

ACT 距離計算パッケージ

Version 8.0

距離計算パッケージ
距離計算コアライブラリ
関数リファレンス

ACT 距離計算パッケージ

Version 8.0

距離計算パッケージ・距離計算コアライブラリ・関数リファレンス

1997年 9月20日	初版発行
1998年 2月10日	改訂第1版発行
2000年 3月31日	改訂第2版発行
2001年 3月 1日	改訂第3版発行
2001年 8月 1日	改訂第4版発行
2010年 7月 5日	改訂第5版発行 (Version 4.0)
2011年 1月21日	改訂第6版発行 (Version 5.0)
2012年 3月26日	改訂第7版発行 (Version 6.0)
2013年 3月15日	改訂第8版発行 (Version 7.0)
2014年 4月 7日	改訂第9版発行
2016年 5月11日	改訂第10版発行 (Version 8.0)
2017年 3月22日	改訂第11版発行

編著者・発行人

アドバンスド・コア・テクノロジー株式会社

〒105-0004 東京都港区新橋3-7-4 赤レンガ通りビル2階

電話 03-5512-9021 FAX 03-5512-9022

e-mail mail@act-inc.co.jp

本書に記載されている事項は、予告なしに変更されることがあります。

アドバンスド・コア・テクノロジー株式会社は本書に記載されている事項に関して一切の責任を負いかねますのでご了承ください。

本書の一部または全部をアドバンスド・コア・テクノロジー株式会社の書面による承諾なしに複製することは禁じられています。

Copyright (C) 1996-2017 by Advanced Core Technologies, Inc.

All rights reserved. No part of the contents of this book may be reproduced or transmitted in any form or by any means without the written permission of the publisher.

Windows は米国マイクロソフト社の登録商標です。

MapInfo および MapBasic は米国ピツニーボウズ・ソフトウェア社の登録商標です。

本書掲載の製品または製品名称は各社の商標または登録商標です。

ACT 距離計算パッケージ 距離計算コアライブラリ 関数リファレンス

目次

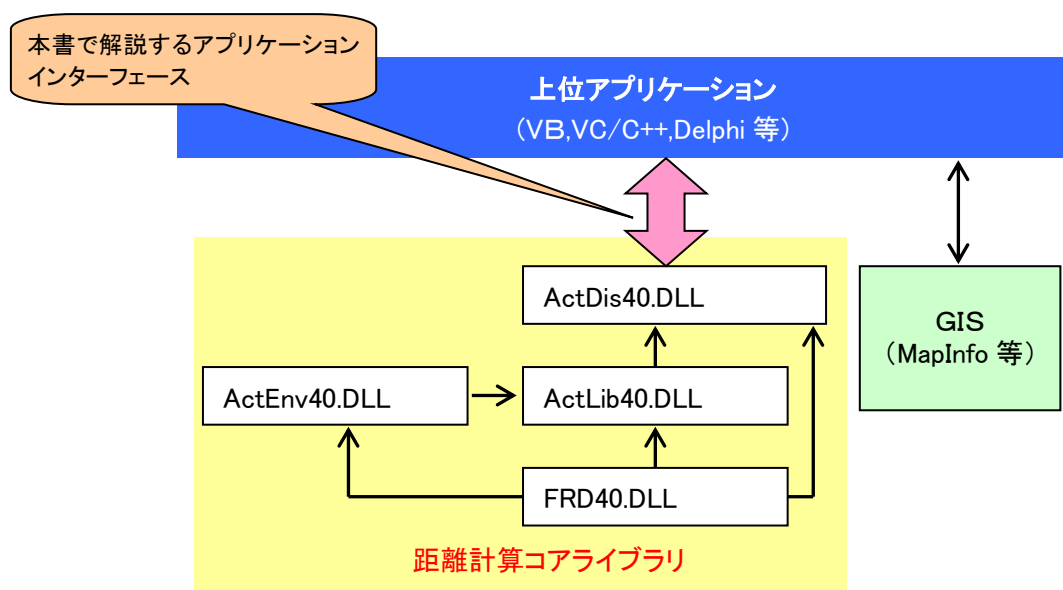
1. はじめに	1-1
2. 稼働環境	2-1
3. プログラム作成上の注意点	3-1
4. 関数の種類	4-1
5. 関数一覧	5-1
6. 稼働環境設定関数	6-1
7. 計算モード開始／終了関数	7-1
8. 計算実行関数	8-1
9. 計算結果取得関数	9-1
10. リンク速度管理関数	10-1
11. 通行料金計算関数	11-1
12. 区間距離表計算関数	12-1
13. 2点間距離計算関数	13-1
14. 最適拠点選択計算関数	14-1
15. 最適拠点立地計算関数	15-1
16. その他の関数	16-1
17. 定数／構造体一覧	17-1

1. はじめに

本書は、距離計算コアライブラリの距離計算関連関数のコール手順を示したものです。

距離計算コアライブラリは、下図のとおり、ACTDIS40.DLL、ACTLIB40.DLL、FRD40.DLL、ActEnv40.DLLの4モジュールで構成されています。本書で解説する距離計算機能に関連する関数はすべてACTDIS40.DLLからエクスポートされています。

これらのモジュールは、距離計算パッケージのインストールフォルダ下に作成されます。



【Version 8.0 の変更点】

(1) DLL 名の変更・64 ビットモジュールの提供

Version 8.0 では、ActDis32.dll、ActEnv32.dll が廃止され、それぞれ、ActDis40.dll、ActEnv40.dll に変更されました。また、32ビットモジュールと64ビットモジュールの両方が提供されます。

なお、ActDis32.dll の関数は同じインタフェースで ActDis40.dll に引き継がれています。

(2) インストール・ライセンスチェック

ACT_DIS_SetWorkDir 関数コール時に、距離計算コアライブラリおよびライセンスが正しくインストールされていることをチェックします。距離計算コアライブラリおよびライセンスが正しくインストールされていない場合、ACT_DIS_SetWorkDir 関数は 0 を返します。

(3) 標準型ポリゴンの中間点拡張

標準型(凹凸型)ポリゴン作成関数(ACT_DIS_MakeSuperPolygon、ACT_DIS_CalcMultiArea)で、端点拡張モードがオンのとき、同時に中間点補完が行われます。

2. 稼働環境

距離計算コアライブラリは下記の Windows のバージョンに対応しています。

Windows 7 (32bit/64bit)

Windows 8.1 (32bit/64bit)

Windows 10 (32bit/64bit)

3. プログラム作成上の注意点

距離計算コアライブラリを操作するプログラムを作成する場合には以下の点に注意してください。

(1) 複数のスレッドからコアライブラリを同時にコールしない

本バージョンの距離計算コアライブラリは、マルチスレッドに対応しておりません。マルチスレッド環境で距離計算機能をご利用になる場合は、**ACT** 距離計算コア・エンタープライズ版をご使用ください。

(2) 複数プロセスからコアライブラリを同時にコールしない

距離計算コアライブラリは、ディスク上に保存された計算用道路データを参照します。また、一時ファイルをディスクに保存する場合があります。このため、複数プロセスがコアライブラリを同時に実行した場合に、これらのファイルの競合が発生することがあります。

