Tibero

TDP.NET 안내서

Tibero 7

Copyright © 2022 TmaxTibero Co., Ltd. All Rights Reserved.

Copyright Notice

Copyright © 2022 TmaxTibero Co., Ltd. All Rights Reserved.

대한민국 경기도 성남시 분당구 황새울로258번길 29, BS 타워 9층 우)13595

Website

http://www.tmaxtibero.com

기술서비스센터

Tel : +82-1544-8629

E-Mail : info@tmax.co.kr

Restricted Rights Legend

All TmaxTibero Software (Tibero®) and documents are protected by copyright laws and international convention. TmaxTibero software and documents are made available under the terms of the TmaxTibero License Agreement and this document may only be distributed or copied in accordance with the terms of this agreement. No part of this document may be transmitted, copied, deployed, or reproduced in any form or by any means, electronic, mechanical, or optical, without the prior written consent of TmaxTibero Co., Ltd. Nothing in this software document and agreement constitutes a transfer of intellectual property rights regardless of whether or not such rights are registered) or any rights to TmaxTibero trademarks, logos, or any other brand features.

This document is for information purposes only. The company assumes no direct or indirect responsibilities for the contents of this document, and does not guarantee that the information contained in this document satisfies certain legal or commercial conditions. The information contained in this document is subject to change without prior notice due to product upgrades or updates. The company assumes no liability for any errors in this document.

이 소프트웨어(Tibero®) 사용설명서의 내용과 프로그램은 저작권법과 국제 조약에 의해서 보호받고 있습니다. 사용설명서의 내용과 여기에 설명된 프로그램은 TmaxTibero Co., Ltd.와의 사용권 계약 하에서만 사용이 가능하며, 사용설명서는 사용권 계약의 범위 내에서만 배포 또는 복제할 수 있습니다. 이 사용설명서의 전부 또는 일부분을 TmaxTibero의 사전 서면 동의 없이 전자, 기계, 녹음 등의 수단을 사용하여 전송, 복제, 배포, 2차적 저작물작성 등의 행위를 하여서는 안 됩니다.

이 소프트웨어 사용설명서와 프로그램의 사용권 계약은 어떠한 경우에도 사용설명서 및 프로그램과 관련된 지적 재산권(등록 여부를 불문)을 양도하는 것으로 해석되지 아니하며, 브랜드나 로고, 상표 등을 사용할 권한을 부여하지 않습니다. 사용설명서는 오로지 정보의 제공만을 목적으로 하고, 이로 인한 계약상의 직접적 또는 간접적 책임을 지지 아니하며, 사용설명서 상의 내용은 법적 또는 상업적인 특정한 조건을 만족시키는 것을 보장하지는 않습니다. 사용설명서의 내용은 제품의 업그레이드나 수정에 따라 그 내용이 예고 없이 변경될 수 있으며, 내용상의 오류가 없음을 보장하지 아니합니다.

Trademarks

Tibero® is a registered trademark of TmaxTibero Co., Ltd. Other products, titles or services may be registered trademarks of their respective companies.

Tibero®는 TmaxTibero Co., Ltd.의 등록 상표입니다. 기타 모든 제품들과 회사 이름은 각각 해당 소유주의 상표로서 참조용으로만 사용됩니다.

Open Source Software Notice

Some modules or files of this product are subject to the terms of the following licenses. : OpenSSL, RSA Data Security, Inc., Apache Foundation, Jean-loup Gailly and Mark Adler, Paul Hsieh's hash

Detailed Information related to the license can be found in the following directory : ${INSTALL_PATH}/license/oss_licenses

본 제품의 일부 파일 또는 모듈은 다음의 라이선스를 준수합니다. : OpenSSL, RSA Data Security, Inc., Apache Foundation, Jean-loup Gailly and Mark Adler, Paul Hsieh's hash

관련 상세한 정보는 제품의 다음의 디렉터리에 기재된 사항을 참고해 주십시오. : ${INSTALL_PATH}/license/oss_licenses

안내서 정보

안내서 제목: Tibero TDP.NET 안내서

발행일: 2024-08-22

소프트웨어 버전: Tibero 7.2.2

안내서 버전: v7.2.2

내용 목차

안내서에 대하여 vii

제1장 Tibero TDP.NET 소개 1

1.1. Tibero TDP.NET 1

1.1.1. 동작 구조 1

1.1.2. TDP.NET 어셈블리 2

1.1.3. ODBC, OLEDB와 차이점 2

1.1.4. 시스템 요구 사항 2

1.1.5. 기본 경로 및 설정 3

제2장 TDP.NET 기본 사용 5

2.1. 개발 과정 5

2.2. Connection Properties 7

2.3. 데이터 타입 8

2.4. 기본 예제 9

2.5. psm 호출 10

제3장 API 구현 목록 13

3.1. 기본 클래스 13

3.1.1. TiberoClientFactory 13

3.1.2. TiberoCommand 13

3.1.3. TiberoCommandBuilder 14

3.1.4. TiberoConnection 15

3.1.5. TiberoConnectionStringBuilder 15

3.1.6. TiberoDataAdapter 16

3.1.7. TiberoDataReader 16

3.1.8. TiberoException 18

3.1.9. TiberoParameter 18

3.1.10. TiberoParameterCollection 19

3.1.11. TiberoTransaction 19

3.2. 타입 클래스 20

3.2.1. TiberoBlob 20

3.2.2. TiberoClob 21

3.2.3. TiberoRefCursor 22

3.2.4. TiberoBinary 22

3.2.5. TiberoBoolean 23

3.2.6. TiberoDate 24

3.2.7. TiberoDecimal 25

3.2.8. TiberoIntervalDS 27

3.2.9. TiberoIntervalYM 28

3.2.10. TiberoString 28

3.2.11. TiberoTimeStamp 29

3.3. UDT 29
3.3.1. TiberoCustomTypeMappingAttribute 29
3.3.2. TiberoObjectMappingAttribute 30
3.3.3. ITiberoCustomType 30
3.3.4. ITiberoCustomTypeFactory 30
3.3.5. ITiberoArrayTypeFactory 30
3.3.6. TiberoUdt 31
3.4. Bulk Copy 31
3.4.1. TiberoBulkCopy 31
3.4.2. TiberoBulkCopyColumnMapping 32
3.4.3. TiberoBulkCopyColumnMappingCollection 32
3.4.4. TiberoRowsCopiedEventArgs 33

안내서에 대하여

안내서의 대상

본 안내서는 Tibero®(이하 Tibero)에서 제공하는 Tibero Data Provider for .NET 기능을 이용하여 프로그램을 개발하려는 애플리케이션 개발자를 대상으로 기술한다.

안내서의 전제 조건

본 안내서를 원활히 이해하기 위해서는 다음과 같은 사항을 미리 알고 있어야 한다.

- 데이터베이스의 이해
- RDBMS의 이해
- .NET 프레임워크의 이해

안내서의 제한 조건

본 안내서는 TDP.NET 애플리케이션 프로그램을 실무에 적용하거나 운용하는 데 필요한 모든 사항을 포함하고 있지 않다. 따라서 TPD.NET을 활용한 운용과 관리 또는 고급 프로그래밍에 대해서는 각 제품 안내서를 참고하기 바란다.

안내서 구성

Tibero TDP.NET 안내서는 총 3개의 장으로 구성되어 있다.

각 장의 주요 내용은 다음과 같다.

- 제1장: Tibero TDP.NET 소개

  TDP.NET의 기본 개념과 동작 구조 및 제약사항을 기술한다.

- 제2장: TDP.NET 기본 사용

  TDP.NET을 활용한 Tibero 연동과 개발 과정 예시를 기술한다.

- 제3장: 구현 API 목록

  TDP.NET에서 제공하는 API 목록을 기술한다.

