryFullPath, integer SearchOption, string SearchText,
string CheckOutUser, string CheckInStartDate, string CheckInEndDate, string
ItemName, array ItemFullPaths, array HistoryItemVersions)
```

Parameters

CategoryFullPath	아이템을 검색할 카테고리 전체 경로
SearchOption	검색 옵션. SearchOptionConst 참고 " "를 이용하여 여러 개의 값을 조합하여 설정 가능
SearchText	주석 또는 설명
CheckOutUser	체크아웃 사용자
CheckInStartDate	체크인 시작 날짜
CheckInEndDate	체크인 종료 날짜
ItemName	아이템 이름
ItemFullPaths	검색된 아이템의 전체 경로를 가져올 배열
HistoryItemVersions	검색된 아이템의 버전을 가져올 배열

Reference

아이템을 가져온 경우 true가 리턴되고, 실패한 경우 false가 리턴됩니다.

Example

```

var repository = OZCreator.GetBuiltInObject(OZBuiltInObject.REPOSITORY);
var successToLogin = repository.Login('admin', 'admin');
if(successToLogin)
    _DEBUG("Success to log in to the repository. ");
var Options = SearchOptions.IncludeSubCategory | SearchOptions.AllCheckoutItem |
SearchOptions.CheckItemName;
_DEBUG("Search option " + Options);
var ItemIDs = new Array();
var Versions = new Array();
var result = repository.SearchItems("/", Options, "param", "", "", "", "", ItemIDs, Versions);
var length = ItemIDs.length;
for(var i = 0; i < length; i++) {
    _DEBUG("Item ID [" + i + "]" + ItemIDs[i]);
}

```

See Also

[SearchOptionConst](#)

UndoCheckOut

아이템을 체크아웃 취소합니다.

Prototype

bool UndoCheckOut(string ItemServerFullPath, [bool Download], [string DownloadFullPath])

bool UndoCheckOut(string ItemServerFullPath, [bool Download], [object DownloadFullPath])

bool UndoCheckOut(array ItemServerFullPaths, array ErrorCodes, [array Downloads], [array DownloadFullPaths])

Parameters

ItemServerFullPath	로컬 파일 전체 경로
Download	로컬에 다운로드 여부
DownloadFullPath	로컬 파일 전체 경로 문자열 또는 오즈 바이너리 객체로 설정하여야 함

ItemServerFullPaths	로컬 파일 전체 경로를 가진 배열
ErrorCodes	에러 코드를 가져올 배열
Downloads	로컬에 다운로드 여부를 가진 배열
DownloadFullPaths	로컬 파일 전체 경로를 가진 배열 오즈 바이너리 객체를 가진 배열로 설정하여야 함

Reference

체크아웃 취소된 경우 true가 리턴되고, 실패한 경우 false가 리턴됩니다.

Example 1

```
function undoCheckOut(){
    var rep = OZCreator.GetBuiltInObject(OZBuiltInObject.REPOSITORY);
    var login = rep.Login("admin","admin");
    if (login) {
        var itemServerFullPath = "/parameter_test.odi";
        var download = true;
        var downloadFullPath = "c:/parameter_test.odi";
        var result = rep.UndoCheckOut(itemServerFullPath, download, downloadFullPath);
        if (result == true) {
            _DEBUG("result="+result);
        } else {
            _DEBUG("result="+result+" ["+rep.GetLastErrorCode()+"]
"+rep.GetErrorMessageByCode(rep.GetLastErrorCode()));
        }
        rep.Logout();
    }
}
```

Example 2

```
function undoCheckOut(){
    var rep = OZCreator.GetBuiltInObject(OZBuiltInObject.REPOSITORY);
    var login = rep.Login("admin","admin");
    if (login) {
        var itemServerFullPath = "/parameter_test.odi";
        var download = true;
        var downloadFullPath = OZCreator.GetBuiltInObject(OZBuiltInObject.BINARY);
        var result = rep.UndoCheckOut(itemServerFullPath, download, downloadFullPath);
    }
```

```
        if (result == true) {
            _DEBUG("result="+result);
            downloadFullPath.SaveToFile("c:/parameter_test.odi");
        } else {
            _DEBUG("result="+result+" ["+rep.GetLastErrorCode()+"]
"+rep.GetErrorMessageByCode(rep.GetLastErrorCode()));
        }
        rep.Logout();
    }
}
```

Example 3

```
function undoCheckOut(){
    var rep = OZCreator.GetBuiltInObject(OZBuiltInObject.REPOSITORY);
    var login = rep.Login("admin","admin");
    if (login) {
        var itemServerFullPaths = new Array();
        itemServerFullPaths[0] = "/parameter_test.odi";
        itemServerFullPaths[1] = "/parameter_test.ozr";
        var errorCodes = new Array();
        var download = new Array();
        download[0] = true;
        download[1] = true;
        var downloadFullPath = new Array();
        downloadFullPath[0] = "c:/parameter_test.odi";
        downloadFullPath[1] = "c:/parameter_test.ozr";
        var result = rep.UndoCheckOut(itemServerFullPaths, errorCodes, download,
downloadFullPath);
        if (result == true) {
            _DEBUG("result="+result);
        }else {
            for (i=0; i < errorCodes.length; i++) {
                _DEBUG("i="+i+" ["+errorCodes[i]+"]
"+rep.GetErrorMessageByCode(errorCodes[i]));
            }
        }
        rep.Logout();
    }
}
```

UpdateCategoryGroupAuth

카테고리에 대한 그룹 권한을 변경합니다.

Prototype

bool UpdateCategoryGroupAuth(string GroupPath, string CategoryFullPath, string NewPermission)

Parameters

GroupPath	그룹 경로
CategoryFullPath	카테고리 전체 경로
NewPermission	권한 정보 None, View, Read, ReadWrite 중 하나의 값으로 설정

Reference

권한이 변경된 경우 true가 리턴되고, 실패한 경우 false가 리턴됩니다.

Example

```
function updateCategoryGroupAuth(){
    var rep = OZCreator.GetBuiltInObject(OZBuiltInObject.REPOSITORY);
    var login = rep.Login("admin","admin");
    if (login) {
        var groupPath = "/";
        var itemFullPath = "/";
        var newPermission = "ReadWrite";
        var result = rep.UpdateCategoryGroupAuth(groupPath, itemFullPath, newPermission);

        if (result == true) {
            _DEBUG("result="+result);
        } else {
            _DEBUG("result="+result+" ["+rep.GetLastErrorCode()+"]
"+rep.GetErrorMessageByCode(rep.GetLastErrorCode()));
        }
        rep.Logout();
    }
}
```

UpdateCategoryUserAuth

카테고리에 대한 사용자 권한을 변경합니다.

Prototype

bool UpdateCategoryUserAuth(string UserName, string CategoryFullPath, string NewPermission)

Parameters

UserName	사용자 이름
CategoryFullPath	카테고리 전체 경로
NewPermission	권한 정보 None, View, Read, ReadWrite 중 하나의 값으로 설정

Reference

권한이 변경된 경우 true가 리턴되고, 실패한 경우 false가 리턴됩니다.

Example