(3) 同時に複数の計算用道路データを計算に使用しない

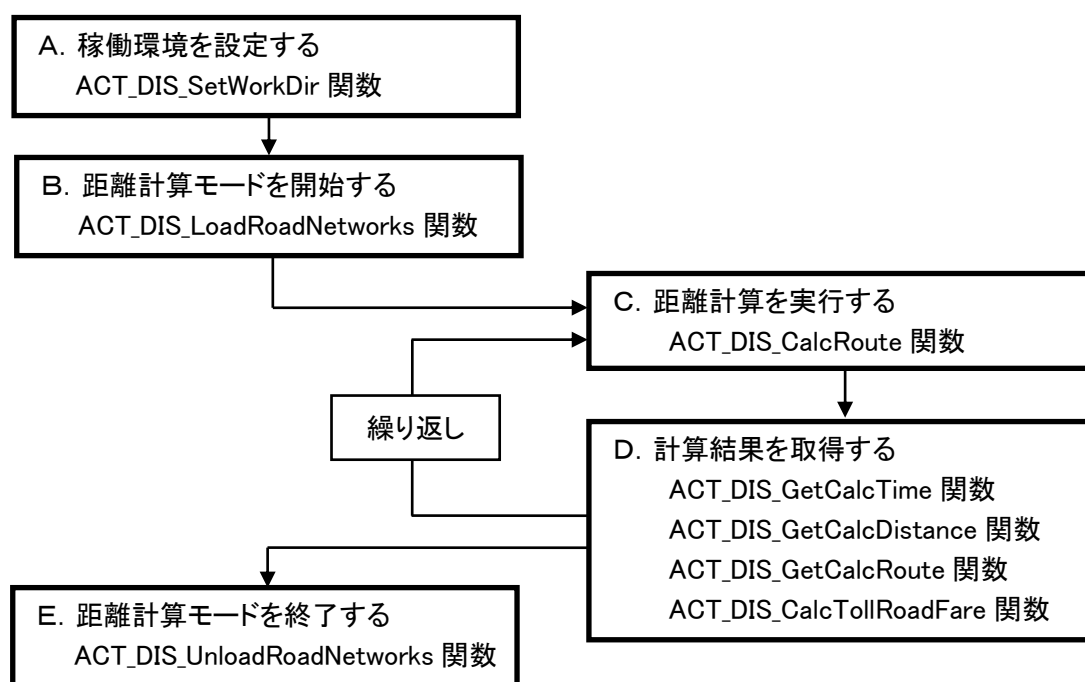
距離計算コアライブラリは、同時に複数の計算用道路データに対して計算を実行することができません。別の計算用道路データで計算を行うには、一旦計算を終了させてから、計算用道路データを指定しなおすようにして下さい。

4. 関数の種類

本書では、距離計算コアライブラリの関数群を、下記の 11 のカテゴリに分けて解説します。

項番	カテゴリ	解説
1	稼働環境設定関数	計算用道路データの在処(フォルダ)の指定や、計算用道路データの作成を行う関数
2	計算モード開始/終了関数	距離計算を実行することができる状態(計算モード)を開始/終了する関数
3	計算実行関数	距離計算を実行する関数
4	計算結果取得関数	上記の計算実行関数の計算結果(所要時間、道のり、ルート情報)を取得する関数
5	リンク速度管理関数	道路速度の取得/変更を行う関数
6	通行料金計算関数	通行料金データの設定や通行料金計算を行う関数
7	区間距離表計算関数	区間距離表計算を行う関数
8	2点間距離計算関数	2点間距離計算を行う関数
9	最適拠点選択計算関数	最適拠点選択計算を行う関数
10	最適拠点立地計算関数	最適拠点立地計算を行う関数
11	その他の関数	最短巡回順序の計算や、到達圏ポリゴンを作成する関数

距離計算を実行する最も基本的なシーケンスは、下図のとおりです。ここでは2点間のルート計算を行い、所要時間、道のり、ルートを取得しています。



5. 関数一覧

以下は、カテゴリ別の関数一覧です。

(1) 稼働環境設定関数

関数名	解説
ACT_DIS_GetACTDISVersion	距離計算コアライブラリのバージョン番号を取得します
ACT_DIS_SetWorkDir	計算用道路データが保存されているフォルダを指定します 【Version 8.0 機能変更関数】
ACT_DIS_GetWorkDir	現在指定されている計算用道路データのフォルダのフルパスを取得します
ACT_DIS_IsCalcNetworkExist	現在指定されたいる計算用道路データのフォルダに計算用道路データが存在するかどうかチェックします
ACT_DIS_BuildRoadNetworks400	計算用道路データを作成します

(2) 計算モード開始／終了関数

関数名	解説
ACT_DIS_LoadRoadNetworks	計算用道路データの一部をメモリにロードし(ACT1 型)、距離計算を実行できる状態にします
ACT_DIS_UnloadRoadNetworks	計算用道路データ用に確保されたメモリ領域を解放します(ACT1 型)
ACT_DIS_LoadRoadNetworksTrf	計算用道路データの一部をメモリにロードし(TRF1 型)、距離計算を実行できる状態にします
ACT_DIS_UnloadRoadNetworksTrf	計算用道路データ用に確保されたメモリ領域を解放します(TRF1 型)
ACT_DIS_IsRoadNetworksLoaded	計算用道路データの一部がメモリにロードされ距離計算可能状態にあるかどうかチェックします

(3) 計算実行関数

関数名	解説
ACT_DIS_CalcRoute	指定された2点間の最短時間経路を計算します
ACT_DIS_CalcFullRoute	指定された起点からの最短時間経路を計算します
ACT_DIS_CalcFullRouteEx	指定された起点から、指定時間(分単位)以内の最短時間経路を計算します
ACT_DIS_CalcFullRouteExDSec	指定された起点から、指定時間(1/10 秒単位)以内の最短時間経路を計算します
ACT_DIS_CalcRouteDist	指定された2点間の最短距離経路を計算します
ACT_DIS_CalcFullRouteDist	指定された起点からの最短距離経路を計算します
ACT_DIS_CalcFullRouteDistEx	指定された起点から、距離(m単位)以内の最短距離経路を計算します

(4) 計算結果取得関数

関数名	解説
ACT_DIS_GetCalcTime	指定された点までの所要時間(分単位)を求めます
ACT_DIS_GetCalcDSecTime	指定された点までの所要時間(1/10 秒単位)を求めます
ACT_DIS_GetCalcDistance	指定された点までの道のり(m単位)を求めます
ACT_DIS_GetCalcRouteXY	指定された点までのルート情報(XY座標の系列)を求めます
ACT_DIS_GetCalcRouteNode	指定された点までのルート情報(ノード番号の系列)を求めます
ACT_DIS_GetCalcRouteLink	指定された点までのルート情報(リンク番号の系列)を求めます
ACT_DIS_GetRouteNameFileName	文字ルート情報ファイルのフルパスを取得します
ACT_DIS_SetRouteNameFileName	文字ルート情報ファイルのフルパスを設定します
ACT_DIS_OutputCalcRouteName	文字ルートファイルに文字ルート情報を出力します
ACT_DIS_IsDetailRouteExist	詳細ルート情報取得が可能かどうかチェックします
ACT_DIS_GetCalcRouteDetail	詳細ルート情報(XY座標の系列)を取得します
ACT_DIS_GetCalcRouteDetailMIF	詳細ルート情報(XY座標の系列)をMIFファイルに出力します