안내서 규약

표기	의미
<<AaBbCc123>>	프로그램 소스 코드의 파일명
<Ctrl>+C	Ctrl과 C를 동시에 누름
[Button]	GUI의 버튼 또는 메뉴 이름
진하게	강조
" "(따옴표)	다른 관련 안내서 또는 안내서 내의 다른 장 및 절 언급
'입력항목'	화면 UI에서 입력 항목에 대한 설명
하이퍼링크	메일 계정, 웹 사이트
>	메뉴의 진행 순서
+----	하위 디렉터리 또는 파일 있음
|----	하위 디렉터리 또는 파일 없음
참고	참고 또는 주의사항
주의	주의할 사항
[그림 1.1]	그림 이름
[예 1.1]	예제 이름
`AaBbCc123`	Java 코드, XML 문서
[*command argument*]	옵션 파라미터
< xyz >	'<'와 '>' 사이의 내용이 실제 값으로 변경됨
|	선택 사항. 예) A|B: A나 B 중 하나
…	파라미터 등이 반복되어서 나옴
${ }	환경변수

시스템 사용 환경

	요구 사항
Platform	Windows(x86) 32bit/64bit
Hardware	최소 2.5GB 하드디스크 공간
	1GB 이상 메모리 공간
Compiler	PSM (C99 지원 필요)
	tbESQL/C (C99 지원 필요)

관련 안내서

안내서	설명
Tibero 설치 안내서	설치 시 필요한 시스템 요구사항과 설치 및 제거 방법을 기술한 안내서이다.
Tibero tbCLI 안내서	Call Level Interface인 tbCLI의 개념과 구성요소, 프로그램 구조를 소개하고 tbCLI 프로그램을 작성하는 데 필요한 데이터 타입, 함수, 에러 메시지를 기술한 안내서이다.
Tibero 애플리케이션 개발자 안내서	각종 애플리케이션 라이브러리를 이용하여 애플리케이션 프로그램을 개발하는 방법을 기술한 안내서이다.
Tibero External Procedure 안내서	External Procedure를 소개하고 이를 생성하고 사용하는 방법을 기술한 안내서이다.
Tibero JDBC 개발자 안내서	Tibero에서 제공하는 JDBC 기능을 이용하여 애플리케이션 프로그램을 개발하는 방법을 기술한 안내서이다.
Tibero tbESQL/C 안내서	C 프로그래밍 언어를 사용해 데이터베이스 작업을 수행하는 각종 애플리케이션 프로그램을 작성하는 방법을 기술한 안내서이다.
Tibero tbESQL/COBOL 안내서	COBOL 프로그래밍 언어를 사용해 데이터베이스 작업을 수행하는 각종 애플리케이션 프로그램을 작성하는 방법을 기술한 안내서이다.
Tibero tbPSM 안내서	저장 프러시저 모듈인 tbPSM의 개념과 문법, 구성요소를 소개하고, 프로그램을 작성하는 데 필요한 제어 구조, 복합 타입, 서브 프로그램, 패키지와 SQL 문장을 실행하고 에러를 처리하는 방법을 기술한 안내서이다.
Tibero tbPSM 참조 안내서	저장 프러시저 모듈인 tbPSM의 패키지를 소개하고, 이러한 패키지에 포함된 각 프러시저와 함수의 프로토타입, 파라미터, 예제 등을 기술한 참조 안내서이다.
Tibero 관리자 안내서	Tibero의 동작과 주요 기능의 원활한 수행을 보장하기 위해 DBA가 알아야 할 관리 방법을 논리적 또는 물리적 측면에서 설명하고, 관리를 지원하는 각종 도구를 기술한 안내서이다.
Tibero 유틸리티 안내서	데이터베이스와 관련된 작업을 수행하기 위해 필요한 유틸리티의 설치 및 환경설정, 사용 방법을 기술한 안내서이다.
Tibero	Tibero Active Storage(TAS)를 사용해서 Tibero의 파일을 관리하고자 하는 관리자를 대상으로 기술한 안내서이다.

안내서	설명
TAS 안내서	
Tibero 에러 참조 안내서	Tibero를 사용하는 도중에 발생할 수 있는 각종 에러의 원인과 해결 방법을 기술한 안내서이다.
Tibero 참조 안내서	Tibero의 동작과 사용에 필요한 초기화 파라미터와 데이터 사전, 정적 뷰, 동적 뷰를 기술한 참조 안내서이다.
Tibero SQL 참조 안내서	데이터베이스 작업을 수행하거나 애플리케이션 프로그램을 작성할 때 필요한 SQL 문장을 기술한 참조 안내서이다.
Tibero Spatial 참조 안내서	Tibero에서 Geometry 타입에 대한 설명과 Spatial 기능 관련 프러시저 함수 목록 및 사용 방법 등을 기술한 안내서이다.
Tibero TEXT 참조 안내서	Tibero의 제공하는 Text Index를 소개하고, Text Index를 생성 하고 사용하는 방법을 기술하는 안내서이다.
Tibero IMCS 안내서	Tibero에서 제공하는 In-Memory Column Store(이하 IMCS) 기능을 기술하는 안내서이다.

제1장 Tibero TDP.NET 소개

본 장에서는 Tibero에서 제공하는 Tibero Data Provider for .NET (TDP.NET)를 소개한다.

1.1. Tibero TDP.NET

Tibero Data Provider for .NET (TDP.NET)은 Tibero가 제공하는 .NET Data Provider로서 사용자의 .NET 애플리케이션 프로그램과 Tibero간의 SQL 인터페이스 역할을 수행한다. TDP.NET은 ADO.NET의 핵심 클래스들을 구현함은 물론 별도의 API를 추가로 제공하고 있다.

TDP.NET은 다음과 같은 특징이 있다.

- TDP.NET은 ADO.NET의 핵심 개체들을 상속받아 구현하였다.
- 소스 코드 전체가 C#언어로 작성된 Managed Driver이다.
- 사용자는 C#, Visual Basic .NET과 같은 .NET 언어로 TDP.NET 프로그램을 작성할 수 있다.
- .NET 프레임워크 2.0 버전을 기반으로 작성되었다.
- Any CPU 모드로 빌드되었기 때문에 Windows 32-bit, 64-bit에서 모두 실행 가능하다.

1.1.1. 동작 구조

Tibero Data Provider for .NET(TDP.NET)은 Tibero 데이터베이스를 사용하기 위한 .NET Data Provider의 구현체이다. .NET Data Provider는 데이터를 조작하기 위한 ADO.NET의 구성요소로서 다음 4가지의 핵심 개체로 구성되어 있다.

개체	설명
Connection	데이터소스(data source)에 연결한다. TDP.NET의 구현체는 TiberoConnection 클래스이다.
Command	데이터소스에 대해 명령을 실행한다. TDP.NET의 구현체는 TiberoCommand 클래스이다.
DataReader	데이터소스에서 데이터를 조회한다. TDP.NET의 구현체는 TiberoDataReader 클래스이다.

개체	설명
DataAdapter	데이터소스에서 조회해온 내용을 DataSet에 채우거나, 업데이트된 내용을 데이터소스에 적용한다. TDP.NET의 구현체는 TiberoDataAdapter 클래스이다.

TDP.NET은 System.Data.Common 네임스페이스(namespace)로 정의되는 ADO.NET 클래스들을 상속하며, 추가로 Tibero 특정 기능 및 타입들을 제공하기 위한 클래스들을 포함하고 있다.

TDP.NET은 소스 코드 전체가 C# 언어로 작성된 Managed Driver로서, Tibero 데이터베이스 서버를 설치하지 않아도 .NET언어로 작성된 애플리케이션 프로그램을 개발할 수 있는 API를 제공한다.

1.1.2. TDP.NET 어셈블리

Tibero.DataAccess.dll 어셈블리는 두 개의 네임스페이스를 제공한다.

- Tibero.DataAccess.Client

  ADO.NET의 구현체와 TDP.NET에서 추가로 제공하는 클래스들을 포함한다.

- Tibero.DataAccess.Types

  Tibero에서 제공하는 타입을 지원하기 위한 TDP.NET 타입 클래스들을 포함한다.

1.1.3. ODBC, OLEDB와 차이점

.NET 환경에서 Tibero 데이터베이스에 접속하는 방법으로 ODBC, OLEDB, TDP.NET 3가지가 있다.

