DLL DataFetcher の機能

アクセス型のデータベースを作成・更新・削除・複雑な検索機能を持つＶｉｓｕａｌ　Ｃ＃のライブラリ群

データベース構造（複数のテーブルを持ち、テーブルの構造は下記）

フィールド（それぞれフィールド名を持つ）

ID	A	B	C	D	E	F	G	……	Z
S001	alpha	190	…						
X005									
T010									
U09									
V1									

レコード（文字列または数値）

DLL DataFetcher の構造

namespace DataFetcher

public class DataHandler

=> DataFetcherを使うときはStringHandler.dllとControlLibrary.exeを参照設定し、読み出し側に

using StringHandler;

using ControlLibrary;

の宣言と下記のインスタンス設定が必要。

internal pickTop alpha（任意名） = new pickTop();//StringHandler
internal Analog gamma（任意名） = new Analog();//ControlLibrary

また、下記が空文字指定の場合、下記がdefault設定になる。

internal string strProvider = “Provider=Microsoft.ACE.OLEDB.12.0;Data Source=”;
internal string DB_path = （データベースを保存しているアドレス）

ライブラリ目次

No.	プログラム名称	機能
1	getFileLast	ファイル全体のパス名と最下位ファイル/フォルダ名を分離して出力する。
2	connectDB	OleDbシステムでOleDbConnectionを使ってデータベースを読み込む。
3	listDBtables	DB中のTable名をすべて取り出してリストにする。
4	transDBtoDT	DB中の１Tableを丸ごとDataTableに置き換える
5	listFieldName	DB中の１Tableのフィールド名のリストを作る
6	getFieldNum	DB中の指定した表中のフィールド名を与え、それに一致するそのフィールド番号を返す
7	packFieldName	DB中の指定した表のフィールド名をList<string>に変換する
8	listFieldData	DB中の指定した表の指定したフィールド内のデータ全体から重複のないデータのリストを作る
9	selectFieldData	DB中の指定した表から条件付で抽出したデータをDataTableに一旦書き出し、フィールド名を指定して列データを抽出し、文字配列に書き出す。

ライブラリ目次

No.	プログラム名称	機能
10	outDBformat	DBの1レコードのフィールド名とデータの組を順にテキストファイルに書き出して（書式：フィールド番号：フィールド名<<フィールドデータ>>）、保存する。
11	intakeArray	outDBFormatで作成したテキストファイルのデータを文字配列に出力する。（outDBFormatの逆操作）
12	arrayGetter	Tableの行列データ[numColumns,numRows]をそのまま抽出し、文字配列に出力する。
13	listGetter	データベース中のテーブルの行列データの指定した列からそのまま(intMode=0)または重複なし(intMode=1）に列データを抽出し、文字配列に出力する。
14	pickTable	DataTable の列名がstrTargetであるデータを1件のみ抽出し、行データをstrDelimiterで区切って単一の文字変数に戻す。
15	getOneRecord	指定した表中の指定したフィールドのデータが検索語であるレコードがあれば、該当レコードに含まれるデータを全て取り出す

ライブラリ目次

No.	プログラム名称	機能
16	deleteRecord	データベースのテーブル中の指定したフィールドのデータが検索語と一致したレコードを削除する。
17	transSQLinsert	データベース中のテーブルのフィールド名のリストを使ってレコード新規挿入用のSQL コマンドを出力する。actEntryNewとセットで使う。
18	transSQLupdate	リストに収納されたテーブルのフィールドからを、指定したフィールド番号のデータ（別途与える）と一致するデータを抽出する汎用のSQLコマンドを生成する。actUpdateとセットで使う。
19	transSQLfetch	テーブル中のフィールドの中で文字配列strFieldで指定されるフィールドのデータがactFetch（別途）で指定される文字列を含むもの全てを抽出する汎用のSQLコマンドを生成する。
20	actEntryNew	データベース中のテーブルのフィールド名列リストからデータをレコード追加するＳＱＬコマンドを使ってデータベースに新規レコードを追加する。
21	actUpdate	データベースのテーブルのフィールド名列strFieldのintWhere番目のデータを更新したstrBase[]を入れてデータベースを更新する。
22	actFetch	データベース中のテーブルの、フィールド名strField[](複数）のデータがそれぞれstrData[](複数）であるレコードをすべて抽出する。

プログラム名	public string[] getFileLast(string strPath)	
機能	ファイル/フォルダ全体のパス名strPathから、Path名string[0]とファイル/フォルダ名string[1]を分離して出力する。	
引数	strPath	データベースのパス名(string)
戻り変数	文字変数 (string[2])	string[0]：path名 string[1]：最下位ファイル/フォルダ名
使用例	string[] filePart = new string[2]; filePart = beta.getFileLast(strIOFolder[i]);	