(5) リンク速度管理関数

関数名	解説
ACT_DIS_ShowAllSpeed	すべての道路区間の速度一覧表を表示します
ACT_DIS_ChangeBothSpeedEx	指定されたリンクの速度(両方向)を設定します
ACT_DIS_ChangeSpeedBatch	指定されたリンクの速度をバッチ処理で変更します
ACT_DIS_GetSpeed	指定されたリンクの速度(片方向)を取得します
ACT_DIS_GetSpeedEx	指定されたリンクの速度(両方向)を取得します
ACT_DIS_CountTempSpeed	一時的速度が設定されているリンク数を取得します
ACT_DIS_CommitTempSpeed	一時的速度を恒久的速度として保存します
ACT_DIS_ClearTempSpeed	一時的速度変更を取り消し元の速度(恒久的速度)に戻します
ACT_DIS_ResumeOneWayInfo	右左折禁止情報から一方通行情報を抽出し、通行不可能な方向の速度をゼロに設定します

(6) 通行料金計算関数

関数名	解説
ACT_DIS_SetTollRoadFare	計算用道路データに通行料金データを設定します
ACT_DIS_GetNumFareVersion	利用可能な通行料金データのバージョン数を取得します
ACT_DIS_GetFareVersionText	通行料金データのバージョンテキストを取得します
ACT_DIS_IsFareExist	計算用道路データに通行料金データが設定されているかどうかをチェックします
ACT_DIS_CalcTollRoadFare	指定された点までの通行料金を取得します
ACT_DIS_CalcTollRoadFareDetail	通行料金明細データをファイルに出力します

(7) 区間距離表計算関数

関数名	解説
ACT_DIS_MTRX_ExecuteCalc	区間距離表計算を実行し、区間距離表計算結果内部データを作成します
ACT_DIS_MTRX_ExportCoreToMif	区間距離表計算結果内部データを変換し、MIF ファイルに出力します
ACT_DIS_MTRX_DeleteCoreResult	区間距離表計算結果内部データを削除します

(8) 2点間距離計算関数

関数名	解説
ACT_DIS_CalcFromToRoute	2点間距離計算を実行し、計算結果内部データを作成します

(9) 最適拠点選択計算関数

関数名	解説
ACT_DIS_CalcOptSite	最適拠点選択計算を実行し、計算結果内部データを作成します
ACT_DIS_CountCombination	最適拠点選択計算で計算する組み合わせの数を求めます
ACT_DIS_VerifyCombination	最適拠点選択計算実行の可否を尋ねるメッセージボックスを表示します
ACT_DIS_VerifyCombinationEx	最適拠点選択計算実行の可否を尋ねるメッセージボックスを表示します。固定地点の有無を考慮します。

(10) 最適拠点立地計算関数

関数名	解説
ACT_DIS_CalcOptPnt	最適拠点立地計算を実行し、計算結果内部データを作成します
ACT_DIS_VerifyOptPnt	最適拠点立地計算実行の可否を尋ねるメッセージボックスを表示します

(11) その他の関数

関数名	解説
ACT_DIS_CalcOptDSecTime	複数地点を巡回する最短時間経路を求めます
ACT_DIS_CalcOptDSecTimeEx	複数地点を巡回する最短時間経路を求めます。発着地点の固定の有無を指定することができます。
ACT_DIS_CalcOptDistance	複数地点を巡回する最短距離経路を求めます
ACT_DIS_CalcOptDistanceEx	複数地点を巡回する最短距離経路を求めます。発着地点の固定の有無を指定することができます。
ACT_DIS_MakePolygon	指定された到達時間／距離の外縁をなす簡易型(凸型)ポリゴンを作成します
ACT_DIS_MakeSuperPolygon	指定された到達時間／距離の外縁をなす標準型(凹凸型)ポリゴンを作成します 【Version 8.0 機能変更関数】
ACT_DIS_MakeHyperPolygon	指定された到達時間／距離の外縁をなす詳細型ポリゴンを作成します
ACT_DIS_GetNodeX	指定されたノードのX座標を取得します
ACT_DIS_GetNodeY	指定されたノードのY座標を取得します
ACT_DIS_GetNearestNode	指定されたXY座標に最も近いノードを求めます
ACT_DIS_IsHighwayNode	指定されたノードが高速道路上のノードかどうかを判定します
ACT_DIS_GetDistCalcIni	設定ファイル(DistCalc.ini)のフルパスを取得します
ACT_DIS_AboutDistCalc	距離計算パッケージのバージョン情報を表示します
ACT_DIS_CalcMultiArea	複数の到達圏／流入圏を一括計算します 【Version 8.0 機能変更関数】

6. 稼働環境設定関数

MapBasic や Visual Basic 等の簡易言語で上位アプリケーションを作成した場合、使用するライブラリを明示的にロードすることが必要です。ACT_DIS_GetACTDISVersion 関数を上位アプリケーションの起動と同時にコールすることで、距離計算コアライブラリをロードすることができます。

距離計算コアライブラリの各関数を正常に作動させるには、まず、ACT_DIS_SetWorkDir 関数を用いて、計算用道路データの在処(フォルダ)を指定することが必要です。距離計算コアライブラリのすべての関数は、ここで指定されたフォルダ内の計算用道路データに対して演算を行います。

```
long WINAPI ACT_DIS_GetACTDISVersion( void )
```

本関数は、距離計算コアライブラリのバージョン番号を取得します。

パラメータ	説明
-------	----

なし	
----	--

戻り値

距離計算コアライブラリのバージョン番号。20160325 という数値が返されます。

備考

MapBasic や Visual Basic 等の簡易言語で上位アプリケーションを作成した場合に、使用するライブラリを明示的にロードすることが必要です。ライブラリをロードするには、本関数をコールするか、Windows の LoadLibrary 関数をコールして下さい。

【Version 8.0 機能追加関数】

int WINAPI ACT_DIS_SetWorkDir(LPSTR lpszWorkDir)

LPSTR lpszWorkDir 計算用道路データが保存されているフォルダのフルパス

本関数は、計算用道路データが保存されているフォルダを指定します。

パラメータ	説明
LPSTR lpszWorkDir	計算用道路データが保存されているフォルダのフルパスを格納した文字列へのポインタ

戻り値

設定が正常に完了した場合は1。正常に完了しない場合は0。

備考

距離計算コアライブラリのすべての関数は、本関数で指定されたフォルダ下にある計算用道路データに対して演算を行います。尚、この指定は、距離計算コアライブラリがメモリからアンロードされる(アプリケーションを終了する)と消去されます。