ODBC 또는 OLEDB를 사용할 경우 ODBC 드라이버(또는 MSDASQL 등)와 같은 별도의 계층을 거쳐 Tibero 데이터베이스에 접속하게 된다. 반면에 TDP.NET을 사용하게 되면, .NET언어로 작성한 문장을 Tibero 데이터베이스 고유의 프로토콜로 변환하므로 별도의 계층이 필요하지 않다.

또한 TDP.NET은 LOB, Ref Cursor와 같은 Tibero가 추가로 제공하는 타입이나 기능을 사용할 수 있도록 지원한다. 즉, TDP.NET을 사용하면 .NET 환경에서 가장 효율적으로 Tibero 데이터베이스에 접근이 가능하다.

1.1.4. 시스템 요구 사항

TDP.NET을 사용하기 위한 시스템 사양은 다음과 같다.

- 운영체제

  TDP.NET은 Windows 운영체제 32-bit, 64-bit에서 사용 가능하다. TDP.NET 어셈블리는 Any CPU 모드로 빌드되었기 때문에 32-bit, 64-bit 환경에서 모두 실행 가능하다.

- Microsoft .NET 프레임워크

  TDP.NET은 .NET 프레임워크 2.0 이상의 버전을 지원한다. 사용자의 .NET 프레임워크 환경이 4.0 이상일 경우 .NET 프레임워크 3.5 버전이 설치되어 있어야 한다.

1.1.5. 기본 경로 및 설정

TDP.NET 어셈블리가 위치한 기본 경로, 로그 설정 방법은 다음과 같다.

- 기본 경로

  Tibero를 서버에 설치한 후 TDP.NET 어셈블리는 $TB_HOME/client/win32/lib, $TB_HOME/client/win64/lib 폴더에 생성된다.

참고

Windows 이외의 플랫폼은 해당 경로 제외된다.

- 로그 설정

  Windows 환경변수로 다음 두 가지를 설정하여 로그 파일을 남길 수 있다.

환경변수	설명
TDPNET_LOG_PATH	로그 파일이 생성되는 경로를 지정한다.
TDPNET_LOG_LVL	로그 레벨을 지정한다. 현재 TDP.NET은 레벨로 TRACE 한 가지만 지원한다.

제2장 TDP.NET 기본 사용

본 장에서는 TDP.NET을 사용하여 애플리케이션 프로그램을 개발하는 과정을 순서대로 설명하고, 연결 속성, 데이터 타입, PSM 호출 방법 등을 설명한다.

2.1. 개발 과정

다음은 TDP.NET 을 사용하여 애플리케이션 프로그램을 개발하는 과정에 대한 설명이다.

1. **네임스페이스 추가**

   기본이 되는 Tibero.DataAccess.Client 네임스페이스를 추가한다.

```
using Tibero.DataAccess.Client;
```

2. **데이터베이스 연결**

   데이터베이스에 연결하기 위해서 접속 정보를 Connection string으로 직접 명시하여 TiberoConnection 객체를 생성한 뒤 Open() 메소드를 호출하여 데이터베이스에 접속한다.

```
string connStr = "Data Source=((INSTANCE="
                 + "(HOST=localhost)(PORT=8629)(DB_NAME=dbsvr)));"
                 + "User Id=tibero;Password=tmax;";
TbieroConnection conn = new TiberoConnection(connStr);
conn.Open();
```

   또는 다음과 같이 TiberoConnectionStringBuilder를 통해 Connection string을 생성하여 TiberoConnection 객체를 생성할 수 있다.

```
TiberoConnectionStringBuilder builder = new TiberoConnectionStringBuilder();
builder.DataSource = "((INSTANCE=(HOST=localhost)(PORT=8629)(DB_NAME=dbsvr)))";
builder.UserID = "tibero";
builder.Password = "tmax";

TbieroConnection conn = new TiberoConnection(builder.ToString());
conn.Open();
```

   Connection string의 Data Source에 Data Source Name을 명시하면, $TB_HOME/client/config/tbdsn.tbr 파일에 명시된 DSN의 HOST, PORT, DB_NAME을 파싱하여 접속 정보로 사용한다.

```
string connStr = "Data Source=tibero;User Id=tibero;Password=tmax;";
TbieroConnection conn = new TiberoConnection(connStr);
conn.Open();
```

3. **TiberoCommand 객체 생성과 SQL문 수행**

   데이터베이스에 연결되면, TiberoCommand 객체를 생성한 뒤 ExecuteXXX() 메소드를 통해 SQL 커맨드를 데이터베이스에 보낸다.

```
conn.Open();
TiberoCommand cmd = new TiberoCommand("create table t (c1 varchar2(10), c2 number)",

                                      conn);
cmd.ExecuteNonQuery();
```

4. **질의문 수행과 TiberoDataReader 객체 받기**

   커맨드 객체를 생성하여 질의문을 수행하고 TiberoDataReader 객체를 받는다.

```
TiberoCommand cmd = new TiberoCommand("select * from t");
cmd.Connection = conn;
cmd.CommandType = CommandType.Text;
TiberoDataReader reader = cmd.ExecuteReader();
```

5. **TiberoDataReader 객체 처리**

   TiberoDataReader 객체는 Read() 메소드를 이용하여 결과 집합의 각 로우에 순차적으로 접근할 수 있다. 더이상 읽을 로우가 없으면 Read() 메소드는 false를 반환한다. TiberoDataReader의 GetXXX() 메소드 호출을 통해 컬럼 값을 얻거나, 컬럼 배열에 접근하는 직관적인 방식으로 컬럼 값을 얻어 올 수 있다.

```
while (reader.Read())
{
   string c1 = reader.GetString(0);
   int c2 = reader["c2"];
}
```

6. **TiberoDataAdapter를 통한 질의문 수행**

   TiberoDataAapter를 통해서도 데이터베이스에 커맨드를 요청하고, Fill 메소드를 통해 DataSet 객체에 결과 집합을 담을 수 있다. 데이터베이스와의 연결이 닫혀있는 상태일지라도, Fill 메소드 내부에서 암시적으로 데이터베이스와 연결을 맺어 데이터를 조회해온 후 다시 연결을 해제하여 연결 상태를 닫힌 상태로 원복한다.

```
TiberoCommand cmd = new TiberoCommand("select * from t", conn);
DataSet ds = new DataSet();
TiberoDataAdapter da = new TiberoDataAdapter(cmd);

da.Fill(ds);
foreach (DataRow r in ds.Tables[0].Rows)
{
```

```
   Console.WriteLine(r[0] + "," + r[1]);
}
```

7. **데이터베이스 연결 해제**

   TiberoConnection 객체의 Close() 메소드를 호출하면 데이터베이스와의 연결이 해제된다.

```
conn.Close();
```

참고

Close() 메소드를 호출하지 않으면 불필요한 커넥션이 남아서 성능 저하를 유발시킬 수 있으므로, 항상 Close() 메소드를 사용하여 연결을 해제해야 한다.

---

8. **객체 소멸**

   명시적으로 Dispose() 메소드를 호출하면 TiberoDataReader, TiberoCommand, TiberoConnection 등의 객체가 사용한 리소스가 정리되며 소멸된다.

```
cmd.Dispose();
reader.Dispose();
conn.Dispose();
```

2.2. Connection Properties

다음은 데이터베이스에 연결할 때에 Connection String을 통해 설정할 수 있는 속성들이다.

속성명	타입	설명
DB_NAME	String	서버에 존재하는 특정 데이터베이스의 이름이다.
Data Source	String	데이터소스의 이름이다.
INSTANCE	String	접속할 인스턴스 정보다.
Password	String	서버 접속을 위한 패스워드이다.
User Id	String	서버 접속을 위한 사용자 이름이다.
PORT	String	서버 리스너의 포트 번호이다.
HOST	String	접속할 인스턴스의 호스트 및 IP 주소 정보다.
Enlist	String	트랙잭션 범위 내에서 Connection open할 때 암시적으로 System.Transactions의 트랜잭션에 참여할지 여부를 지정한다. (기본값: True)
Pooling	String	Connection Pooling 기능을 사용할지 여부를 지정한다.