プログラム名	public string connectDB(string strChannel, string strDbase)	
機能	OleDbシステムでOleDbConnectionを使ってstrChannel(プロバイダ； デフォールトはstrProvider)+ strDbase(データベースパス)により、 データベースstrDbaseを読み込む。	
引数	strChannel	プロバイダパス(string) Default:“Provider=Microsoft.ACE.OLEDB.12.0;Data Source=”
	strDbase	データベースパス(string) Default:@"I:¥002_Homepage¥homepage_ja¥MarineViewer_forWeb¥Access¥Marine_Life.accdb"
戻り変数	データベースの作成期日	string
使用例	string strConnect = beta.connectDB(strProvider, reefDB_path);	

プログラム名	public string[] listDBtables(string strChannel, string strDbase)	
機能	DB(パスstrDbase)中のTable名をすべて取り出す	
引数	strChannel	プロバイダ (string)
	strDbase	データベースパス(string)
戻り変数	データベース内の テーブル名全て	string[]
使用例		

プログラム名	public DataTable transDBtoDT(string strChannel, string strDbase, string strTable, string strOrder)	
機能	DB中の1Table　strTableを丸ごとDataTableに置き換える	
引数	strChannel	プロバイダパス(string)
	strDbase	データベースパス(string)
	strTable	テーブル名(string)
	strOrder	オーダーを並べなおす(SQL　ORDER　BY　Field名)
戻り変数	DataTable	テーブル内のデータを全て取り込んだDataTable
使用例	dTView = beta.transDBtoDT(strProvider, reefDB_path, strTable, strOrder); DbViewReef.DataSource = dTView;	

プログラム名	public string[] listFieldName(string strChannel, string strDbase, string strTable)	
機能	DB中の1Table　strTableのフィールド名strFieldのリストを作る	
引数	strChannel	プロバイダパス(string)
	strDbase	データベースパス(string)
	strTable	テーブル名(string)
戻り変数	フィールド名の配列	string[]
使用例	string[] strField = beta.listFieldName(strProvider, reefDB_path, ″Life_Data″);	

プログラム名	public int getFieldNum(string strChannel, string strDbase, string strTable, string FieldName)	
機能	DB、1テーブル中のfield名を与え、それに一致するそのフィールド番号を返す	
引数	strChannel	プロバイダパス(string)
	strDbase	データベースパス(string)
	strTable	テーブル名(string)
	FieldName	フィールド名(string)
戻り変数	フィールド番号	int;0,1,2…
使用例		

プログラム名	public List<string> packFieldName(string strChannel, string strDbase, string strTable)	
機能	DB(strChannel+strDbase)中の1Table strTableのフィールド名strFieldをList generic <string>に変換する	
引数	strChannel	プロバイダパス(string)
	strDbase	データベースパス(string)
	strTable	テーブル名(string)
戻り変数	フィールド名のList	List<string>
使用例	stQuizField = beta.packFieldName(strProvider, MarinePath, "QuizSource");	

プログラム名	public string[] listFieldData(string strChannel, string strDbase, string strTable, string strField)	
機能	DB中の1Table　strTableのフィールド名strField内のデータ列から重複のないデータのリストを作る	
引数	strChannel	プロバイダパス(string)
	strDbase	データベースパス(string)
	strTable	テーブル名(string)
	strField	フィールド名(string)
戻り変数	重複のないデータの配列	string[]
使用例	string[] strLifeName = beta.listFieldData(strProvider, MarinePath, ″Life_Data″, ″JapanName″);	

プログラム名	public string[] selectFieldData(string strChannel, string strDbase, string strTable, string strField,string strWhere)	
機能	DB中の1Table　strTableのフィールド名strField内のデータ列から重複のないデータのリストを作る	
引数	strChannel	プロバイダパス(string)
	strDbase	データベースパス(string)
	strTable	テーブル名(string)
	strField	フィールド名(string)
	strWhere	strTableからのレコードの抽出条件(string、SQLコマンド)
戻り変数	データの文字配列	string[]
使用例		