距離計算コアライブラリ、およびライセンスが正しくインストールされていない場合は、本関数は 0 を返します。

```
int WINAPI ACT_DIS_GetWorkDir( LPSTR lpszWorkDir )
```

本関数は、現在指定されている計算用道路データのフォルダのフルパスを取得します。

パラメータ	説明
LPSTR lpszWorkDir	計算用道路データが保存されているフォルダのフルパスを取得する文字列へのポインタ。MAX_PATH(260)以上のサイズが必要です。

戻り値

正常に取得した場合は1。正常に取得できない場合は0。

備考

本関数は、現在指定されている計算用道路データのフォルダのフルパスを取得します。

```
int WINAPI ACT_DIS_IsCalcNetworkExist( void )
```

本関数は、現在指定されている計算用道路データのフォルダに計算用道路データが存在するかどうかをチェックします。

パラメータ	説明
-------	----

なし	
----	--

戻り値

計算用道路データが存在する場合は 1。存在しない場合は 0(ゼロ)。

備考

本関数は、ACT_DIS_SetWorkDir 関数で設定されているフォルダに、計算用道路データが存在するかどうかをチェックします。ACT_DIS_SetWorkDir 関数による計算用道路データ保管フォルダの指定が完了していない場合には、本関数は 0(ゼロ)を返します。

int WINAPI ACT_DIS_BuildRoadNetworks400(HWND hWnd, LPSTR lpszControlFile)

HWND hWnd 親ウィンドウのハンドル

LPSTR lpszControlFile コントロールファイルのフルパスへのポインタ

計算用道路データを作成します。

パラメータ	説明
hWnd	進捗状況を示すダイアログボックスの親ウィンドウのハンドル
lpszControlFile	各種道路データファイルを指定したコントロールファイルのフルパスを格納する文字列へのポインタ(詳細は備考を参照)

戻り値

計算用道路データの作成に成功した場合は 1。失敗した場合は 0(ゼロ)。

備考

本関数は、lpszControlFile で示されるコントロールファイルに指定された道路地図データを用いて、計算用道路データを、ACT_DIS_SetWorkDir 関数で設定されたフォルダ内に作成します。

コントロールファイルは、Windows の初期化ファイル(*.INI)に準じた形式のテキストファイルで、下記のセクションとエントリ(キー)からなっています。

セクション	エントリ(キー)	内容
DROTEN	FILENUM	道路点ファイルの数
	FILE01~FILE99	各道路点ファイル(MIF ファイル)のフルパス
DKUKAN	MIFOUTPUT	道路区間ファイルを MIF ファイル形式で出力したかどうかを示すフラグ。道路区間ファイルを MIF 形式で出力すると詳細ルートデータが作成されます。 0=CSV ファイルで出力、1=MIF ファイルで出力
	FILENUM	道路区間ファイルの数
	FILE01~FILE99	各道路区間ファイル(CSV ファイルまたは MIF ファイル)のフルパス
KUKANZ	FILENUM	区間内道路施設データファイルの数
	FILE01~FILE99	各道路区間内道路施設データファイル(CSV ファイル)のフルパス
TUKOUJ	FILENUM	通行情報データファイルの数
	FILE01~FILE99	各通行情報データファイル(CSV ファイル)のフルパス
TUKOUJ99	FILENUM	県跨り通行情報データファイルの数 (注)現バージョンでは0を指定します。
	FILE01~FILE99	各県跨り通行情報データファイル(CSV ファイル)のフルパス。(注)現バージョンでは指定しません。
DEFAULTSPEED	SPEEDA	順方向のデフォルト速度。1/10km/h 単位。
	SPEEDB	逆方向のデフォルト速度。1/10km/h 単位。
ROADFLAG	ICWEIGHT	IC 通過時のペナルティ時間。1/10 秒単位。

各データファイルのフォーマットは、道路点データと道路区間データが MapInfo の MIF 形式(*.MIF ファイルと*.MID ファイル)で、他のデータはCSV形式(カンマ区切り)です。尚、MIF 形式、CSV 形式とも、1行目にカラム名を出力してはいけません。

(1)道路点データ

*.MIF ファイルの形式は MapInfo の標準の MIF 形式でヘッダー部分は読み飛ばされます。データレコードの「Point」で始まるレコードのみがデータとして読み込まれます。オブジェクトのないレコードを出力しないようにしてください。

*.MID ファイルの各レコードのカラムは下記のとおりです。尚、本ファイルに関しては、下記のカラム以外のカラムが各レコードの最後にあってもかまいません。

項番	項目名称
1	都道府県コード
2	道路点ID
3	点種別
4	隣接都道府県コード
5	隣接道路点ID
6	交差点名称
7	接続道路数
8~15	区間(1)ID~区間(8)ID

(2)道路区間データ

*.MIF ファイルの形式は MapInfo の標準の MIF 形式でヘッダー部分は読み飛ばされます。データレコードの「PLine」または「Line」で始まるレコードのみがデータとして読み込まれます。オブジェクトのないレコードを出力しないようにしてください。

.MID ファイル(または.CSV ファイル)の各レコードのカラムは下記のとおりです。尚、下記道路区間データのカラム構成は、通常の道路区間テーブルのカラム構成と異なっていますので、一旦テーブルをセレクトしてから出力する必要があります。

項番	項目名称
1	区間ID
2	区間種別
3	都道府県コード
4	市区町村コード
5	区間長
6	幅員区分コード
7	車線数コード
8	接続点ID1
9	接続点ID2
10	路線番号
11	順方向スピード(1/10Km/h 単位の整数値)
12	逆方向スピード(1/10Km/h 単位の整数値)

Version 3.0 以降で詳細ルート表示機能を使用する場合には、道路区間データを MIF 形式で作成してください。

(3) 区間内道路施設データ

区間内道路施設データのサム構成は、通常の区間内道路施設テーブルと同じです。尚、本データの道路施設名称が、文字ルート情報の道路区間名称に利用されます。

項番	項目名称
1	都道府県コード
2	区間ID
3	道路施設種別
4	道路施設名称
5	道路施設延長

(4) 通行情報データ

通行情報データのサム構成は、通常の通行情報テーブルと同じです。尚、本データからは右左折禁止情報と一方通行情報が作成されます。

項番	項目名称
1	都道府県コード
2	道路点ID
3	通行区間ID
4	対象区間ID
5	通行規制
6	車種／時刻／曜日
7	異常気象
8	重量制限
9	高さ制限
10	幅制限
11	最高速度