속성명	타입	설명
		(기본값: True)
Min Pool Size	String	한 개의 Connection Pool 내에 존재해야 하는 커넥션의 최소 갯수이다. (기본값: 1)
Max Pool Size	String	한 개의 Connection Pool 내에 존재할 수 있는 커넥션의 최대 갯수이다. (기본값: 100)
Decr Pool Size	String	Connection Pool 내의 커넥션들의 단위 감소량이다. (기본값: 1)
Incr Pool Size	String	Connection Pool 내의 커넥션들의 단위 증가량이다. (기본값: 5)
Validate Connection	String	Connection Pool에서 커넥션을 가져올 때마다 데이터베이스 연결 유효성 검사를 할지 여부를 지정한다. True로 지정하면, 연결 유효성 검사에 따른 오버헤드가 생긴다는 것을 감안해야 한다. (기본값: False)
Connection Timeout	String	Connection Pool에서 커넥션을 가져올 때 Max Pool Size 만큼의 커넥션들이 모두 Busy 상태일 경우 커넥션을 획득하기 위해 기다리는 최대 시간이다. (단위: 초, 기본값: 15초)
Promotable Transaction	String	분산 트랜잭션으로의 승격 가능 여부를 지정한다. – Promotable (기본값) – Local

2.3. 데이터 타입

다음은 TDP.NET에서 제공하는 타입의 Value 속성과 .NET 프레임워크 데이터 타입의 대응 관계이다. Tibero.DataAccess.Types 네임스페이스를 추가하면 TDP.NET 타입을 사용할 수 있다.

Tibero DB 데이터 타입	TDP.NET 타입	.NET 프레임워크 데이터 타입
BFILE	TiberoBFile	System.Byte[]
BINARY_DOUBLE	TiberoDecimal	System.Decimal
BINARY_FLOAT	TiberoDecimal	System.Decimal
BINARY_INTEGER	TiberoDecimal	System.Decimal
BLOB	TiberoBlob	System.Byte[]
BOOLEAN	TiberoBoolean	System.Boolean
CHAR	TiberoString	System.String
CLOB	TiberoClob	System.String
DATE	TiberoDate	System.DateTime

Tibero DB 데이터 타입	TDP.NET 타입	.NET 프레임워크 데이터 타입
INTERVAL DAY TO SECOND	TiberoIntervalDS	System.TimeSpan
INTERVAL YEAR TO MONTH	TiberoIntervalYM	System.Int64
LONG	TiberoString	System.Byte[]
LONG RAW	TiberoBinary	System.Byte[]
NCHAR	TiberoString	System.String
NCLOB	TiberoClob	System.String
NUMBER	TiberoDecimal	System.Decimal
NVARCHAR2	TiberoString	System.String
REF	TiberoRef	System.String
REF CURSOR	TiberoRefCursor	-
ROWID	TiberoString	System.String
TIMESTAMP	TiberoTimeStamp	System.DateTime
TIMESTAMP WITH LOCAL TIME ZONE	TiberoTimeStampLTZ	System.DateTime
TIMESTAMP WITH TIME ZONE	TiberoTimeStampTZ	System.DateTime
VARCHAR2	TiberoString	System.String
XMLTYPE	TiberoXmlType	System.String

2.4. 기본 예제

다음은 파라미터를 바인딩하여 DML을 수행하는 예제에 대한 설명이다.

1. **BindbyName 설정 후 파라미터 바인딩**

   TiberoCommand 객체의 BindbyName 값을 True로 설정하는 경우 지정한 파라미터의 이름에 따라 바인딩된다.

```
TiberoCommand cmd = new TiberoCommand("create table test " +
                                      "(c1 varchar2(20), c2 varchar2(20))",
                                      conn);
cmd.ExecuteNonQuery();
cmd = new TiberoCommand("insert into test values (:col1, :col2)", conn);
cmd.BindByName = true;

TiberoParameter param = new TiberoParameter();
param.ParameterName = "col2";
param.TiberoDbType = TiberoDbType.Varchar2;
param.Direction = ParameterDirection.Input;
param.Size = 20;
```

```
param.CollectionType = TiberoCollectionType.None;
param.Value = "second column";
cmd.Parameters.Add(param);
cmd.Parameters.Add("col1", TiberoDbType.Varchar2).Value = "first column";
int affectedRowCnt = cmd.ExecuteNonQuery();
```

2. **일반적인 파라미터 바인딩**

TiberoCommand 객체에 BindbyName 설정을 별도로 하지 않으면 기본값은 False로 설정된다. 따라서 파라미터의 이름은 의미가 없고 파라미터를 추가하는 순서대로 바인딩 됨을 주의해야 한다.

```
TiberoCommand cmd = new TiberoCommand("create table test2 " +
                                      "(c1 date, c2 number, c3 number)", conn);
cmd.ExecuteNonQuery();
cmd = new TiberoCommand("insert into test2 values (:col1, :col2, :colNull)", conn);

DateTime dt = new DateTime(2020, 3, 23, 17, 50, 30);
cmd.Parameters.Add("col1", TiberoDbType.Date, dt, ParameterDirection.Input);
cmd.Parameters.Add("col2", TiberoDbType.Decimal, 123.45m, ParameterDirection.Input);
cmd.Parameters.Add("col3", null);
int affectedRowCnt = cmd.ExecuteNonQuery();
```

2.5. psm 호출

다음은 ref 커서를 아웃 파라미터로 바인딩하여 Stored Procedure를 수행하여 테이블의 데이터를 조회해 오는 예제에 대한 설명이다.

```
TiberoCommand cmd = new TiberoCommand("create table test3 " +
                                      "(c1 varchar(50), c2 number)", conn);
cmd.ExecuteNonQuery();
cmd = new TiberoCommand("insert into test3 values ('hello', 10)", conn);
cmd.ExecuteNonQuery();
cmd = new TiberoCommand("create or replace procedure " +
                        "out_cursor(p1 out sys_refcursor)" +
                        "as " +
                        "begin " +
                        "open p1 for 'select * from test3'; " +
                        "end; ", conn);
cmd.ExecuteNonQuery();
cmd = new TiberoCommand();
cmd.Connection = conn;
cmd.CommandText = "out_cursor";
cmd.CommandType = CommandType.StoredProcedure;
cmd.Parameters.Add("csr",TiberoDbType.RefCursor, ParameterDirection.Output);
```

```
TiberoDataAdapter oAdapter = new TiberoDataAdapter(cmd);
DataSet ds = new DataSet();
oAdapter.Fill(ds);
foreach (DataRow r in ds.Tables[0].Rows)
{
   Console.WriteLine(r[0] + ", " + r[1]);
}
```

제3장 API 구현 목록

본 장에서는 TDP.NET에서 제공하는 API에 대해 설명한다.

3.1. 기본 클래스

다음은 TDP.NET 기본 클래스에 대한 구현 목록이다.

3.1.1. TiberoClientFactory

- 메소드

  다음은 TiberoClientFactory에서 제공하는 메소드이다.

메소드	설명
CreateCommand	TiberoCommand 객체를 나타내는 DbCommand 객체를 반환한다.
CreateCommandBuilder	TiberoCommandBuilder를 나타내는 DbCommandBuilder 객체를 반환한다.
CreateConnection	TiberoConnection 객체를 나타내는 DbConnection 객체를 반환한다.
CreateConnectionStringBuilder	TiberoConnectionStringBuilder 객체를 나타내는 DbConnectionStringBuilder 객체를 반환한다.
CreateDataAdapter	TiberoDataAdapter 객체를 나타내는 DbDataAdapter 객체를 반환한다.
CreateParameter	TiberoParameter 객체를 나타내는 DbParameter 객체를 반환한다.

3.1.2. TiberoCommand

- 속성

  다음은 TiberoCommand에서 제공하는 속성이다.

속성	설명
ArrayBindCount	바인딩할 최대 배열 요소 수를 지정한다.
BindByName	collection에서 바인딩 방법을 지정한다.
CommandText	SQL문이나 저장 프러시저를 지정한다.
CommandTimeout	초 단위의 타임아웃을 지정한다.
CommandType	CommandText의 해석 방법을 지정한다.