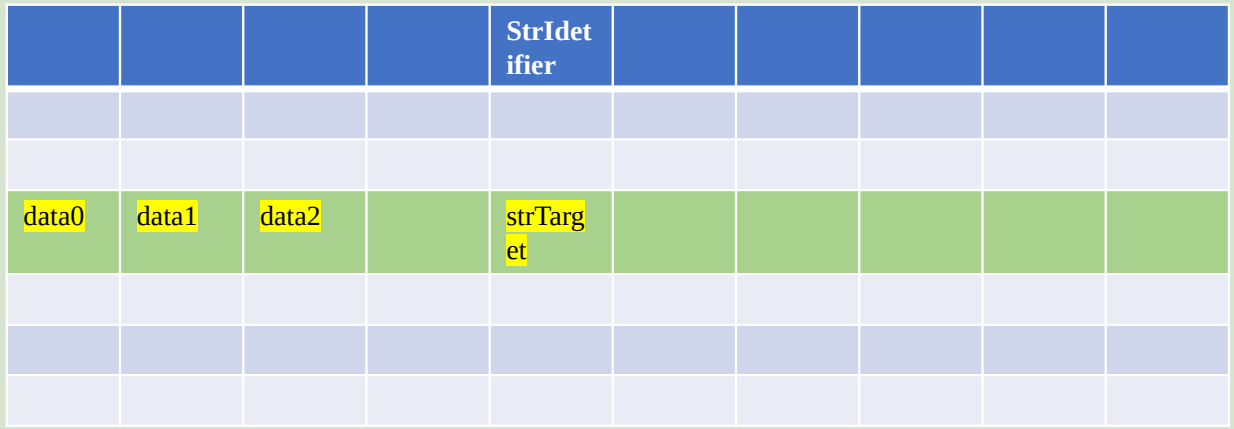
プログラム名	public void outDBformat(string[] strField, string[] strData, string outFile)	
機能	DBのフィールド名strFieldとデータstrDataの組を順にテキストファイルoutFileに書き出して（書式：フィールド番号：フィールド名strField<<フィールドデータstrData>>）、保存する。	
引数	strField	フィールド名 (string[])
	strData	データ(string[])
	outFile	出力パス(string、.txt)
戻り変数	なし	void
使用例	outPath = txtFolder.Text + ″¥temp.txt″; beta.outDBformat(strField, strBase, outPath);	

プログラム名	public string[] intakeArray(string intakePath)	
機能	intakePath のDBフォーマットのデータを文字配列に出力する。 （outDBFormatの逆操作）	
引数	intakePath	読み込むファイルのパス (string)
戻り変数	フィールド番号順のデータの配列	string[]
使用例	string[] strData = beta.intakeArray(outFile);	

プログラム名	public string[,] arrayGetter(string fileDB, string strTable)	
機能	Tableの行列データstring[numColumns,numRows]をそのまま抽出し、 strArrayに出力する。	
引数	fileDB	データベース名(string)(プロバイダパスはarrayGetter内で指定)
	strTable	DB中のテーブル名(string)
戻り変数	データの配列	string[,](2次元)
使用例		

プログラム名	public string[] listGetter(string fileDB, string strTable, int intCol, int intMode)	
機能	データベースfileDB中のテーブルTableの行列データのintCol 番目の列から、そのまま(intMode=0)または重複なし(intMode=1)に列データを抽出し、文字配列に出力する。	
引数	fileDB	データベース名(string)
	strTable:	DB中のテーブル名(string)
	intCol	行番号(int)
	intMode	int：0または1
戻り変数	データの配列	string[]
使用例		

プログラム名	public string pickTable(string strIdentifier, string strTarget, DataTable dTable, string strDelimiter)	
機能	DataTable dTableの列名strIdentifierの値がstrTargetであるデータを1件のみ抽出し、行データをstrDelimiterで区切って単一の文字変数に戻す。	
引数	dTable	DataTable名(string)
	strIdetifier	対象の列名(string)
	strTarget	検索名(string)
	strDelimiter	行データを接続する時の記号(string:たとえば"#")
戻り変数	文字変数	string
使用例	この列のデータを抽出	



戻り変数:data0#data1#data2#...strTarget#...

プログラム名	public string[] getOneRecord(string strChannel, string strDbase,string strTable,string strField,string strKey)	
機能	(Accessなど)テーブルstrTable中のフィールド名strFieldの(重複のない)データが検索ワードstrKeyであるレコードがあれば、該当レコードに含まれるデータを全て取り出す	
引数	strChannel	プロバイダパス(string)
	strDbase	データベースパス(string)
	strTable	テーブル名(string)
	strField	フィールド名(string)
	strKey	検索データ(string)
戻り変数	条件に合致したレコード全ての配列	string[]
使用例	string[] strBase = beta.getOneRecord(strProvider, MarinePath, ″Life_Data″, ″JapanName″, strFishName);	