```
function updateCategoryUserAuth(){
    var rep = OZCreator.GetBuiltInObject(OZBuiltInObject.REPOSITORY);
    var login = rep.Login("admin","admin");
    if (login) {
        var username = "admin";
        var itemFullPath = "/";
        var newPermission = "ReadWrite";
        //None, View, Read, ReadWrite
        var result = rep.UpdateCategoryUserAuth(username, itemFullPath, newPermission);
        if (result == true) {
            _DEBUG("result="+result);
        } else {
            _DEBUG("result="+result+" ["+rep.GetLastErrorCode()+"]
"+rep.GetErrorMessageByCode(rep.GetLastErrorCode()));
        }
        rep.Logout();
    }
}
```

UpdateGroupName

그룹 이름을 변경합니다.

Prototype

bool UpdateGroupName(string GroupPath, string NewGroupName)

Parameters

GroupPath	그룹 경로
NewGroupName	그룹 이름

Reference

그룹 이름이 변경된 경우 true가 리턴되고, 실패한 경우 false가 리턴됩니다.

Example

```
function updateGroupName(){
    var rep = OZCreator.GetBuiltInObject(OZBuiltInObject.REPOSITORY);
    var login = rep.Login("admin","admin");
    if (login) {
        var groupPath = "/newgroup";
        var newGroupName = "newgroup2";
        var result = rep.UpdateGroupName(groupPath, newGroupName);
        if (result == true) {
            _DEBUG("result="+result);
        } else {
            _DEBUG("result="+result+" ["+rep.GetLastErrorCode()+"]
"+rep.GetErrorMessageByCode(rep.GetLastErrorCode()));
        }
        rep.Logout();
    }
}
```

UpdateItemGroupAuth

아이템에 대한 그룹의 권한을 변경합니다.

Prototype

bool UpdateItemGroupAuth(string GroupPath, string ItemFullPath, string NewPermission)

Parameters

GroupPath	그룹 경로
ItemFullPath	아이템 전체 경로
NewPermission	권한 정보 None, View, Read, ReadWrite 중 하나의 값으로 설정

Reference

그룹 이름이 변경된 경우 true가 리턴되고, 실패한 경우 false가 리턴됩니다.

Example

```
function updateItemGroupAuth(){
    var rep = OZCreator.GetBuiltInObject(OZBuiltInObject.REPOSITORY);
    var login = rep.Login("admin","admin");
    if (login) {
        var groupPath = "/";
        var itemFullPath = "/parameter_test.odi";
        var newPermission = "Read";
        // None, View, Read, ReadWrite
        var result = rep.UpdateItemGroupAuth(groupPath, itemFullPath, newPermission);
        if (result == true) {
            _DEBUG("result="+result);
        } else {
            _DEBUG("result="+result+" ["+rep.GetLastErrorCode()+"]
"+rep.GetErrorMessageByCode(rep.GetLastErrorCode()));
        }
        rep.Logout();
    }
}
```

UpdateItemUserAuth

아이템에 대한 사용자의 권한을 변경합니다.

Prototype

bool UpdateItemUserAuth(string UserName, string ItemFullPath, string NewPermission)

Parameters

UserName	사용자 이름
ItemFullPath	아이템 전체 경로
NewPermission	권한 정보 None, View, Read, ReadWrite 중 하나의 값으로 설정

Reference

권한이 변경된 경우 true가 리턴되고, 실패한 경우 false가 리턴됩니다.

Example

```
function updateItemUserAuth(){
    var rep = OZCreator.GetBuiltInObject(OZBuiltInObject.REPOSITORY);
    var login = rep.Login("admin","admin");
    if (login) {
        var username = "admin";
        var itemFullPath = "/parameter_test.odi";
        var newPermission = "Read";
        // None, View, Read, ReadWrite
        var result = rep.UpdateItemUserAuth(username, itemFullPath, newPermission);
        if (result == true) {
            _DEBUG("result="+result);
        } else {
            _DEBUG("result="+result+"["+rep.GetLastErrorCode()+"]
"+rep.GetErrorMessageByCode(rep.GetLastErrorCode()));
        }
        rep.Logout();
    }
}
```

UpdateUserDescription

사용자 설명을 변경합니다.

Prototype

bool UpdateUserDescription(string UserName, string NewDescription)

Parameters

UserName	사용자 이름
NewDescription	사용자 설명

Reference

사용자 설명이 변경된 경우 true가 리턴되고, 실패한 경우 false가 리턴됩니다.

Example

```
function updateUserDescription(){
    var rep = OZCreator.GetBuiltInObject(OZBuiltInObject.REPOSITORY);
    var login = rep.Login("admin","admin");
    if (login) {
        var username = "admin";
        var newDescription = "new desc";
        var result = rep.UpdateUserDescription(username, newDescription);
        if (result == true) {
            _DEBUG("result="+result);
        } else {
            _DEBUG("result="+result+" ["+rep.GetLastErrorCode()+"]
"+rep.GetErrorMessageByCode(rep.GetLastErrorCode()));
        }
        rep.Logout();
    }
}
```

UpdateUserName

사용자 이름을 변경합니다.

Prototype

bool UpdateUserName(string UserName, string NewUserName)

Parameters

UserName	사용자 이름
NewUserName	변경할 사용자 이름

Reference

사용자 이름이 변경된 경우 true가 리턴되고, 실패한 경우 false가 리턴됩니다.

Example

```
function updateUserName(){
    var rep = OZCreator.GetBuiltInObject(OZBuiltInObject.REPOSITORY);
    var login = rep.Login("admin","admin");
    if (login) {
        var username = "newgroup2";
        var newUsername = "newgroup2_new";
        var result = rep.UpdateUserName(username, newUsername);
        if (result == true) {
            _DEBUG("result="+result);
        } else {
            _DEBUG("result="+result+" ["+rep.GetLastErrorCode()+"]
"+rep.GetErrorMessageByCode(rep.GetLastErrorCode()));
        }
        rep.Logout();
    }
}
```

UpdateUserPassword

사용자 패스워드를 변경합니다.

Prototype

bool UpdateUserPassword(string UserName, string OldPassword, string NewPassword)

Parameters

UserName	사용자 이름
OldPassword	변경전 사용자 패스워드
NewPassword	변경할 사용자 패스워드

Reference

사용자 패스워드가 변경된 경우 true가 리턴되고, 실패한 경우 false가 리턴됩니다.

Example

```
function updateUserPassword(){
    var rep = OZCreator.GetBuiltInObject(OZBuiltInObject.REPOSITORY);
    var login = rep.Login("admin","admin");
    if (login) {
        var username = "admin";
        var newPassword = "admin";
        var result = rep.UpdateUserPassword(username, "admin", newPassword);
        if (result == true) {
            _DEBUG("result="+result);
        } else {
            _DEBUG("result="+result+" ["+rep.GetLastErrorCode()+"]
"+rep.GetErrorMessageByCode(rep.GetLastErrorCode()));
        }

        var result = rep.UpdateUserPassword(username, "admin", "admin");
        rep.Logout();
    }
}
```

UploadFile

아이템을 업로드합니다.

Prototype

bool UploadFile(string ItemFullPath, string ServerFullPath, [bool KeepCheckout], string Comment)

bool UploadFile(array ItemFullPaths, array ServerFullPaths, array ErrorCodes, [array KeepCheckouts], string Comment)

bool UploadFile(object Item, string ServerFullPath, [bool KeepCheckout], string Comment)

bool UploadFile(array Items, array ServerFullPaths, array ErrorCodes, [array KeepCheckouts], string Comment)

Parameters

ItemFullPath	아이템 전체 경로
ServerFullPath	서버 전체 경로
KeepCheckout	체크아웃 상태 유지 여부
Comment	주석
ItemFullPaths	아이템 전체 경로를 가진 배열
ServerFullPaths	서버 전체 경로를 가진 배열
ErrorCodes	에러 코드를 가져올 배열
KeepCheckouts	체크아웃 상태 유지 여부를 가진 배열
Item	아이템 오즈 바이너리 객체로 설정하여야 함
Items	아이템을 가진 배열 오즈 바이너리 객체를 가진 배열로 설정하여야 함

Reference

아이템이 업로드된 경우 true가 리턴되고, 실패한 경우 false가 리턴됩니다.

에러가 발생한 아이템의 인덱스에 에러 코드 값을 리턴하며, 성공한 경우에는 에러 코드 값에 0을 리턴합니다.

Example 1