(5) 県跨り通行情報ファイル

現バージョンでは、県跨り通行情報ファイルを処理しません。

例

下記は、実際に指定されたコントロールファイルの例です。この例では複数の道路点ファイルと道路区間ファイルを指定しています。

```
[DROTEN]
FILENUM=2
FILE01=C:\東京 95\計算地図\横断点.mif
FILE02=C:\東京 95\計算地図\道路点.mif
[DKUKAN]
MIFOUTPUT=1
FILENUM=2
FILE01=C:\東京 95\計算地図\横断区間.mif
FILE02=C:\東京 95\計算地図\道路区間.mif
[KUKANZ]
FILENUM=4
FILE01=C:\東京 95\計算地図\Kokudouz.csv
FILE02=C:\東京 95\計算地図\kousokuz.csv
FILE03=C:\東京 95\計算地図\Kukanz70.csv
FILE04=C:\東京 95\計算地図\Kukanz_c.csv
[TUKOUJ]
FILENUM=2
FILE01=C:\東京 95\計算地図\Tukouj70.csv
FILE02=C:\東京 95\計算地図\Tukouj_c.csv
[TUKOUJ99]
FILENUM=0
[DEFAULTSPEED]
SPEEDA=0
SPEEDB=0
[ROADFLAG]
ICWEIGHT=3000
```

7. 計算モード開始／終了関数

距離計算を実際に行うには、ACT_DIS_LoadRoadNetworks 関数 (ACT1 型)、または、ACT_DIS_LoadRoadNetworksTrf 関数 (TRF1 型) をコールして、ACT_DIS_SetWorkDir 関数で指定したフォルダ内の計算用道路データの一部をメモリにロードする必要があります。所定の計算作業が終了したら、ACT_DIS_UnloadRoadNetworks 関数 (ACT1 型)、または、ACT_DIS_UnloadRoadNetworksTrf 関数 (TRF1 型) をコールしてメモリを解放します。なお、これらの関数をコールしてメモリを解放すると、メモリ内に保持されている計算結果は失われます。

ACT1 型と TRF1 型について

距離計算コアライブラリでは、計算用道路データを作成したときに通行情報テーブルから抽出された右左折禁止情報を考慮して計算することができます。

本書のなかで、TRF1 型と呼ばれる計算方法が、右左折禁止情報を考慮する計算方法で、ACT1 型と呼ばれる計算方法は、右左折禁止を考慮しない計算方法です。

右左折禁止を考慮する計算 (TRF1 型) を行うには、考慮しない場合に比べて、約3倍のメモリと約3倍の計算時間を要しますので、通常は右左折禁止を考慮しない計算 (ACT1 型) を行うことをお勧めします。

long WINAPI ACT_DIS_LoadRoadNetworks(void)

本関数は、計算用道路データの一部をメモリにロードし (ACT1 型)、距離計算を実行できる状態にします。

パラメータ	説明
-------	----

なし	
----	--

戻り値

正常に終了した場合は1。正常に終了しなかった場合は0。

備考

右左折禁止を考慮しない方法で計算をするために、計算用道路データの一部をメモリにロードします。計算終了後は、ACT_DIS_UnloadRoadNetworks 関数をコールしてメモリを解放してください。

```
long WINAPI ACT_DIS_UnloadRoadNetworks( void )
```

計算用道路データ用に確保されたメモリ領域を解放します (ACT1 型)。

パラメータ	説明
-------	----

なし

戻り値

正常に終了した場合は1。正常に終了しなかった場合は0。

備考

本関数は、ACT_DIS_LoadRoadNetworks 関数でメモリにロードした計算用道路データを、メモリから解放します。これによって、メモリ内に保持されていた計算結果は消去されます。

long WINAPI ACT_DIS_LoadRoadNetworksTrf(void)

計算用道路データの一部をメモリにロードし(TRF1 型)、距離計算を実行できる状態にします。

パラメータ	説明
-------	----

なし

戻り値

正常に終了した場合は1。正常に終了しなかった場合は0。

備考

右左折禁止を考慮する方法で計算をするために、計算用道路データの一部をメモリにロードします。計算終了後は、ACT_DIS_UnloadRoadNetworksTrf 関数をコールしてメモリを解放してください。

long WINAPI ACT_DIS_UnloadRoadNetworksTrf(void)

計算用道路データ用に確保されたメモリ領域を解放します (TRF1 型)。

パラメータ	説明
-------	----

なし

戻り値

正常に終了した場合は1。正常に終了しなかった場合は0。

備考

本関数は、ACT_DIS_LoadRoadNetworksTrf 関数でメモリにロードした計算用道路データを、メモリから解放します。これによって、メモリ内に保持されていた計算結果は消去されます。

`int WINAPI ACT_DIS_IsRoadNetworksLoaded(void)`

本関数は、計算用道路データの一部がメモリにロードされ距離計算が可能な状態にあるかどうかチェックします。

パラメータ	説明
-------	----

なし

戻り値

計算用道路データの一部がメモリにロードされている場合は1。ロードされていない場合は0。

備考

計算用道路データの一部がメモリにロードされ距離計算が可能な状態にあるかどうかチェックします。ACT1 型、TRF1 型どちらでも利用できます。

8. 計算実行関数

距離計算を実行するには、ACT_DIS_LoadRoadNetworks 関数 (ACT_DIS_LoadRoadNetworksTrf 関数) をコールして、距離計算モードに移行した後に、以下の計算実行関数をコールします。

距離計算の計算方法には、「所要時間」を最短とする最短時間経路を求める方法と、「道のり」を最短とする最短距離経路を求める方法があり、それぞれの場合について別々の関数が用意されています。

計算の結果は、次節の「計算結果取得関数」で取得します。計算実行関数で計算した結果は、次に計算実行関数をコールして計算を行った場合、または、ACT_DIS_UnloadRoadNetworks 関数 (ACT_DIS_UnloadRoadNetworksTrf 関数) をコールして計算モードを終了するまで、メモリ上に保持されます。

```
long WINAPI ACT_DIS_CalcRoute( short nFromPrefCode, long lnFromNode,
                              short nToPrefCode, long lnToNode, BOOL bNoHighWay )
```

short	nFromPrefCode	発地点の都道府県コード
long	lnFromNode	発地点の道路点ID
short	nToPrefCode	着地点の都道府県コード
long	lnToNode	着地点の道路点ID
BOOL	bNoHighWay	高速道路使用フラグ

本関数は、指定された2点間の最短時間経路を計算します。

パラメータ	説明
nFromPrefCode	発地点の都道府県コード。2桁の数値。
lnFromNode	発地点の道路点ID。
nToPrefCode	着地点の都道府県コード。2桁の数値。
lnToNode	着地点の道路点ID
bNoHighWay	高速道路使用フラグ。高速道路を使用しない場合には TRUE(1)を、使用する場合には FALSE(0)を設定します。

戻り値

計算が正常に終了した場合は1。正常に終了しなかった場合は0。

発地点または着地点の道路点を示す都道府県コードまたは道路点IDに不備がある場合、これらの点が計算用道路データに存在しない場合、発地点から着地点に到達できなかった場合には、0(ゼロ)を返します。

備考

指定された2点間の最短時間経路を計算します。計算結果は、計算結果取得関数で取得します。

long WINAPI ACT_DIS_CalcFullRoute(short nStartPrefCode, long lnStartNodeID, BOOL bNoHighWay)

short	nStartPrefCode	発地点の都道府県コード
long	lnStartNodeID	発地点の道路点ID
BOOL	bNoHighWay	高速道路使用フラグ