속성	설명
Connection	TiberoConnection 객체를 지정한다.
Parameters	SQL문이나 저장 프러시저의 파라미터들을 지정한다.
Transaction	TiberoTransaction 객체를 지정한다.

- 메소드

  다음은 TiberoCommand에서 제공하는 메소드이다.

메소드	설명
Cancel	현재 실행 중인 명령을 취소한다.
CreateParameter	TiberoParameter 클래스의 새 인스턴스를 만든다.
ExecuteNonQuery	SQL문을 실행한다.
ExecuteReader	Command를 실행하고 TiberoDataReader 객체를 반환한다.
ExecuteScalar	Command를 실행하고 첫 번째 결과를 반환한다.

3.1.3. TiberoCommandBuilder

- 속성

  다음은 TiberoCommandBuilder에서 제공하는 속성이다.

속성	설명
DataAdapter	SQL문이 생성되는 TiberoDataAdapter를 나타낸다.

- 메소드

  다음은 TiberoCommandBuilder에서 제공하는 메소드이다.

메소드	설명
GetDeleteCommand	Delete를 수행하는 SQL구문을 가진 자동 생성된 TiberoCommand 객체를 가져온다.
GetInsertCommand	Insert를 수행하는 SQL구문을 가진 자동 생성된 TiberoCommand 객체를 가져온다.
GetUpdateCommand	Update를 수행하는 SQL구문을 가진 자동 생성된 TiberoCommand 객체를 가져온다.

3.1.4. TiberoConnection

- 속성

  다음은 TiberoConnection에서 제공하는 속성이다.

속성	설명
ClientId	연결을 위한 클라이언트 식별자를 지정한다.
ConnectionString	연결에 사용되는 연결 정보를 지정한다.
ConnectionTimeout	요청이 종료되기 전에 Open 메소드가 pooled connection을 얻을 수 있는 최대 시간을 나타낸다.
DataSource	데이터소스의 이름을 나타낸다.
ServerVersion	Tibero 서버 버전을 나타낸다.
DatabaseName	데이터베이스 이름을 나타낸다.
HostName	클라이언트의 호스트 이름을 나타낸다.
InstanceName	인스턴스 이름을 나타낸다.
ServiceName	서비스 이름을 나타낸다.
State	현재 커넥션의 상태를 나타낸다.

- 메소드

  다음은 TiberoConnection에서 제공하는 메소드이다.

메소드	설명
BeginTransaction	로컬 트랜잭션을 시작한다.
Close	데이터베이스 연결을 종료한다.
CreateCommand	TiberoConnection 객체와 연관된 TiberoCommand 객체를 만들고 반환한다.
Open	ConnectionString에 지정된 설정으로 데이터베이스 연결을 시작한다.

3.1.5. TiberoConnectionStringBuilder

- 속성

  다음은 TiberoConnectionStringBuilder에서 제공하는 속성이다.

속성	설명
ConnectionTimeout	ConnectionString에서 Connection Timeout에 해당하는 값을 지정한다.
DataSource	ConnectionString에서 Data Source에 해당하는 값을 지정한다.

속성	설명
DecrPoolSize	ConnectionString에서 Decr Pool Size에 해당하는 값을 지정한다.
Enlist	ConnectionString에서 Enlist에 해당하는 값을 지정한다.
IncrPoolSize	ConnectionString에서 Incr Pool Size에 해당하는 값을 지정한다.
MaxPoolSize	ConnectionString에서 Max Pool Size에 해당하는 값을 지정한다.
MinPoolSize	ConnectionString에서 Min Pool Size에 해당하는 값을 지정한다.
Password	ConnectionString에서 Password에 해당하는 값을 지정한다.
Pooling	ConnectionString에서 Pooling에 해당하는 값을 지정한다.
UserId	ConnectionString에서 User Id에 해당하는 값을 지정한다.
ValidateConnection	ConnectionString에서 Validate Connection에 해당하는 값을 지정한다.

3.1.6. TiberoDataAdapter

- 속성

  다음은 TiberoDataAdapter에서 제공하는 속성이다.

속성	설명
DeleteCommand	Delete를 위한 SQL문 또는 저장 프러시저이다.
InsertCommand	Insert를 위한 SQL문 또는 저장 프러시저이다.
SelectCommand	Select를 위한 SQL문 또는 저장 프러시저이다.
UpdateCommand	Update를 위한 SQL문 또는 저장 프러시저이다.

- 메소드

  다음은 TiberoDataAdapter에서 제공하는 메소드이다.

메소드	설명
Fill	데이터베이스의 데이터와 일치하도록 DataSet의 행을 추가하거나 고친다.
Update	업데이트를 수행한다.

3.1.7. TiberoDataReader

- 속성

  다음은 TiberoDataReader에서 제공하는 속성이다.

속성	설명
FieldCount	결과 집합의 열 수를 가져온다.
HasRows	TiberoDataReader에 하나 이상의 행이 있는지 나타낸다.
HiddenFieldCount	TiberoDataReader에서 숨겨진 필드 수를 가져온다.
RecordsAffected	SQL문 수행으로 수정되거나 삽입 또는 삭제된 행의 수를 가져온다.
VisibleFieldCount	TiberoDataReader에서 숨겨지지 않은 필드 수를 가져온다.

- 메소드

  다음은 TiberoDataReader에서 제공하는 메소드이다.

메소드	설명
Close	TiberoDataReader를 닫는다.
GetBytes	제공된 byte 배열을 지정된 열에서 최대 바이트 수까지 채운다.
GetDataTypeName	지정된 열의 타입 이름을 반환한다.
GetDateTime	지정된 열의 DateTime 값을 반환한다.
GetDecimal	지정된 열의 decimal 값을 반환한다.
GetDouble	지정된 열의 double 값을 반환한다.
GetFieldType	지정된 열의 타입을 반환한다.
GetFloat	지정된 열의 float 값을 반환한다.
GetInt16	지정된 열의 Int16 값을 반환한다.
GetInt32	지정된 열의 Int32 값을 반환한다.
GetInt64	지정된 열의 Int64 값을 반환한다.
GetName	지정된 열의 이름을 반환한다.
GetOrdinal	지정된 열 이름의 0부터 시작하는 서수를 반환한다.
GetString	지정된 열의 string 값을 반환한다.
GetTiberoBinary	지정된 열의 TiberoBinary를 반환한다.
GetTiberoBlob	지정된 열의 TiberoBlob을 반환한다.
GetTiberoClob	지정된 열의 TiberoClob을 반환한다.
GetTiberoDate	지정된 열의 TiberoDate를 반환한다.
GetTiberoDecimal	지정된 열의 TiberoDecimal을 반환한다.
GetTiberoIntervalDS	지정된 열의 TiberoIntervalDS를 반환한다.
GetTiberoIntervalYM	지정된 열의 TiberoIntervalYM을 반환한다.
GetTiberoString	지정된 열의 TiberoString을 반환한다.
GetTiberoTimestamp	지정된 열의 TiberoTimestamp를 반환한다.

메소드	설명
GetTimeSpan	지정된 열의 TimeSpan 값을 반환한다.
GetValue	열 값을 .NET type으로 반환한다.
GetValues	모든 열 값을 .NET type으로 반환한다.
IsDBNull	열 값이 null인지 여부를 나타낸다.
NextResult	다음 결과 집합을 읽는다.
Read	결과 집합에서 다음 행을 읽는다.

3.1.8. TiberoException

- 메소드

  다음은 TiberoException에서 제공하는 메소드이다.

메소드	설명
ToString	Exception을 나타내는 String을 반환한다.

3.1.9. TiberoParameter

- 속성

  다음은 TiberoParameter에서 제공하는 속성이다.

속성	설명
ArrayBindSize	Array 바인딩 전 혹은 후에 매개변수의 Value 속성에서 입력 또는 출력 크기를 지정한다.
ArrayBindStatus	Array 바인딩 전 혹은 후에 매개변수의 Value 속성에서 입력 또는 출력 상태를 지정한다.
CollectionType	TiberoParameter의 collection type을 지정한다.
DbType	Data.DbType 열거 형을 사용하여 매개 변수의 데이터 유형을 지정한다.
Direction	매개변수가 입력 전용인지, 출력 전용인지, 양방향인지를 지정한다.
ParameterName	매개변수의 이름을 지정한다.
Precision	Value 속성을 나타내기 위한 최대 자릿수를 지정한다.
Scale	Value 속성을 나타내기 위한 소수 자릿수를 지정한다.
SourceColumn	DataSet의 DataTable 열의 이름을 지정한다.
TiberoDbType	Tibero 데이터 타입을 지정한다.