```
function uploadFile(){
    var rep = OZCreator.GetBuiltInObject(OZBuiltInObject.REPOSITORY);
    var login = rep.Login("admin","admin");
    if (login) {
        var result = rep.UploadFile("c:/parameter_test.odi", "/data/parameter_test.odi", true,
"comment");
        if (result == true) {
            _DEBUG("result="+result);
        } else {
            _DEBUG("result="+result+" ["+rep.GetLastErrorCode()+"]
"+rep.GetErrorMessageByCode(rep.GetLastErrorCode()));
        }
    }
}
```

```
        rep.Logout();
    }
}
```

Example 2

```
function uploadFile(){
    var rep = OZCreator.GetBuiltInObject(OZBuiltInObject.REPOSITORY);
    var login = rep.Login("admin","admin");
    if (login) {
        var itemFullPaths = new Array();
        itemFullPaths[0] = "c:/parameter_test.odi";
        itemFullPaths[1] = "c:/parameter_test.ozr";
        var serverFullPaths = new Array();
        serverFullPaths[0] = "/data/parameter_test.odi";
        serverFullPaths[1] = "/report/parameter_test.ozr";
        var errorCodes = new Array();
        var keepCheckouts = new Array();
        keepCheckouts[0] = true;
        keepCheckouts[1] = true;
        var result = rep.UploadFile(itemFullPaths, serverFullPaths, errorCodes, keepCheckouts,
"comment");
        if (result == true) {
            _DEBUG("result="+result);
        } else {
            for (i=0; i < errorCodes.length; i++) {
                _DEBUG("i="+i+" ["+errorCodes[i]+"
"+rep.GetErrorMessageByCode(errorCodes[i]));
            }
        }
        rep.Logout();
    }
}
```

Example 3

```
function uploadFile(){
    var rep = OZCreator.GetBuiltInObject(OZBuiltInObject.REPOSITORY);
    var login = rep.Login("admin","admin");
    if (login) {
        var item = OZCreator.GetBuiltInObject(OZBuiltInObject.BINARY);
        item.ReadFile("c:/parameter_test.odi");
    }
}
```

```
var serverFullPath = "/data/parameter_test.odi";
var keepCheckout = true;
var result = rep.UploadFile(item, serverFullPath, keepCheckout, "comment");
if (result == true) {
    _DEBUG("result="+result);
} else {
    _DEBUG("result="+result+" ["+rep.GetLastErrorCode()+"]
"+rep.GetErrorMessageByCode(rep.GetLastErrorCode()));
}
rep.Logout();
}
```

Example 4

```
function uploadFile1(){
    var rep = OZCreator.GetBuiltInObject(OZBuiltInObject.REPOSITORY);
    var login = rep.Login("admin","admin");
    if (login) {
        var items = new Array();
        items[0] = OZCreator.GetBuiltInObject(OZBuiltInObject.BINARY);
        items[0].ReadFromFile("c:/parameter_test.odi");
        items[1] = OZCreator.GetBuiltInObject(OZBuiltInObject.BINARY);
        items[1].ReadFromFile("c:/parameter_test.ozr");
        var serverFullPaths = new Array();
        serverFullPaths[0] = "/data/parameter_test.odi";
        serverFullPaths[1] = "/report/parameter_test.ozr";
        var errorCodes = new Array();
        var keepCheckouts = new Array();
        keepCheckouts[0] = true;
        keepCheckouts[1] = true;
        var result = rep.UploadFile(items, serverFullPaths, errorCodes, keepCheckouts);
        if (result == true) {
            _DEBUG("result="+result);
        } else {
            for (i=0; i< errorCodes.length; i++) {
                _DEBUG("i="+i+" ["+errorCodes[i]+"]
"+rep.GetErrorMessageByCode(errorCodes[i]));
            }
        }
        rep.Logout();
    }
}
```

```
}  
}
```

See Also

[OZBinary](#)

OZProcedureParameter

- [Property](#)

Property

Name	Description
Name	패러미터 이름
DataType	패러미터 데이터 타입
DataTypeName	패러미터 데이터 타입의 문자형 값
Direction	출력 패러미터 방향
IsNull	널 적용 여부
Selected	출력 패러미터 선택 여부
Value	입력 패러미터 값

Name

패러미터 이름

Prototype

[string](#) [Name](#)

Example

```
var ProcedureParameter = This.GetProcedureParameter("param1");  
_DEBUG("Name " + ProcedureParameter.Name);
```

DataType

패러미터 데이터 타입의 문자형 값

Prototype

integer **DataType**

Example

```
var ProcedureParameter = This.GetProcedureParameter("param1");  
_DEBUG("DataType " + ProcedureParameter.DataType);
```

See Also

[DataFieldTypeConst](#)

DataTypeName

패러미터 데이터 타입의 문자형 값

Prototype

string **DataTypeName**

Example

```
var ProcedureParameter = This.GetProcedureParameter("param1");  
_DEBUG("DataTypeName " + ProcedureParameter.DataTypeName);
```

Direction

출력 패러미터 방향

Prototype

integer **Direction**

Example

```
var ProcedureParameter = This.GetProcedureParameter("param1");  
_DEBUG("Direction " + ProcedureParameter.Direction);
```

See Also

[ProcedureDirectionConst](#)

IsNull

널 적용 여부

Prototype

bool IsNull

Example

```
var ProcedureParameter = This.GetProcedureParameter("param1");  
_DEBUG("IsNull " + ProcedureParameter.IsNull);
```

Selected

출력 패러미터 선택 여부

Prototype

bool Selected

Example

```
var ProcedureParameter = This.GetProcedureParameter("param1");  
_DEBUG("Selected " + ProcedureParameter.Selected);
```

Value

입력 패러미터 값

Prototype

string Value

Example