指定された起点からの最短時間経路を計算します。

パラメータ	説明
nStartPrefCode	発地点の都道府県コード。2桁の数値。
lnStartNodeID	発地点の道路点ID。
bNoHighWay	高速道路使用フラグ。高速道路を使用しない場合には 1 を、使用する場合には 0 を設定します。 流入圏を計算する場合には 4096 (Hex 1000) を加えます。

戻り値

計算が正常に終了した場合は1。正常に終了しなかった場合は0。

発地点の道路点を示す都道府県コードまたは道路点IDに不備がある場合、この点が計算用道路データに存在しない場合には、0(ゼロ)を返します。

備考

指定された発地点から、計算用道路データ内のすべての道路点までの最短時間経路を計算します。計算結果は、計算結果取得関数で取得します。

Version 4.0 では bNoHighWay フラグの指定方法に流入圏計算の指定が追加されました。

```

long WINAPI ACT_DIS_CalcFullRouteEx( short nStartPrefCode, long lnStartNodeID,
                                     long lnThreshold, BOOL bNoHighWay )
short      nStartPrefCode      発地点の都道府県コード
long       lnStartNodeID       発地点の道路点ID
long       lnThreshold          到達時間
BOOL       bNoHighWay          高速道路使用フラグ

```

指定された起点から、指定時間(分単位)以内の最短時間経路を計算します

パラメータ	説明
nStartPrefCode	発地点の都道府県コード。2桁の数値。
lnStartNodeID	発地点の道路点ID。
lnThreshold	到達時間。分単位。
bNoHighWay	高速道路使用フラグ。高速道路を使用しない場合には 1 を、使用する場合には 0 を設定します。 流入圏を計算する場合には 4096 (Hex 1000)を加えます。

戻り値

計算が正常に終了した場合は1。正常に終了しなかった場合は0。

発地点の道路点を示す都道府県コードまたは道路点IDに不備がある場合、この点が計算用道路データに存在しない場合には、0(ゼロ)を返します。

備考

指定された発地点から、指定された時間内に到達できるすべての道路点までの最短時間経路を計算します。計算結果は、計算結果取得関数で取得します。

Version 4.0 では bNoHighWay フラグの指定方法に流入圏計算の指定が追加されました。

```

long WINAPI ACT_DIS_CalcFullRouteExDSec( short nStartPrefCode, long lnStartNodeID,
                                         long lnThreshold, BOOL bNoHighWay )
short      nStartPrefCode      発地点の都道府県コード
long      lnStartNodeID       発地点の道路点ID
long      lnThreshold          到達時間
BOOL      bNoHighWay          高速道路使用フラグ

```

指定された起点から、指定時間(1/10 秒単位)以内の最短時間経路を計算します。

パラメータ	説明
nStartPrefCode	発地点の都道府県コード。2桁の数値。
lnStartNodeID	発地点の道路点ID。
lnThreshold	到達時間。1/10 秒単位。
bNoHighWay	高速道路使用フラグ。高速道路を使用しない場合には 1 を、使用する場合には 0 を設定します。 流入圏を計算する場合には 4096 (Hex 1000)を加えます。

戻り値

計算が正常に終了した場合は1。正常に終了しなかった場合は0。

発地点の道路点を示す都道府県コードまたは道路点IDに不備がある場合、この点が計算用道路データに存在しない場合には、0(ゼロ)を返します。

備考

指定された発地点から、指定された時間内に到達できるすべての道路点までの最短時間経路を計算します。計算結果は、計算結果取得関数で取得します。

Version 4.0 では bNoHighWay フラグの指定方法に流入圏計算の指定が追加されました。

```
long WINAPI ACT_DIS_CalcRouteDist( short nFromPrefCode, long lnFromNode,
                                   short nToPrefCode, long lnToNode, BOOL bNoHighWay )
```

short	nFromPrefCode	発地点の都道府県コード
long	lnFromNode	発地点の道路点ID
short	nToPrefCode	着地点の都道府県コード
long	lnToNode	着地点の道路点ID
BOOL	bNoHighWay	高速道路使用フラグ

本関数は、指定された2点間の最短距離経路を計算します。

パラメータ	説明
nFromPrefCode	発地点の都道府県コード。2桁の数値。
lnFromNode	発地点の道路点ID。
nToPrefCode	着地点の都道府県コード。2桁の数値。
lnToNode	着地点の道路点ID
bNoHighWay	高速道路使用フラグ。高速道路を使用しない場合には TRUE(1)を、使用する場合には FALSE(0)を設定します。

戻り値

計算が正常に終了した場合は1。正常に終了しなかった場合は0。

発地点または着地点の道路点を示す都道府県コードまたは道路点IDに不備がある場合、これらの点が計算用道路データに存在しない場合、発地点から着地点に到達できなかった場合には、0(ゼロ)を返します。

備考

指定された2点間の最短距離経路を計算します。計算結果は、計算結果取得関数で取得します。

**long WINAPI ACT_DIS_CalcFullRouteDist(short nStartPrefCode, long lnStartNodeID,
 BOOL bNoHighWay)**

short	nStartPrefCode	発地点の都道府県コード
long	lnStartNodeID	発地点の道路点ID
BOOL	bNoHighWay	高速道路使用フラグ

指定された起点からの最短距離経路を計算します。

パラメータ	説明
nStartPrefCode	発地点の都道府県コード。2桁の数値。
lnStartNodeID	発地点の道路点ID。
bNoHighWay	高速道路使用フラグ。高速道路を使用しない場合には 1 を、使用する場合には 0 を設定します。 流入圏を計算する場合には 4096 (Hex 1000) を加えます。

戻り値

計算が正常に終了した場合は1。正常に終了しなかった場合は0。

発地点の道路点を示す都道府県コードまたは道路点IDに不備がある場合、この点が計算用道路データに存在しない場合には、0(ゼロ)を返します。

備考

指定された発地点から、計算用道路データ内のすべての道路点までの最短距離経路を計算します。計算結果は、計算結果取得関数で取得します。

Version 4.0 では bNoHighWay フラグの指定方法に流入圏計算の指定が追加されました。

**long WINAPI ACT_DIS_CalcFullRouteDistEx(short nStartPrefCode, long lnStartNodeID,
long lnThreshold, BOOL bNoHighWay)**

short	nStartPrefCode	発地点の都道府県コード
long	lnStartNodeID	発地点の道路点ID
long	lnThreshold	到達距離
BOOL	bNoHighWay	高速道路使用フラグ

指定された起点から、指定距離(m単位)以内の最短距離経路を計算します。

パラメータ

説明

nStartPrefCode	発地点の都道府県コード。2桁の数値。
lnStartNodeID	発地点の道路点ID。
lnThreshold	到達距離。m単位。
bNoHighWay	高速道路使用フラグ。高速道路を使用しない場合には1を、使用する場合には0を設定します。 流入圏を計算する場合には4096(Hex 1000)を加えます。