속성	설명
Value	Parameter의 값을 지정한다.

3.1.10. TiberoParameterCollection

- 속성

  다음은 TiberoParameterCollection에서 제공하는 속성이다.

속성	설명
Count	TiberoParameter 수를 지정한다.
Item	TiberoParameter 객체를 가져오고 설정한다.

- 메소드

  다음은 TiberoParameterCollection에서 제공하는 메소드이다.

메소드	설명
Add	객체를 추가한다.
AddRange	TiberoParameterCollection의 끝에 요소를 추가한다.
Clear	모든 TiberoParameter 객체를 제거한다.
Contains	객체의 존재여부를 나타낸다.
CopyTo	제공된 인덱스부터 시작하여 제공된 배열로 TiberoParameter 객체들을 복사한다.
IndexOf	객체의 인덱스를 반환한다.
Insert	지정된 인덱스에 제공된 TiberoParameter를 삽입한다.
Remove	Collection에서 객체를 제거한다.
RemoveAt	지정된 인덱스나 이름을 통해 객체를 제거한다.

3.1.11. TiberoTransaction

- 속성

  다음은 TiberoTransaction에서 제공하는 속성이다.

속성	설명
IsolationLevel	트랜잭션의 isolation level을 지정한다.
Connection	트랜잭션과 연관된 커넥션을 지정한다.

- 메소드

  다음은 TiberoTransaction에서 제공하는 메소드이다.

메소드	설명
Commit	트랜잭션을 커밋한다.
Rollback	트랜잭션을 롤백한다.

3.2. 타입 클래스

다음은 TDP.NET 타입 클래스에 대한 구현 목록이다.

3.2.1. TiberoBlob

- 속성

  다음은 TiberoBlob에서 제공하는 속성이다.

속성	설명
CanRead	LOB 스트림을 읽을 수 있는지 여부를 나타낸다.
CanSeek	정방향 및 역방향 탐색 작업 수행 여부를 나타낸다.
CanWrite	LOB 객체가 쓰기를 지원하는지 여부를 나타낸다.
Connection	BLOB 데이터를 검색하고 쓰는데 사용되는 TiberoConnection을 나타낸다.
IsEmpty	BLOB이 비어있는지 여부를 나타낸다.
IsNull	현재 인스턴스에 null 값이 있는지 여부를 나타낸다.
IsTemporary	현재 인스턴스가 temporary BLOB인지 나타낸다.
Length	BLOB 데이터의 크기를 나타낸다.
Position	LOB 스트림에서 현재 읽거나 쓰는 위치를 나타낸다.
Value	byte 배열로 데이터를 반환한다.

- 메소드

  다음은 TiberoBlob에서 제공하는 메소드이다.

메소드	설명
Append	제공된 데이터를 현재 TiberoBlob 인스턴스에 추가한다.
Clone	TibeoBlob 객체의 사본을 만든다.
Close	현재 스트림을 닫고 관련 자원을 해제한다.

메소드	설명
Compare	현재 인스턴스의 데이터를 제공된 객체와 비교한다.
CopyTo	데이터를 TiberoBlob으로 복사한다.
Dispose	객체가 할당한 자원을 해제한다.
Erase	데이터를 지운다.
IsEqual	두 TiberoBlob의 데이터를 비교한다.
Read	지정된 수만큼 바이트를 읽고 버퍼에 채운다.
Search	바이너리 패턴을 검색한다.
Seek	현재 LOB 스트림에서 위치를 설정한다.
SetLength	BLOB 값을 지정된 길이로 자른다.
Write	제공된 버퍼를 TiberoBlob에 쓴다.

3.2.2. TiberoClob

- 속성

  다음은 TiberoClob에서 제공하는 속성이다.

속성	설명
CanRead	LOB 스트림을 읽을 수 있는지 여부를 나타낸다.
CanSeek	정방향 및 역방향 탐색 작업 수행 여부를 나타낸다.
CanWrite	LOB 객체가 쓰기를 지원하는지 여부를 나타낸다.
Connection	CLOB 데이터를 검색하고 쓰는데 사용되는 TiberoConnection을 나타낸다.
IsEmpty	CLOB이 비어있는지 여부를 나타낸다.
IsNClob	TiberoClob 객체가 NCLOB인지 나타낸다.
IsNull	현재 인스턴스에 null 값이 있는지 여부를 나타낸다.
IsTemporary	현재 인스턴스가 temporary CLOB인지 나타낸다.
Length	CLOB 데이터의 크기를 나타낸다.
Position	LOB 스트림에서 현재 읽거나 쓰는 위치를 나타낸다.
Value	String으로 데이터를 반환한다.

- 메소드

  다음은 TiberoClob에서 제공하는 메소드이다.

메소드	설명
Append	제공된 데이터를 현재 TiberoClob 인스턴스에 추가한다.
Clone	TibeoClob 객체의 사본을 만든다.
Close	현재 스트림을 닫고 관련 자원을 해제한다.
Compare	현재 인스턴스의 데이터를 제공된 객체와 비교한다.
CopyTo	데이터를 TiberoClob으로 복사한다.
Dispose	객체가 할당한 자원을 해제한다.
Erase	지정된 양 만큼 데이터를 지운다.
IsEqual	두 TiberoClob의 데이터를 비교한다.
Read	현재 인스턴스로부터 읽는다.
Search	문자 패턴을 검색한다.
Seek	현재 LOB 스트림에서 위치를 설정한다.
SetLength	CLOB 값을 지정된 길이로 자른다.
Write	제공된 버퍼를 TiberoClob에 쓴다.

3.2.3. TiberoRefCursor

- 속성

  다음은 TiberoRefCursor에서 제공하는 속성이다.

속성	설명
Connection	REF CURSOR 데이터를 fetch하는 데 사용되는 TiberoConnection에 대한 참조이다.

- 메소드

  다음은 TiberoRefCursor에서 제공하는 메소드이다.

메소드	설명
GetDataReader	REF CURSOR에 대한 TiberoDataReader 객체를 반환한다.

3.2.4. TiberoBinary

- 속성

  다음은 TiberoBinary에서 제공하는 속성이다.

속성	설명
IsNull	현재 인스턴스가 null 값을 가졌는지 여부를 나타낸다.
Length	바이너리 데이터의 길이를 반환한다.
Value	바이너리 데이터를 반환한다.

- 정적 메소드

  다음은 TiberoBinary에서 제공하는 정적 메소드이다.

메소드	설명
Equals	두 TiberoBinary 값이 같은지 확인한다.
GreaterThan	첫 번째 TiberoBinary 값이 두 번째 값보다 큰지 확인한다.
GreaterThanOrEqual	첫 번째 TiberoBinary 값이 두 번째 값보다 크거나 같은지 확인한다.
LessThan	첫 번째 TiberoBinary 값이 두 번째 값보다 작은지 확인한다.
LessThanOrEqual	첫 번째 TiberoBinary 값이 두 번째 값보다 작거나 같은지 확인한다.
NotEquals	두 TiberoBinary 값이 같지 않은지 확인한다.

- 메소드

  다음은 TiberoBinary에서 제공하는 메소드이다.

메소드	설명
Equals	두 객체가 같은 바이너리 데이터를 갖는지 확인한다.

3.2.5. TiberoBoolean

- 속성

  다음은 TiberoBoolean에서 제공하는 속성이다.

속성	설명
Value	현재 인스턴스가 나타내는 boolean 값을 반환한다.

- 정적 메소드

  다음은 TiberoBoolean에서 제공하는 정적 메소드이다.

메소드	설명
And	두 TiberoBoolean 인스턴스의 비트 AND 연산 결과를 반환한다.