```
var ProcedureParameter = This.GetProcedureParameter("param1");  
_DEBUG("Value " + ProcedureParameter.Value);
```

Constant

- [DataFieldTypeConst](#)
- [ODIClientDMTypeConst](#)
- [ODIFetchTypeConst](#)
- [ODIFetchUnitConst](#)
- [ODIServerDMTypeConst](#)
- [OZJSResultSetConst](#)
- [OZServerTypeConst](#)
- [ProcedureDirectionConst](#)
- [ProcedureTypeConst](#)
- [RecipientTypeConst](#)
- [SearchOptionConst](#)
- [ShowZeroDecimalPointConst](#)

DataFieldTypeConst

Name	Value	Description
BIGINT	BIGINT	BIGINT 타입
BINARY	BINARY	BINARY 타입
BIT	BIT	BIT 타입
BLOB	BLOB	BLOB 타입
CHAR	CHAR	CHAR 타입
CLOB	CLOB	CLOB 타입
DATE	DATE	DATE 타입
DECIMAL	DECIMAL	DECIMAL 타입
DOUBLE	DOUBLE	DOUBLE 타입
FLOAT	FLOAT	FLOAT 타입
INTEGER	INTEGER	INTEGER 타입
LONGVARBINARY	LONGVARBINARY	LONGVARBINARY 타입
LONGVARCHAR	LONGVARCHAR	LONGVARCHAR 타입

NUMERIC	NUMERIC	NUMERIC 타입
REAL	REAL	REAL 타입
SMALLINT	SMALLINT	SMALLINT 타입
TIME	TIME	TIME 타입
TIMESTAMP	TIMESTAMP	TIMESTAMP 타입
TINYINT	TINYINT	TINYINT 타입
VARBINARY	VARBINARY	VARBINARY 타입
VARCHAR	VARCHAR	VARCHAR 타입
OTHER	OTHER	기타 타입

ODIClientDMTypeConst

Name	Value	Description
DM_CLIENT_USE_NORMAL	1	서버에 설정된 속성 값에 따름
DM_CLIENT_USE_MEMORY	2	클라이언트 메모리 사용
DM_CLIENT_USE_FILE	3	클라이언트 파일 사용

ODIFetchTypeConst

Name	Value	Description
DM_CONCURRENT_FETCH	16	CONCURRENT 타입
DM_BATCH_FETCH	17	BATCH 타입

ODIFetchUnitConst

Name	Value	Description
DM_PER_DATAMODULE	380	데이터 모듈 단위
DM_PER_DATASET	383	데이터셋 단위

ODIServerDMTypeConst

Name	Value	Description
DM_SERVER_USE_MEMORY	32	서버 메모리 사용
DM_SERVER_USE_FILE	33	서버 파일 사용

OZJSResultSetConst

Name	Value	Description
CLOSE_CURSORS_AT_COMMIT	2	Commit이 되었을 때 결과셋을 Close함
CONCUR_READ_ONLY	1007	동시적으로 Update된 경우 반영 안 함
CONCUR_UPDATABLE	1008	동시적으로 Update된 경우 반영함
HOLD_CURSORS_OVER_COMMIT	1	Commit이 되었을 때 결과셋을 Close 안 함
TYPE_FORWARD_ONLY	1003	커서를 앞으로만 이동 가능
TYPE_SCROLL_INSENSITIVE	1004	커서를 앞뒤로 이동 가능, 변경사항 체크 못함
TYPE_SCROLL_SENSITIVE	1005	커서를 앞뒤로 이동 가능, 변경사항 체크 가능

OZServerTypeConst

Name	Value	Description
Daemon	1	데몬 서버
Servlet	2	서블릿 서버
DotNet	3	닷넷 서버

ProcedureDirectionConst

Name	Value	Description
Input	1	입력
InputOutput	3	입출력

Output	2	출력
ReturnValue	4	리턴 값

ProcedureTypeConst

Name	Value	Description
Function	2	함수
Procedure	1	프로시저

RecipientTypeConst

Name	Value	Description
BCC	3	숨은 참조
CC	2	참조
TO	1	받는 사람

SearchOptionConst

Name	Value	Description
AllCheckoutItem	256	체크아웃된 모든 아이템
CheckItemName	16	아이템 이름
Default	4112	기본 값 CheckItemName, IncludeSubCategory가 조합된 옵션
IncludeCheckinDate	8192	체크인 날짜
IncludeComment	64	아이템 주석
IncludeDescription	32	아이템 설명
IncludeHistory	1024	히스토리 정보
IncludeSubCategory	4096	하위 카테고리 포함
MatchCase	2	대소문자 구분

MatchWholeWord	1	단어 단위
None	0	검색 옵션 없음
SpecificCheckoutItem	512	지정한 사용자가 체크아웃한 아이템
UseRegEx	8	정규식 사용
UseWildcard	4	와일드카드 사용

ShowZeroDecimalPointConst

Name	Value	Description
Default	0	DB 서버에 설정된 속성 값에 따름
True	1	DB 서버에 설정된 속성 값에 상관없이 소수점 앞에 무조건 0을 표시
False	2	DB 서버에 설정된 속성 값에 상관없이 소수점 앞에 무조건 0을 표시하지 않음

Reference

디자이너 환경 설정 방법, 리파지토리 서버 사용 방법 등에 대해 설명합니다.

디자이너 환경 설정

디자이너의 환경을 설정하는 방법에 대해 설명합니다.

작업 폴더 설정

작업 폴더는 리파지토리 클라이언트가 리파지토리 서버로부터 파일을 복사받아 저장하는 로컬 작업 폴더입니다. 작업 폴더는 리파지토리를 이용한 파일의 공유 과정에서 매우 중요한 역할을 담당하는 부분입니다.