戻り値

計算が正常に終了した場合は1。正常に終了しなかった場合は0。

発地点の道路点を示す都道府県コードまたは道路点IDに不備がある場合、この点が計算用道路データに存在しない場合には、0(ゼロ)を返します。

備考

指定された発地点から、指定された距離内にあるすべての道路点までの最短時間経路を計算します。計算結果は、計算結果取得関数で取得します。

Version 4.0 では bNoHighWay フラグの指定方法に流入圏計算の指定が追加されました。

9. 計算結果取得関数

前節の「計算実行関数」で計算された結果は、次に計算実行関数をコールするか、計算モードを終了する (ACT_DIS_UnloadRoadNetworks 関数、ACT_DIS_UnloadRoadNetworksTrf 関数) まで、メモリ内に保持されています。本節の「計算結果取得関数」をコールすることで、この計算結果を取得することができます。

```
long WINAPI ACT_DIS_GetCalcTime( short nToPrefCode, long lnToNodeID )
```

```
short      nToPrefCode      着地点都道府県コード
```

```
long      lnToNodeID       着地点道路点ID
```

本関数は、指定された点までの所要時間(分単位)を求めます。

パラメータ	説明
nToPrefCode	着地点都道府県コード。2桁の数値。
lnToNodeID	着地点道路点ID

戻り値

値が取得できた場合は発地点から到達地点までの所要時間(分単位)を返します。値が取得できなかった場合は-1を返します。

着地点を示す都道府県コードまたは道路点IDが不備である場合、指定された点に到達できない場合、着地点が計算用道路データに存在しない場合は-1を返します。

備考

本関数をコールする前に、発地点(および着地点)を指定して、距離計算関数が正常に終了していることが必要です。

```
long WINAPI ACT_DIS_GetCalcDSecTime( short nToPrefCode, long lnToNodeID )
```

```
short      nToPrefCode      着地点都道府県コード
```

```
long      lnToNodeID        着地点道路点ID
```

本関数は、指定された点までの所要時間(1/10 秒単位)を求めます。

パラメータ	説明
-------	----

nToPrefCode	着地点都道府県コード。2桁の数値。
-------------	-------------------

lnToNodeID	着地点道路点ID
------------	----------

戻り値

値が取得できた場合は発地点から到達地点までの所要時間(1/10 秒単位)を返します。値が取得できなかった場合は-1を返します。

着地点を示す都道府県コードまたは道路点IDが不備である場合、指定された点に到達できない場合、着地点が計算用道路データに存在しない場合は-1を返します。

備考

本関数をコールする前に、発地点(および着地点)を指定して、距離計算関数が正常に終了していることが必要です。

```
long WINAPI ACT_DIS_GetCalcDistance( short nToPrefCode, long InToNodeID )
```

```
short      nToPrefCode    着地点都道府県コード
```

```
long      InToNodeID     着地点道路点ID
```

本関数は、指定された点までの道のり(m単位)を求めます。

パラメータ	説明
nToPrefCode	着地点都道府県コード。2桁の数値。
InToNodeID	着地点道路点ID

戻り値

値が取得できた場合は発地点から到達地点までの道のり(m単位)を返します。値が取得できなかった場合は-1を返します。

着地点を示す都道府県コードまたは道路点IDが不備である場合、指定された点に到達できない場合、着地点が計算用道路データに存在しない場合は-1を返します。

備考

本関数をコールする前に、発地点(および着地点)を指定して、距離計算関数が正常に終了していることが必要です。

```
int WINAPI ACT_DIS_GetCalcRouteXY( short nToPrefCode, long lnToNodeID,
                                   long * lpBufSize, double * lpRouteBuf,
                                   BOOL bContinue )
```

short nToPrefCode	着地点都道府県コード
long lnToNodeID	着地点道路点ID
long * lpBufSize	ルートバッファサイズ
double * lpRouteBuf	ルートバッファ
BOOL bContinue	継続フラグ

指定された点までのルート情報(XY座標の系列)を求めます。

パラメータ	説明
nToPrefCode	着地点都道府県コード。2桁の数値。
lnToNodeID	着地点道路点ID
lpBufSize	ルートバッファサイズを指定する long 値へのポインタ。関数コール時に lpRouteBuf が格納できる最大の要素数を指定します。関数リターン時には、ルートバッファにセットされたルートの要素数がセットされます。
lpRouteBuf	ルートバッファを示すポインタ。
bContinue	継続フラグ。初回コール時には FALSE(0)。継続コール時には TRUE(1)を指定します。

戻り値

- 0: 正常終了。すべてのルートデータがバッファにセットされた。
- 1: 正常終了。途中までのルートデータがバッファにセットされた。
- 2: エラー。到着地点に到達するルートが存在しない場合。

備考

本関数の実行前に、距離計算関数が正常に終了している必要があります。

本関数を最初にコールする場合 bContinue の値は、FALSE(0)を設定してください。ルートのシーケンスを格納するバッファ(lpRouteBuf)のサイズが実際のルートが辿るノード数より小さい場合は、バッファのサイズまでルートのシーケンスが格納され、戻り値=1となります。この場合、bContinue に TRUE(1)を設定して、本関数を再度コールしてください。

尚、2回目以降に本関数をコールして取得したルートデータの最初の要素は、前回コール時に取得したルートデータの最後の要素に等しくなります。

ルートバッファに格納されるデータは、ルートが辿るノードのX座標(経度)、Y座標(緯度)の系列で、それぞれ度単位、小数点以下6桁の倍精度浮動小数点数値となります。尚、ルートデータが格納されていないルートバッファの領域の値は不定となります。

尚、本関数で使用している、ルートの「要素」とは、一組のX座標とY座標を示しており、ルートバッファの構造体の要素とは異なります。

すなわち、ルートバッファのサイズは、

ルートデータの要素数 × 8【sizeof(double)】 × 2【経度と緯度】

となります。

```
int WINAPI ACT_DIS_GetCalcRouteNode( short nToPrefCode, long lnToNodeID,
                                     long * lpBufSize, double * lpRouteBuf,
                                     BOOL bContinue )
```

short nToPrefCode	着地点都道府県コード
long lnToNodeID	着地点道路点ID
long * lpBufSize	ルートバッファサイズ
double * lpRouteBuf	ルートバッファ
BOOL bContinue	継続フラグ

指定された点までのルート情報(ノード番号の系列)を求めます。

パラメータ	説明
nToPrefCode	着地点都道府県コード。2桁の数値。
lnToNodeID	着地点道路点ID
lpBufSize	ルートバッファサイズを指定する long 値へのポインタ。関数コール時に lpRouteBuf が格納できる最大のノード数を指定します。関数リターン時には、ルートバッファにセットされたノード数がセットされます。
lpRouteBuf	ルートバッファを示すポインタ。
bContinue	継続フラグ。初回コール時には FALSE(0)。継続コール時には TRUE(1)を指定します。

戻り値

- 0: 正常終了。すべてのルートデータがバッファにセットされた。
- 1: 正常終了。途中までのルートデータがバッファにセットされた。
- 2: エラー。到着地点に到達するルートが存在しない場合。