메소드	설명
Equals	두 TiberoBoolean 값이 같은지 확인한다.
GreaterThan	첫 번째 TiberoBoolean 값이 두 번째 값보다 큰지 확인한다.
GreaterThanOrEquals	첫 번째 TiberoBoolean 값이 두 번째 값보다 크거나 같은지 확인한다.
LessThan	첫 번째 TiberoBoolean 값이 두 번째 값보다 작은지 확인한다.
LessThanOrEquals	첫 번째 TiberoBoolean 값이 두 번째 값보다 작거나 같은지 확인한다.
NotEquals	두 TiberoBoolean 값이 같지 않은지 확인한다.
OnesComplement	지정된 TiberoBoolean 값의 1의 보수 연산 결과를 반환한다.
Or	두 TiberoBoolean 인스턴스의 비트 OR 연산 결과를 반환한다.
Parse	String을 사용해 값이 설정된 TiberoBoolean을 반환한다.
Xor	두 TiberoBoolean 인스턴스의 비트 XOR 연산 결과를 반환한다.

- 메소드

  다음은 TiberoBoolean에서 제공하는 메소드이다.

메소드	설명
CompareTo	현재 인스턴스를 제공된 객체와 비교하고 상대 값을 나타내는 정수를 반환한다.
Equals	객체가 TiberoBoolean의 인스턴스인지 여부와 객체의 값이 현재 인스턴스와 같은지 여부를 결정한다.
GetHashCode	현재 인스턴스의 해시 코드를 반환한다.
ToString	현재 인스턴스를 나타내는 String을 반환한다.

3.2.6. TiberoDate

- 속성

  다음은 TiberoDate에서 제공하는 속성이다.

속성	설명
Day	TibeoDate의 day 값을 가져온다.
Hour	TiberoDate의 hour 값을 가져온다.
Minute	TiberoDate의 minute 값을 가져온다.
Month	TiberoDate의 month 값을 가져온다.
Second	TiberoDate의 second 값을 가져온다.
Value	TiberoDate의 날짜와 시간을 가져온다.

속성	설명
Year	TiberoDate의 year 값을 가져온다.

- 정적 메소드

  다음은 TiberoDate에서 제공하는 정적 메소드이다.

메소드	설명
Equals	두 TiberoDate 값이 같은지 확인한다.
GreaterThan	첫 번째 TiberoDate 값이 두 번째 값보다 큰지 확인한다.
GreaterThanOrEqual	첫 번째 TiberoDate 값이 두 번째 값보다 크거나 같은지 확인한다.
LessThan	첫 번째 TiberoDate 값이 두 번째 값보다 작은지 확인한다.
LessThanOrEqual	첫 번째 TiberoDate 값이 두 번째 값보다 작거나 같은지 확인한다.
NotEquals	두 TiberoDate 값이 같지 않은지 확인한다.
Parse	String을 사용해 값이 설정된 TiberoDate를 반환한다.

- 메소드

  다음은 TiberoDate에서 제공하는 메소드이다.

메소드	설명
CompareTo	현재 인스턴스를 제공된 객체와 비교하고 상대 값을 나타내는 정수를 반환한다.
Equals	객체가 TiberoDate의 인스턴스인지 여부와 객체의 값이 현재 인스턴스와 같은지 여부를 결정한다.
ToString	현재 인스턴스를 나타내는 String을 반환한다.

3.2.7. TiberoDecimal

- 속성

  다음은 TiberoDecimal에서 제공하는 속성이다.

속성	설명
Value	decimal 값을 반환한다.

- 정적 메소드

  다음은 TiberoDecimal에서 제공하는 정적 메소드이다.

메소드	설명
Equals	두 TiberoDecimal 값이 같은지 확인한다.
GreaterThan	첫 번째 TiberoDecimal 값이 두 번째 값보다 큰지 확인한다.
GreaterThanOrEqual	첫 번째 TiberoDecimal 값이 두 번째 값보다 크거나 같은지 확인한다.
LessThan	첫 번째 TiberoDecimal 값이 두 번째 값보다 작은지 확인한다.
LessThanOrEqual	첫 번째 TiberoDecimal 값이 두 번째 값보다 작거나 같은지 확인한다.
NotEquals	두 TiberoDecimal 값이 같지 않은지 확인한다.
Abs	TiberoDecimal의 절대 값을 반환한다.
Add	두 TiberoDecimal을 더한다.
Ceiling	천장 함수값을 TiberoDecimal로 반환한다.
Divide	한 TiberoDecimal을 다른 것으로 나눈 값을 반환한다.
Floor	바닥 함수값을 TiberoDecimal로 반환한다.
Max	두 TiberoDecimal의 최대 값을 반환한다.
Min	두 TiberoDecimal의 최소 값을 반환한다.
Mod	두 TiberoDecimal의 Modulo 연산 값을 반환한다.
Multiply	두 TiberoDecimal의 곱을 반환한다.
Negate	제공된 TiberoDecimal의 부정 값을 반환한다.
Parse	String을 TiberoDecimal로 변환한다.
Round	지정된 위치에서 반올림한 값을 반환한다.
Sign	TiberoDecimal의 부호를 확인한다.
Sqrt	제공된 TiberoDecimal의 제곱근 값을 반환한다.
Subtract	한 TiberoDecimal을 다른 것으로 뺀 값을 반환한다.
Exp	제공된 값으로 e를 거듭제곱한 값을 반환한다.
Log	제공된 TiberoDecimal의 제공된 값의 로그값을 반환한다.
Pow	제공된 TiberoDecimal을 제공된 값으로 거듭제곱한 값을 반환한다.
Acos	cosine이 제공된 TiberoDecimal인 각도를 라디안 단위로 반환한다.
Asin	sine이 제공된 TiberoDecimal인 각도를 라디안 단위로 반환한다.
Atan	tangent가 제공된 TiberoDecimal인 각도를 라디안 단위로 반환한다.
Atan2	tangent가 제공된 두 TiberoDecimal의 몫인 각도를 라디안 단위로 반환한다.
Cos	라디안 단위로 제공된 각도의 cosine 값을 반환한다.
Sin	라디안 단위로 제공된 각도의 sine 값을 반환한다.
Tan	라디안 단위로 제공된 각도의 tangent 값을 반환한다.
Cosh	라디안 단위로 제공된 각도의 hyperbolic cosine 값을 반환한다.

메소드	설명
Sinh	라디안 단위로 제공된 각도의 hyperbolic sine 값을 반환한다.
Tanh	라디안 단위로 제공된 각도의 hyperbolic tangent 값을 반환한다.

- 메소드

  다음은 TiberoDecimal에서 제공하는 메소드이다.

메소드	설명
Equals	객체가 TiberoDecimal의 인스턴스인지 여부와 객체의 값이 현재 인스턴스와 같은지 여부를 결정한다.
ToByte	현재 인스턴스를 나타내는 byte를 반환한다.
ToDouble	현재 인스턴스를 나타내는 double을 반환한다.
ToInt16	현재 인스턴스를 나타내는 Int16을 반환한다.
ToInt32	현재 인스턴스를 나타내는 Int32을 반환한다.
ToInt64	현재 인스턴스를 나타내는 Int64를 반환한다.
ToSingle	현재 인스턴스를 나타내는 Single을 반환한다.
ToString	현재 인스턴스를 나타내는 String을 반환한다.

3.2.8. TiberoIntervalDS

- 속성

  다음은 TiberoIntervalDS에서 제공하는 속성이다.

속성	설명
Days	TiberoIntervalDS의 days 값을 가져온다.
Hours	TiberoIntervalDS의 hours 값을 가져온다.
Milliseconds	TiberoIntervalDS의 milliseconds 값을 가져온다.
Minutes	TiberoIntervalDS의 minutes 값을 가져온다.
Nanoseconds	TiberoIntervalDS의 nanoseconds 값을 가져온다.
Seconds	TiberoIntervalDS의 seconds 값을 가져온다.
Value	TiberoIntervalDS에 저장되는 시간 간격을 지정한다.

3.2.9. TiberoIntervalYM

- 속성

  다음은 TiberoIntervalYM에서 제공하는 속성이다.