작업 폴더 기본 값

오즈 쿼리 디자이너가 처음 실행되면 로그인한 사용자 계정의 내 문서 폴더의 OZRepository라는 폴더가 작업 폴더 기본 값으로 설정됩니다. 만일 해당 디렉토리에 OZRepository라는 폴더가 없으면 오즈 쿼리 디자이너가 실행될 때 자동으로 생성됩니다.

작업 폴더 설정 메뉴로 작업 폴더 설정

리파지토리 창에서 작업 폴더로 설정할 폴더를 선택한 후 마우스 오른쪽 버튼을 클릭하여 나타나는 팝업 메뉴에서 **[작업 폴더 설정]** 메뉴를 클릭하면 해당 폴더가 작업 폴더로 설정됩니다.

로컬 탐색기에서 작업 폴더를 설정할 경우 작업 폴더는 "작업 폴더 [작업 폴더 위치]"의 형태로 표시됩니다. 만일 선택한 경로가 작업 폴더 매니저에 등록되어 있는 경로일 경우 작업 폴더 이름과 작업 폴더 경로가 함께 표시됩니다.

예를 들어 "C:/Repository" 경로가 "Working folder1"로 작업 폴더 매니저에 등록되어 있을 경우 리파지토리 창에서 작업 폴더를 설정하면 "작업 폴더 (Working folder1) [C:WRepository]" 형태로 작업 폴더 이름과 경로가 함께 표시됩니다.

작업 폴더 매니저로 작업 폴더 설정

작업 폴더 매니저는 로컬 탐색기에서 여러 개의 작업 폴더를 추가하여 관리할 수 있는 관리 도구입니다.

리파지토리 창에서 작업 폴더를 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 **[작업 폴더 매니저]** 메뉴를 클릭하여 실행합니다.

작업 폴더 매니저 다이얼로그에서 **[가져오기]** 버튼을 클릭하면 작업 폴더 리스트 정보를 저장하고 있는 XML 파일을 가져와서 작업 폴더 리스트를 표시하며, **[내보내기]** 버튼을 클릭하면 현재 다이얼로그에 추가되어 있는 작업 폴더 리스트 정보를 XML 파일로 저장합니다.

작업 폴더 매니저에서 작업 폴더를 설정할 경우 작업 폴더는 "작업 폴더 (작업 폴더 이름) [작업 폴더 위치]"의 형태로 표시됩니다.

디자이너 기본 옵션 설정

레이아웃, 언어 설정, 서버 설정 옵션 등 오즈 쿼리 디자이너의 기본 옵션을 설정합니다.

[파일] > [옵션] 메뉴를 클릭하여 나타나는 "옵션 다이얼로그"를 이용하여 설정합니다. "옵션 다이얼로그"에서 변경한 값을 기본값으로 변경하려면 [기본값으로 변경] 버튼을 클릭합니다.

일반

결과 데이터셋 최대 행 수 제한 쿼리 실행 결과로 표시되는 데이터의 최대 행 수의 제한 여부를 설정하고 최대 행 수를 입력합니다.

"결과 데이터셋 최대 행 수 제한" 옵션은 쿼리 창의 [결과] 탭에만 적용됩니다

레이아웃

디자인 모드 디자인 모드를 설정합니다.

디자인 뷰: 디자인 뷰 모드로 설정합니다.

매뉴얼: 매뉴얼 모드로 설정합니다.

종료할 때 윈도우 배치 저장 오즈 쿼리 디자이너 종료 시 윈도우 배치를 저장합니다.

윈도우 배치 초기화 윈도우의 배치를 초기화합니다.

현재 윈도우 배치 저장 현재 윈도우 배치를 저장합니다.

언어 설정

언어 언어를 설정합니다.

한국어: 한국어로 설정합니다.

일본어: 일본어로 설정합니다.

중국어: 중국어로 설정합니다.

영어: 영어로 설정합니다.

재 설정 현재 실행 중인 언어로 다시 설정합니다.

저장

자동 저장 작업 중인 ODI 파일을 백업하는 옵션으로 지정된 시간마다 백업 파일을 생성합니다

1 ~ 60 사이의 값으로 설정합니다.

파일 저장 시 암호화 보고서 파일 저장 시 암호화하여 저장할지 여부를 설정합니다.
파일 저장 시 암호화 기능은 특정 사이트에서만 파일을 열 수 있도록 파일을 암호화하는 기능으로, 해당 기능에 대한 자세한 사항은 오즈 기술 지원 센터로 문의하시기 바랍니다.

서버 설정

최근 접속한 리파지토리 서버로 자동 연결 디자이너 실행 시 최근 접속한 리파지토리 서버로 자동 연결합니다.

로컬 서버 타입 오즈 쿼리 디자이너의 로컬 서버 타입을 설정합니다.

자바: 자바 서버로 설정합니다.

닷넷: 닷넷 서버로 설정합니다.

서버 연결 위치 쿼리문 실행 시 데이터를 바인딩할 서버의 위치를 설정합니다.
OpenType: ODI 파일을 열기한 위치에 따라 연결할 서버를 설정합니다.

Local: 로컬 서버로 설정합니다.

OZServer: 연결되어있는 오즈 서버로 설정됩니다.

ForceLocal: 무조건 로컬 서버로 설정합니다.

ForceOZServer: 연결되어 있는 오즈 서버로 설정합니다.

ForceOpenType: ODI 파일을 열기한 위치에 따라 연결할 서버를 설정합니다.

"ForceLocal" 값을 제외한 나머지 값은 오즈 데이터 트리의 [서버 연결 위치](#) 속성 값에 따라 다르게 설정될 수 있습니다.

데이터베이스 옵션 설정

데이터베이스 옵션, SQL 함수 리스트 옵션을 설정합니다.

[파일] > [옵션] 메뉴를 클릭하여 나타나는 "옵션 다이얼로그"를 이용하여 설정합니다.

일반

조인 조건 자동 감지 데이터베이스에 설정된 테이블 간의 조인을 자동으로 감지하도록 설정합니다.

로컬에 캐시된 테이블 정보 사용

최초 데이터베이스 접속 시 가져온 테이블 정보를 로컬 캐시에 저장하고 다음 접속 시부터는 로컬에 저장된 테이블 정보를 사용할 것인지 여부를 설정합니다.

로컬 캐시에서 테이블 정보 로드 시 확인: 로컬 캐시에서 테이블 정보 로딩 시 확인 창을 띄울 것인지 여부를 설정합니다.

데이터베이스 연결 시 자동으로 테이블 정보 가져옴 데이터베이스 접속 시 테이블 정보를 자동으로 가져올지 여부를 설정합니다.

SQL 함수 리스트

데이터베이스 벤더 별 SQL 함수 리스트 파일 설정 데이터베이스 벤더 별 SQL 함수 리스트 파일 설정 여부를 설정합니다.

데이터베이스 벤더: 데이터베이스 종류를 선택합니다.

SQL 함수 리스트 파일: 가져올 SQL 함수 리스트 파일을 설정합니다.

참고 사항

데이터베이스 별 SQL 함수 리스트는 쿼리 마법사에서 사용하며, 파일을 설정하지 않을 경우 해당 데이터베이스의 JDBC Driver에서 지원하는 함수 이름이 자동으로 사용됩니다. 그러나 JDBC Driver에 따라 함수명이 정확하지 않을 수도 있습니다.

SQL 함수 리스트 파일 작성 예는 다음과 같습니다.