プログラム名	public void deleteRecord(string connectDB, string strTable, string strField, string stRecordID)	
機能	データベースのテーブル中のフィールドのデータがstrRecordIDである レコードを検索して削除する。	
引数	connectDB	プロバイダパス(string)＋データベースパス(string)
	strTable	テーブル名(string)
	strField	フィールド名(string)
	strRecordID	検索データ(string)
戻り変数	無し	void
使用例	beta.deleteRecord(connectDB, ″clrComparison″ , ″setName″, strLifeID);	

プログラム名	public string transSQLinsert(List<string> lstField, string strTable)	
機能	アクセスDBのstrTableのフィールド名のList、listFieldを使ってレコード新規挿入用のSQLコマンドをstringに収める。	
引数	lstField	データベースのテーブル内フィールド名のリスト(List<string>)
	strTable	テーブル名(string)
戻り変数	集合SQLコマンド	string
使用例	strSQLinsert = beta.transSQLinsert(lstLife, "Life_Data");	

プログラム名	public void actEntryNew(string strConnect, string strSQL,　string[]　strField, string[] strData)	
機能	プロバイダ＋データベースのpath　strConnectにある、テーブルstrTableのフィールド名列lstTempからデータstrBase[]をレコード追加するSQLコマンドを作り、データベースに新規レコード追加する。	
	strConnect	プロバイダ＋データベースのパス名(string)
引数	strSQL	SQL コマンド(string:INSERT INTO Table名　)
	lstTemp	フィールド名列(List<string>)
	strData	登録するデータの文字配列（string[]）
戻り変数	無し。	
使用例	intStatus = beta.actEntryNew(strProvider + reefDB_path, strSQLinsert, strField, strBase);	

プログラム名	public string transSQLupdate(List<string> lstField, string strTable, int intWhere)	
機能	List<string>に収納されたテーブル名strTableのフィールド名の配列データを、 フィールド番号intWhereのデータ(別途与えられる)と一致するものを 抽出する汎用のSQLコマンドを生成する。	
引数	lstField	データベースのテーブル内フィールド名のリスト(List<string>)
	strTable	テーブル名(string)
	intWhere	抽出条件を与えるフィールド番号(int)
戻り変数	集合SQLコマンド	string: “UPDATE Table名　SET field名[0]=?, field名[1]=?, …., field名[intWhere-1]=?、field名[intWhere+1]=?,…, WHERE field名[intWhere]=?
使用例	strSQLupdate = beta.transSQLupdate(lstLife, ″Life_Data″, intWhere);	

プログラム名	public int actUpdate(string strConnect, string strSQL, string[] strField, string[] strData,int intWhere)	
機能	データベースのpath　strConnectにある、テーブルstrTableのフィールド名列strFieldのintWhere番目のデータを更新したstrBase[]を入れてデータベースを更新する。	
引数	strConnect	プロバイダ＋データベースのパス名(string)
	strSQL	SQL コマンド(string:INSERT INTO Table名　)
	lstTemp	フィールド名列(List<string>)
	strData	登録するデータの文字配列（string[]）
戻り変数	結果の表示	Int（登録成功なら＝１）
使用例	intStatus = beta.actUpdate(strProvider + reefDB_path, strSQLupdate, strField, strBase, intWhere);	

プログラム名	public string transSQLfetch(string[] strField, string strTable)	
機能	テーブル中のフィールドの中で文字配列strFieldで指定されるフィールドのデータがactFetch(別途)で指定される文字列を含むもの全てを抽出する汎用のSQLコマンドを生成する。	
引数	strField	フィールド名を含むList(List<string>)。フィールド数intField。
	strTable	テーブル名(string)
戻り変数	集合SQLコマンド	string: “SELECT * FROM Table名　SET field名[0]=LIKE , field名[1]LIKE?,…,　field名[intField]LIKE?
使用例	string strSQL = beta.transSQLfetch(strField, ″Life_Data″);	

プログラム名	public DataTable actFetch(string strConnect, string strSQL, string[] strField, string[] strData)	
機能	データベースの path　strConnectにあるテーブルstrTableの、フィールド名strField[](複数)のデータがそれぞれstrData[](複数)であるレコードをすべて抽出し、 データベースと同じ構造のデータテーブルに収納する。	
	strConnect	プロバイダ＋データベースのパス名(string)
引数	strSQL	SQL コマンド(string:INSERT INTO Table名　)
	strField	条件を与えるフィールド名(string[])
	strData	該当フィールドが満たすべきデータ(string[])
戻り変数	データテーブル	
使用例	dTView = beta.actFetch(strProvider + reefDB_path, strSQL, strField, strData);	