속성	설명
Months	TiberoIntervalYM의 months 값을 가져온다.
Value	TiberoIntervalYM에 저장되는 총 개월 수를 지정한다.
Years	TiberoIntervalYM의 years 값을 가져온다.

3.2.10. TiberoString

- 속성

  다음은 TiberoString에서 제공하는 속성이다.

속성	설명
IsCaseIgnored	문자열 비교를 수행하는 경우 대소문자를 무시해야 하는지 여부를 나타낸다.
IsNull	현재 인스턴스가 null 값을 가졌는지 여부를 나타낸다.
Item	인덱스를 사용해 TiberoString의 특정 문자를 가져온다.
Length	TiberoString의 길이를 반환한다.
Value	TiberoString의 string 데이터를 반환한다.

- 정적 메소드

  다음은 TiberoString에서 제공하는 정적 메소드이다.

메소드	설명
Concat	두 TiberoString을 연결해 새로운 TiberoString을 반환한다.
Equals	두 TiberoString 값이 같은지 확인한다.
GreaterThan	첫 번째 TiberoString 값이 두 번째 값보다 큰지 확인한다.
GreaterThanOrEqual	첫 번째 TiberoString 값이 두 번째 값보다 크거나 같은지 확인한다.
LessThan	첫 번째 TiberoString 값이 두 번째 값보다 작은지 확인한다.
LessThanOrEqual	첫 번째 TiberoString 값이 두 번째 값보다 작거나 같은지 확인한다.
NotEquals	두 TiberoString 값이 같지 않은지 확인한다.

- 메소드

다음은 TiberoString에서 제공하는 메소드이다.

메소드	설명
Clone	TiberoString 객체의 사본을 만든다.
CompareTo	현재 인스턴스를 제공된 객체와 비교하고 상대 값을 나타내는 정수를 반환한다.
Equals	객체의 값이 현재 인스턴스와 같은 String 값인지 여부를 결정한다.
GetUnicodeBytes	TiberoString 값을 유니코드 형식의 byte 배열로 반환한다.
ToString	현재 인스턴스를 나타내는 String을 반환한다.

3.2.11. TiberoTimeStamp

- 속성

다음은 TiberoTimeStamp에서 제공하는 속성이다.

속성	설명
Day	TiberoTimeStamp의 day 값을 가져온다.
Hour	TiberoTimeStamp의 hour 값을 가져온다.
Millisecond	TiberoTimeStamp의 millisecond 값을 가져온다.
Minute	TiberoTimeStamp의 minute 값을 가져온다.
Month	TiberoTimeStamp의 month 값을 가져온다.
Nanosecond	TiberoTimeStamp의 nanosecond 값을 가져온다.
Second	TiberoTimeStamp의 second 값을 가져온다.
Value	TiberoTimestamp의 날짜와 시간을 가져온다.
Year	TiberoTimeStamp의 year 값을 가져온다.

3.3. UDT

다음은 TDP.NET UDT에 대한 구현 목록이다.

3.3.1. TiberoCustomTypeMappingAttribute

- 속성

다음은 TiberoCustomTypeMappingAttribute에서 제공하는 속성이다.

속성	설명
UdtTypeName	사용자 정의 클래스가 매핑하는 Tibero 사용자 정의 타입 이름을 지정한다.

3.3.2. TiberoObjectMappingAttribute

- 속성

  다음은 TiberoObjectMappingAttribute에서 제공하는 속성이다.

속성	설명
AttributeIndex	검색해야하는 Tibero 오브젝트 속성의 인덱스를 지정한다.
AttributeName	검색해야하는 Tibero 오브젝트 속성의 이름을 지정한다.

3.3.3. ITiberoCustomType

- 메소드

  다음은 ITiberoCustomType에서 제공하는 메소드이다.

메소드	설명
FromCustomObject	지정된 Tibero UDT로 Tibero Object 또는 Collection을 만든다.
ToCustomObject	지정된 Tibero UDT를 사용하여 사용자 정의 객체를 만든다.

3.3.4. ITiberoCustomTypeFactory

- 메소드

  다음은 ITiberoCustomTypeFactory에서 제공하는 메소드이다.

메소드	설명
CreateObject	Tibero Object 또는 Collection을 나타내는 새 사용자 정의 객체를 반환한다.

3.3.5. ITiberoArrayTypeFactory

- 메소드

  다음은 ITiberoArrayTypeFactory에서 제공하는 메소드이다.

메소드	설명
CreateArray	Tibero Collection 요소를 저장하기 위한 지정된 길이의 배열을 반환한다.

3.3.6. TiberoUdt

- 정적 메소드

  다음은 TiberoUdt에서 제공하는 정적 메소드이다.

메소드	설명
GetValue	지정된 Tibero UDT에서 속성 또는 요소를 가져온다.
SetValue	지정된 Tibero UDT에서 속성 또는 요소를 설정한다.

3.4. Bulk Copy

다음은 TDP.NET Bulk Copy에 대한 구현 목록이다.

3.4.1. TiberoBulkCopy

- 속성

  다음은 TiberoBulkCopy에서 제공하는 속성이다.

속성	설명
BatchSize	배치로 데이터베이스에 보낼 행 수를 지정한다.
BulkCopyTimeout	BulkCopy의 Timeout을 초 단위로 지정한다.
ColumnMappings	데이터소스와 대상 테이블간의 열 매핑을 지정한다.
Connection	BulkCopy를 수행하는데 사용되는 TiberoConnection 객체를 지정한다.
DestinationPartitionName	대상 파티션의 이름을 지정한다.
DestinationTableName	대상 테이블의 이름을 지정한다.
NotifyAfter	알림 이벤트가 생성되기 전에 처리 될 행 수를 정의한다.

- 메소드

  다음은 TiberoBulkCopy에서 제공하는 메소드이다.

메소드	설명
Close	TiberoBulkCopy 인스턴스를 닫는다.

메소드	설명
Dispose	할당된 자원과 메모리를 해제한다.
WriteToServer	제공된 DataTable의 행들을 대상 테이블로 복사한다.

3.4.2. TiberoBulkCopyColumnMapping

- 속성

  다음은 TiberoBulkCopyColumnMapping에서 제공하는 속성이다.

속성	설명
DestinationColumn	매핑 대상 테이블의 열 이름을 지정한다.
DestinationOrdinal	매핑 대상 테이블의 열 서수값을 지정한다.
SourceColumn	매핑 중인 데이터소스의 열 이름을 지정한다.
SourceOrdinal	매핑 중인 데이터소스의 열 서수값을 지정한다.

3.4.3. TiberoBulkCopyColumnMappingCollection

- 속성

  다음은 TiberoBulkCopyColumnMappingCollection에서 제공하는 속성이다.

속성	설명
Item[index]	지정된 인덱스의 객체를 가져오거나 설정한다.

- 메소드

  다음은 TiberoBulkCopyColumnMappingCollection에서 제공하는 메소드이다.

메소드	설명
Add	객체를 컬렉션에 추가한다.
Clear	컬렉션을 비운다.
Contains	지정된 TiberoBulkCopyColumnMapping 객체가 컬렉션에 존재하는지를 나타내는 값을 반환한다.
CopyTo	지정된 인덱스부터 요소를 복사해 TiberoBulkCopyColumnMapping의 배열로 반환한다.
IndexOf	지정된 TiberoBulkCopyColumnMapping 객체의 인덱스를 반환한다.
Insert	지정된 인덱스에 새로운 TiberoBulkCopyColumnMapping 객체를 추가한다.

메소드	설명
Remove	컬렉션에서 지정된 TiberoBulkCopyColumnMapping을 제거한다.
RemoveAt	컬렉션에서 지정된 인덱스의 TiberoBulkCopyColumnMapping을 제거한다.

3.4.4. TiberoRowsCopiedEventArgs

- 속성

  다음은 TiberoRowsCopiedEventArgs에서 제공하는 속성이다.

속성	설명
Abort	BulkCopy 작업이 중단되었는지 여부를 나타내는 값을 검색하거나 설정한다.
RowsCopied	현재 BulkCopy 작업 중 복사된 행 수를 나타내는 값을 검색한다.