```
ASCII REPLACE
SUM MIN
USER DUMP
SYSDATE LAST_DAY NEXT_DAY
```

각 함수들은 공백 문자(Space 혹은 Tab)으로 구분되며, 라인은 4개입니다. 또한 함수 개수에 관계없이 1라인은 스트링 함수, 2라인은 수학 함수, 3라인은 시스템 함수, 4라인은 시간 함수를 표현합니다.

편집기 옵션 설정

매뉴얼 모드의 쿼리문 입력 창, 디자인 뷰 모드의 쿼리문 표시 창, 스크립트 창의 텍스트의 글꼴, 전경색, 배경색 등을 설정합니다.

[파일] > [옵션] 메뉴를 클릭하여 나타나는 "옵션 다이얼로그"를 이용하여 설정합니다.

환경 설정

- 구문 강조 기능** 매뉴얼 모드로 쿼리를 작성할 경우 입력한 문자열을 문법 체크를 하여 키워드 별로 텍스트 색을 달리하여 표시할지 여부를 설정합니다.
- 스크립트 도움말 툴팁보기** 스크립트 창에서 함수의 프로토타입을 툴팁으로 표시할지 여부를 설정합니다
.
- 줄 번호 표시** 줄 번호를 표시합니다.

글꼴과 색상

- 글꼴** 텍스트의 글꼴을 설정합니다.
- 크기** 텍스트의 크기를 설정합니다.
- 글꼴 스타일** 텍스트의 글꼴 스타일을 설정합니다.
- 전경색** 전경색을 설정합니다.
- 배경색** 배경색을 설정합니다.

리파지토리 서버 사용

리파지토리 서버는 ODI, OZR, OZA, 데이터 파일 등 각종 파일을 저장하는 원격 저장소로, 다수의 개발자가 원격으로 파일을 공유할 수 있으며, 각 사용자마다 접근 권한을 달리할 수 있습니다. 또한 파일 버전 관리 기능을 통해 개발의 효율을 높여줍니다.

리파지토리 서버 설정

1. 리파지토리 창의 **[리파지토리 서버]**를 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 **[리파지토리 열기]** 메뉴를 클릭하면 저장된 리파지토리 서버 리스트와 접속 정보를 나타내는 "서버 리스트 다이얼로그"가 나타납니다.
2. 서버 리스트 다이얼로그에서 서버 이름, 종류 및 접속 정보를 설정한 후 **[저장]** 버튼을 클릭합니다.
3. 서버 리스트 다이얼로그에서 **[가져오기]** 버튼을 클릭하면 서버 리스트 정보를 저장하고 있는 XML 파일을 가져와서 서버 리스트를 표시하며, **[내보내기]** 버튼을 클릭하면 현재 다이얼로그에 추가되어 있는 서버 리스트 정보를 XML 파일로 저장합니다.

리파지토리 서버 연결, 연결 해제

리파지토리 서버 연결

리파지토리 서버에 접속합니다.

1. 리파지토리 창의 **[리파지토리 서버]**를 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 **[리파지토리 열기]** 메뉴를 클릭한 후 "서버리스트 다이얼로그"에서 연결할 서버를 더블 클릭합니다.
2. 리파지토리 서버 연결이 완료되면 리파지토리 창에 리파지토리 서버 폴더 및 파일이 표시됩니다.

리파지토리 서버 연결 해제

리파지토리 서버 연결을 해제합니다.

1. 접속된 리파지토리 서버를 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴에서 **[리파지토리 닫기]** 메뉴를 클릭하면 접속이 해제됩니다.

리파지토리 서버로부터 파일 열기

리파지토리 서버에 연결된 상태에서는 리파지토리 서버에 저장된 ODI 파일을 직접 열어 편집할 수 있습니다.

1. **[파일]** > **[열기]** 메뉴나 툴바의 열기 아이콘을 클릭하면 "리파지토리 파일 다이얼로그"가 나타납니다.
2. 파일을 리파지토리 서버에서 선택한 후 **[확인]** 버튼을 클릭합니다.

주의 사항

파일을 서버로부터 열기할 때 다운로드하여 로컬의 작업 폴더에 저장한 경우에는 **[파일]** 메뉴의 **[저장]** 메뉴나 툴바의 **[저장 아이콘]**을 클릭하여 저장하는 등의 작업은 로컬의 파일에 저장시키는 것입니다. 즉 다시 파일을 서버로부터 열기를 하면 리파지토리 서버에 있는 파일이 열리기 때문에 로컬의 작업 폴더에 저장한 내용과 다를 수 있습니다.

서버로부터 열기한 파일의 편집한 내용을 리파지토리 서버에 반영하기 위해서는 반드시 체크아웃을 먼저하고 내용을 편집한 다음 체크인을 하는 과정을 거쳐야 합니다.

서버로부터 열기한 파일이 체크아웃 상태에서 편집을 마친 후에는 반드시 체크인하거나 체크아웃을 취소하여야 합니다.

카테고리, 아이템 관리

카테고리 추가

1. 카테고리를 추가하고자 하는 카테고리에서 마우스 오른쪽 버튼을 클릭하여 나타나는 팝업 메뉴에서 **[카테고리 생성]** 메뉴를 클릭합니다.
2. "카테고리 추가 다이얼로그"가 나타나면 추가하고자 하는 카테고리명을 입력한 후 **[확인]** 버튼을 클릭하면 새로운 카테고리가 만들어집니다.

파일 업로드

1. 작업 폴더에서 업로드할 파일을 선택한 후 리파지토리 서버의 원하는 카테고리로 드래그 & 드롭하면 "아이템 업로드 다이얼로그"가 표시되고 업로드할 아이템 리스트를 보여줍니다.
2. **[업로드]** 버튼을 클릭하게 되면 선택된 아이템이 리파지토리 서버의 지정한 카테고리에 추가되게 됩니다. 한 개 이상의 파일을 선택할 때에는 **<Ctrl>** 키나 **<Shift>** 키를 클릭한 상태에서 원하는 파일을 마우스로 클릭합니다.
3. 원하는 파일들을 선택한 후 드래그하여 추가하고자 하는 카테고리에 가져가면 선택된 아이템 리스트가 기록된 "아이템 업로드 다이얼로그"가 나타납니다.
4. 추가하고자 하는 아이템을 선택한 후 **[업로드]** 버튼을 클릭하면 선택된 모든 아이템이 추가됩니다.

5. 이때 선택된 아이템 중 특정 아이템을 제외하고자 하는 경우 "아이템 업로드 다이얼로그"에 나타난 아이템 리스트 중에서 이름에서의 체크 박스에 체크된 부분을 지우면 특정 아이템을 제외하고 추가할 수 있습니다.

파일 다운로드

1. 리파지토리 서버에서 다운로드할 파일을 선택한 후 마우스 오른쪽 버튼을 클릭하여 나타나는 팝업 메뉴에서 **[다운로드]** 메뉴를 클릭합니다.
2. "아이템 다운로드 다이얼로그"에서 다운로드할 아이템을 선택한 후 **[다운로드]** 버튼을 클릭하여 로컬 시스템의 작업 폴더로 다운로드 합니다.
3. "아이템 다운로드 다이얼로그"에서 "체크아웃된 아이템 선택" 옵션을 체크하면 현재 체크 아웃된 아이템 (다른 사용자가 체크 아웃한 아이템 포함)에 대해서도 다운로드를 할 수 있습니다.
4. 로컬 시스템의 작업 폴더 내에 동일한 이름의 파일이 있을 경우에는 업데이트 할 것인지 묻는 확인 창이 나타나고 그렇지 않을 경우에는 바로 다운로드 됩니다.

카테고리, 파일 삭제

1. 삭제하고자 하는 카테고리 또는 파일을 마우스 오른쪽 버튼을 클릭하여 나타나는 팝업 메뉴에서 **[삭제]** 메뉴를 선택하면 "아이템 삭제 다이얼로그"가 표시되고 삭제되는 아이템 리스트를 나타냅니다.
2. 삭제할 아이템을 선택한 후 **[삭제]** 버튼을 클릭하면 리파지토리 서버에서 해당 아이템이나 카테고리가 삭제됩니다.

이름 변경

1. 이름을 변경하고자 하는 카테고리 또는 아이템에서 마우스 오른쪽 버튼을 클릭하여 나타나는 팝업 메뉴에서 **[이름 변경]** 메뉴를 클릭하면 해당 카테고리 또는 아이템의 이름을 변경할 수 있는 "카테고리 이름 변경 다이얼로그"나 "아이템 이름 변경 다이얼로그"가 나타납니다.
2. 해당 창에 변경하고자 하는 이름을 입력하고 **[확인]** 버튼을 클릭하면 아이템이나 카테고리 이름이 변경 됩니다.

체크인, 체크아웃

오즈 리파지토리는 파일의 안전한 공유와 갱신을 위해 하나의 파일이 한 명 이상의 개발자에 의해 동시에 수정되거나 갱신되는 것을 허용하지 않습니다. 이를 위해 리파지토리 서버는 어떤 순간에도 각각의 파일에 대한 수정 권한을 오직 한 명의 개발자에게만 부여하고, 리파지토리 서버가 관리하는 파일을 갱신하기 위해서는 반드시 리파지토리 서버로부터 갱신 권한을 받도록 제한합니다.

오즈 리파지토리가 정의하는 원칙에 따라 리파지토리 서버의 파일을 수정하고 갱신할 수 있는 권한을 받는 과정을 체크아웃이라 하고, 반대로 수정된 파일을 리파지토리에 등록하고 갱신 권한을 반납하는 과정을 체크인이라고 합니다.

체크인

체크인은 리파지토리 서버에서 체크아웃 후 수정된 내용을 리파지토리 서버에 저장하고 갱신 권한을 반납하는 기능입니다.

1. 리파지토리 창에서 체크인할 파일을 선택한 후 마우스 오른쪽 버튼을 클릭하여 나타나는 팝업 메뉴에서 **[체크인]** 메뉴를 클릭합니다.
2. "아이템 체크인 다이얼로그"에서 체크인할 파일을 체크한 후 **[체크인]** 버튼을 클릭하여 체크인 합니다.
3. 만일 체크아웃 상태인 파일을 수정하였을 경우 수정된 내용을 리파지토리 서버에 저장하고 갱신 권한을 반납하지 않고자 할 경우에는 "체크아웃 상태유지" 옵션을 체크한 후 **[체크인]** 버튼을 클릭하시기 바랍니다. 이 옵션을 이용하면 체크아웃 상태인 파일을 리파지토리 서버에 저장하고 갱신 권한을 계속 갖고자 할 경우 체크인을 하고 다시 체크아웃을 해야 하는 번거로움을 줄일 수 있습니다.

체크아웃

체크아웃은 리파지토리 서버에 저장된 파일의 수정을 위해 리파지토리 서버로부터 특정 파일에 대한 갱신 권한을 얻어오는 기능입니다.

1. 리파지토리 창에서 체크아웃할 파일을 선택한 후 마우스 오른쪽 버튼을 클릭하여 나타나는 팝업 메뉴에서 **[체크아웃]** 메뉴를 클릭하고, "아이템 체크아웃 다이얼로그"에서 체크아웃할 파일을 체크한 후 **[체크아웃]** 버튼을 클릭하여 체크아웃 합니다.

체크아웃 취소

체크아웃 취소는 체크아웃된 파일의 수정 내용을 리파지토리 서버에 반영하지 않고 단순히 갱신 권한을 반납하는 기능입니다.

1. 리파지토리 창에서 체크아웃 취소할 파일을 선택한 후 마우스 오른쪽 버튼을 클릭하여 나타나는 팝업 메뉴에서 **[체크아웃 취소]** 메뉴를 클릭하고, "아이템 체크아웃 취소 다이얼로그"에서 체크아웃 취소할 파일을 체크한 후 **[체크아웃 취소]** 버튼을 클릭하여 체크아웃을 취소합니다.

히스토리 보기

아이템을 체크인 또는 체크아웃하여 각 아이템별로 히스토리를 남길 경우 히스토리 보기 기능을 사용하여 아이템의 내역을 볼 수 있습니다.

1. 리파지토리 서버에 있는 아이템을 선택한 후 마우스 오른쪽 버튼을 클릭하여 나타나는 팝업 메뉴에서 **[히스토리 보기]** 메뉴를 클릭합니다.
2. **[히스토리 보기]** 메뉴를 클릭하면 선택한 아이템의 히스토리를 나타낸 다이얼로그가 실행됩니다.
3. 버전은 아이템의 히스토리 버전으로 아이템 체크인 시에 자동으로 부여되며, 낮은 숫자일수록 오래된 버전입니다.
4. 특정 버전을 선택하여 다운로드, 주석 조회, 롤백, 히스토리 삭제 등의 기능을 실행할 수 있으며, 두 개의 버전을 선택하여 버전 간 스크립트를 비교할 수 있습니다.

아이템 찾기

1. 리파지토리 서버에서 검색할 카테고리를 선택한 후 마우스 오른쪽 버튼을 클릭하여 나타나는 팝업 메뉴에서 **[아이템 찾기]** 메뉴를 클릭합니다.
2. **[아이템 찾기]** 메뉴를 클릭하면 "아이템 찾기 다이얼로그"가 실행되고, 찾을 경로에는 선택한 카테고리가 자동으로 표시됩니다. 찾을 내용, 기본 옵션, 아이템 옵션 등을 설정한 후 **[찾기]** 버튼을 클릭하면 검색된 아이템이 리스트 형태로 표시됩니다.
3. 검색된 아이템을 마우스 오른쪽 버튼으로 클릭하여 나타나는 팝업 메뉴를 이용하여 체크인, 체크아웃, 체크아웃 취소 등의 기능을 실행할 수 있습니다.

주의 사항

버전 정보는 "주석", "아이템 히스토리 포함" 옵션이 체크된 경우에만 버전 정보를 가져와서 표시하며, 다른 경우에는 버전 정보를 표시하지 않습니다. 버전의 숫자가 낮을수록 오래된 버전이며, 버전 정보가 있는 히스토리 아이템의 경우 해당 아이템의 팝업 메뉴를 띄우면 **[다운로드]** 메뉴와 **[주석]** 메뉴만 활성화됩니다. 이력 정보 기능을 사용하지 않는 일반적인 None 타입 서버의 경우 체크인, 체크아웃 등 히스토리 기능을 지원하지 않는 서버이므로, "아이템 찾기 다이얼로그"에서 "아이템 옵션"이 모두 비활성화되어 표시됩니다. 아이템 찾기 기능은 리파지토리 서버에서만 사용 가능합니다. (로컬 작업 폴더에서는 사용할 수 없습니다.) 주석으로 찾을 경우 검색 옵션 중 "아이템 히스토리 포함" 옵션을 반드시 체크하시기 바랍니다.

파일 경로 설정

오즈 쿼리 디자이너에서 스토어 추가 시 앨리어스 정보 파일(db.properties, datasource.properties), HTTP/파일 스토어의 XML, CSV 파일 등의 경로를 설정하는 방법에 대해 설명합니다.

열기 다이얼로그로 설정

1. 파일 경로 선택 버튼을 클릭하여 표시되는 "열기 다이얼로그"에서 파일을 선택합니다.
2. 예를 들어 데이터베이스 연결 정보의 앨리어스 정보 파일을 설정할 경우 파일 경로 선택 버튼[...]을 클릭 하면 "열기 다이얼로그"가 표시됩니다.
3. "열기 다이얼로그"에서 선택한 드라이브(ex. C:\w)가 디자이너가 설치된 드라이브(ex. C:\w)와 동일한 경우 자동으로 상대 경로로 표시되며, 다른 드라이브일 경우에는 절대 경로로 표시됩니다.

직접 입력하여 설정

파일 경로는 절대 경로 또는 상대 경로, 오즈 URL로 사용자가 직접 입력하여 설정할 수 있습니다.

절대 경로	C:\wconf\wdb.properties
상대 경로	./db.properties
오즈 URL	ozp://conf/db.properties

문자셋과 유효한 앨리어스

데이터베이스 연결 정보의 인코딩, 디코딩 셋에 들어 들어가는 문자셋 이름입니다. 각 문자셋 이름과 해당 문자셋과 같은 의미를 가지는 앨리어스 이름은 다음과 같습니다.

문자셋 이름	유효한 앨리어스 이름
ASCII	us-ascii, ascii, 646, iso646-us, default, ascii7
Big5	big5
Cp037	ibm037, ibm-037, cp037, 037
Cp1006	ibm1006, ibm-1006, cp1006, 1006
Cp1025	ibm1025, ibm-1025, cp1025, 1025
Cp1026	ibm1026, ibm-1026, cp1026, 1026
Cp1097	ibm1097, ibm-1097, cp1097, 1097
Cp1098	ibm1098, ibm-1098, cp1098, 1098
Cp1112	ibm1112, ibm-1112, cp1112, 1112
Cp1122	ibm1122, ibm-1122, cp1122, 1122
Cp1123	ibm1123, ibm-1123, cp1123, 1123
Cp1124	ibm1124, ibm-1124, cp1124, 1124
Cp1250	windows-1250
Cp1251	windows-1251
Cp1252	windows-1252
Cp1253	windows-1253
Cp1254	windows-1254
Cp1255	windows-1255
Cp1256	windows-1256
Cp1257	windows-1257
Cp1258	windows-1258

Cp1381	ibm1381, ibm-1381, cp1381, 1381
Cp1383	ibm1383, ibm-1383, cp1383, 1383
Cp273	ibm273, ibm-273, cp273, 273
Cp277	ibm277, ibm-277, cp277, 277
Cp278	ibm278, ibm-278, cp278, 278
Cp280	ibm280, ibm-280, cp280, 280
Cp284	ibm284, ibm-284, cp284, 284
Cp285	ibm285, ibm-285, cp285, 285
Cp297	ibm297, ibm-297, cp297, 297
Cp33722	ibm33722, ibm-33722, cp33722, 33722
Cp420	ibm420, ibm-420, cp420, 420
Cp424	ibm424, ibm-424, cp424, 424
Cp437	ibm437, ibm-437, cp437, 437
Cp500	ibm500, ibm-500, cp500, 500
Cp737	ibm737, ibm-737, cp737, 737
Cp775	ibm775, ibm-775, cp775, 775
Cp838	ibm838, ibm-838, cp838, 838
Cp850	ibm850, ibm-850, cp850, 850, cspc850multilingual
Cp852	ibm852, ibm-852, cp852, 852, csppcp852
Cp855	ibm855, ibm-855, cp855, 855, csppcp855
Cp856	ibm856 ibm-856, cp856 856
Cp857	ibm857, ibm-857, cp857, 857, csibm857
Cp860	ibm860, ibm-860, cp860, 860, csibm860
Cp861	ibm861, ibm-861, cp861, cp-is, 861, csibm861
Cp862	ibm862, ibm-862, cp862, cp862, 862, cspc862latinhebrew
Cp863	ibm863, ibm-863, cp863, 863, csibm863

Cp864	ibm864, ibm-864, cp864, csibm864
Cp865	ibm865, ibm-865, cp865, 865, csibm865
Cp866	ibm866, ibm-866, cp866, 866, csibm866
Cp868	ibm868, ibm-868, cp868, 868
Cp869	ibm869, ibm-869, cp869, 869, cp-gr, csibm869
Cp870	ibm870, ibm-870, cp870, 870
Cp871	ibm871, ibm-871, cp871, 871
Cp874	ibm874, ibm-874, cp874, 874
Cp875	ibm875, ibm-875, cp875, 875
Cp918	ibm918, ibm-918, cp918, 918
Cp921	ibm921, ibm-921, cp921, 921
Cp922	ibm922, ibm-922, cp922, 922
Cp930	ibm930, ibm-930, cp930, 930
Cp933	ibm933, ibm-933, cp933, 933
Cp935	ibm935, ibm-935, cp935, 935
Cp937	ibm937, ibm-937, cp937, 937
Cp939	ibm939, ibm-939, cp939, 939
Cp942	ibm942, ibm-942, cp942, 942
Cp942C	cp942c
Cp943	ibm943, ibm-943, cp943, 943
Cp943C	cp943c
Cp948	ibm948, ibm-948, cp948, 948
Cp949	ibm949, ibm-949, cp949, 949
Cp949C	cp949c
Cp950	ibm950, ibm-950, cp950, 950
Cp964	ibm964, ibm-964, cp964, 964

Cp970	ibm970, ibm-970, cp970, 970
EUC_CN	gb2312, gb2312-80, gb2312-1980, euc-cn, euccn
EUC_JP	eucjis, euc-jp, eucjp, extended_unix_code_packed_format_for_japanese, cseucpkdfmtjapanese, x-euc-jp, x-eucjp
EUC_KR	ksc5601, euc-kr, euckr, ks_c_5601-1987, ksc5601-1987, ksc5601_1987, ksc_5601, 5601
EUC_TW	cns11643, euc-tw, euctw
ISO2022JP	jis, iso-2022-jp, csiso2022jp, jis_encoding, csjisencoding
ISO8859_1	8859_1, iso_8859-1:1978, iso-ir-100, iso_8859-1, iso-8859-1, iso8859-1, latin1, l1, ibm819, ibm-819, cp819, 819, csisolatin1
ISO8859_15_FDIS	8859_15, iso-8859-15, iso_8859-15, iso8859-15, ibm923, ibm-923, cp923, 923, latin0, latin9, csisolatin0, csisolatin9
ISO8859_2	8859_2, iso_8859-2:1987, iso-ir-101, iso_8859-2, iso-8859-2, iso8859-2, latin2, l2, ibm912, ibm-912, cp912, 912, csisolatin2
ISO8859_3	8859_3, iso_8859-3:1988, iso-ir-109, iso_8859-3, iso-8859-3, iso8859-3, latin3, l3, ibm913, ibm-913, cp913, 913, csisolatin3
ISO8859_4	8859_4, iso_8859-4:1988, iso-ir-110, iso_8859-4, iso-8859-4, iso8859-4, latin4, l4, ibm914, ibm-914, cp914, 914, csisolatin4
ISO8859_5	8859_5, iso_8859-5:1988, iso-ir-144, iso_8859-5, iso-8859-5, iso8859-5, cyrillic, csisolatincyrillic, ibm915, ibm-915, cp915, 915
ISO8859_6	8859_6, iso_8859-6:1987, iso-ir-127, iso_8859-6, iso-8859-6, iso8859-6, ecma-114, asmo-708, arabic, csisolatarabic, ibm1089, ibm-1089, cp1089, 1089
ISO8859_7	8859_7, iso_8859-7:1987, iso-ir-126, iso_8859-7, iso-8859-7, iso8859-7, elot_928, ecma-118, greek, greek8, csisolatingreek, ibm813, ibm-813, cp813, 813
ISO8859_8	8859_8, iso_8859-8:1988, iso-ir-138, iso_8859-8, iso-8859-8, iso8859-8, hebrew, csisolatinhebrew, ibm916, ibm-916, cp916, 916

ISO8859_9	8859_9, iso-ir-148, iso_8859-9, iso-8859-9, iso8859-9, latin5, l5, ibm920, ibm-920, cp920, 920, csisolatin5
ISO8859_13	8859_13, iso_8859-13, iso-8859-13, iso8859-13,
JISAutoDetect	jis auto detect
Johab	ksc5601-1992, ksc5601_1992, ms1361
KOI8_R	koi8-r, koi8, cskoi8r
MS874	windows-874
MS932	shift_jis, ms_kanji, csshiftjis, windows-31j, cswindows31j, x-sjis
MS949	windows-949
SJIS	shift_jis, pck
TIS620	tis620.2533
UTF8	utf-8, unicode-1-1-utf-8
UTF16	utf-16
Unicode	unicode
UnicodeBig	unicode-1-1, iso-10646-ucs-2, utf-16be, x-utf-16be
UnicodeLittle	utf-16le, x-utf-16le