備考

本関数の実行前に、距離計算関数が正常に終了している必要があります。

本関数を最初にコールする場合 bContinue の値は、FALSE(0)を設定してください。ルートのシーケンスを格納するバッファ(lpRouteBuf)のサイズが実際のルートが辿るノード数より小さい場合は、バッファのサイズまでルートのシーケンスが格納され、戻り値=1となります。この場合、bContinue に TRUE(1)を設定して、本関数を再度コールしてください。

尚、2回目以降に本関数をコールして取得したルートデータの最初の要素は、前回コール時に取得したルートデータの最後の要素に等しくなります。

ルートバッファに格納されるデータは、ルートが辿るノード番号の系列で、14桁の倍精度浮動小数点数値(小数点以下桁数ゼロ)となります。

たとえば、「都道府県コード=13、道路点ID=512」で特定されるノードは、ルートバッファ内の値では、「13000000000512」と表現されています。

尚、ルートデータが格納されていないルートバッファの領域の値は不定となります。

XY座標の系列を返す関数の場合と異なり、本関数のルートバッファのサイズは、
ルートデータの要素数 × 8【sizeof(double)】
となります。

```
int WINAPI ACT_DIS_GetCalcRouteLink( short nToPrefCode, long lnToNodeID,
                                     long * lpBufSize, double * lpRouteBuf,
                                     BOOL bContinue )
```

short nToPrefCode	着地点都道府県コード
long lnToNodeID	着地点道路点ID
long * lpBufSize	ルートバッファサイズ
double * lpRouteBuf	ルートバッファ
BOOL bContinue	継続フラグ

指定された点までのルート情報(リンク番号の系列)を求めます。

パラメータ	説明
nToPrefCode	着地点都道府県コード。2桁の数値。
lnToNodeID	着地点道路点ID
lpBufSize	ルートバッファサイズを指定する long 値へのポインタ。関数コール時に lpRouteBuf が格納できる最大のリンク数を指定します。関数リターン時には、ルートバッファにセットされたリンク数がセットされます。
lpRouteBuf	ルートバッファを示すポインタ。
bContinue	継続フラグ。初回コール時には FALSE(0)。継続コール時には TRUE(1)を指定します。

戻り値

- 0: 正常終了。すべてのルートデータがバッファにセットされた。
- 1: 正常終了。途中までのルートデータがバッファにセットされた。
- 2: エラー。到着地点に到達するルートが存在しない場合。

備考

本関数の実行前に、距離計算関数が正常に終了している必要があります。

本関数を最初にコールする場合 bContinue の値は、FALSE(0)を設定してください。ルートのシーケンスを格納するバッファ(lpRouteBuf)のサイズが実際のルートが辿るリンク数より小さい場合は、バッファのサイズまでルートのシーケンスが格納され、戻り値=1となります。この場合、bContinue に TRUE(1)を設定して、本関数を再度コールしてください。

尚、2回目以降に本関数をコールして取得したルートデータの最初の要素は、前回コール時に取得したルートデータの最後の要素に等しくなります。

ルートバッファに格納されるデータは、ルートが辿るリンク番号の系列で、14桁の倍精度浮動小数点数値(小数点以下桁数ゼロ)となります。

たとえば、「都道府県コード=13、道路区間ID=512」で特定されるリンクは、ルートバッファ内の値では、「13000000000512」と表現されています。

尚、ルートデータが格納されていないルートバッファの領域の値は不定となります。

XY座標の系列を返す関数の場合と異なり、本関数のルートバッファのサイズは、
ルートデータの要素数 × 8【sizeof(double)】
となります。

```
void WINAPI ACT_DIS_GetRouteNameFileName( LPSTR szFileName )
```

LPSTR szFileName 文字ルート情報ファイルのフルパスを受け取る文字列へのポインタ

文字ルート情報ファイルのフルパスを取得します。

パラメータ

説明

パラメータ	説明
szFileName	文字ルート情報ファイルのフルパスを受け取る文字列へのポインタ。本文字列は MAX_PATH(260)以上の長さが必要です。

戻り値

なし

備考

本関数は、文字ルート情報ファイルのフルパスを取得します。

文字ルート情報ファイルは、ACT_DIS_SetRouteNameFileName 関数でそのフルパスを指定しない場合には、計算用道路データを保存するフォルダ内に作成され、その名称は距離計算コアライブラリが決定します。

void WINAPI ACT_DIS_SetRouteNameFileName(LPSTR szFileName)

LPSTR szFileName 文字ルート情報ファイルのフルパスを格納する文字列へのポインタ

文字ルート情報ファイルのフルパスを設定します。

パラメータ

説明

szFileName	文字ルート情報ファイルのフルパスを格納する文字列へのポインタ
------------	--------------------------------

戻り値

なし

備考

文字ルート情報ファイルは、本関数で指定しない場合は、計算用道路データを保存するフォルダ内に作成され、その名称は距離計算コアライブラリが決定します。本関数は、このフルパスを任意に変更します。

BOOL WINAPI ACT_DIS_OutputCalcRouteName(short nPrefCode, long lnNodeID, UINT unFlag)

short nToPrefCode 着地点都道府県コード
 long lnToNodeID 着地点道路点ID
 UINT unFlag 出力情報指定フラグ

指定された点までのルート情報を文字情報として出力します。

パラメータ	説明
nToPrefCode	着地点都道府県コード。2桁の数値。
lnToNodeID	着地点道路点ID
unFlag	出力情報指定フラグ。下記の値のいずれかまたはすべてを指定します。 ACTLIB_ROUTENAME_NODE(0x0001): ノード情報を出力します。 ACTLIB_ROUTENAME_LINK(0x0002): リンク情報を出力します。 ACTLIB_ROUTENAME_FARE(0x0010): 通行料金情報を出力します。

戻り値

文字ルート情報ファイルが正常に出力された場合は TRUE。文字ルート情報ファイルの出力に失敗した場合は FALSE。計算関数が正常に完了していない場合、着地点に到達するルートが存在しない場合には FALSE を返します。

備考

本関数実行前に距離計算実行関数が正常に終了していることが必要です。

通行料金出力 (unFlag = ACTLIB_ROUTENAME_FARE) は、通行料金データが設定されている計算用道路データでのみ可能です。計算用道路データに通行料金データが設定されているかどうかは、ACT_DIS_IsFareExist 関数でチェックします。

本関数は、計算の発地点から指定された着地点までの、ルート情報を次ページのフォーマットの CSV 形式のテキストファイルに出力します。

本関数は、指定された地点までのルート情報を、既存のルート情報ファイルにアペンドしますので、連続した一連のルートの情報を蓄積することができます。一方、別のルートの情報を取得する場合には、必ずファイルを削除することが必要です。

出力されたルート情報ファイルのフルパスは、ACT_DIS_GetRouteNameFileName 関数で取得することができます。

項番	タグ	名称	備考
1	SE	発着地点識別子	発着地点の場合は1。それ以外は0。
2	LINK	道路区間識別子	道路区間の場合は1。それ以外は0。
3	PREF	都道府県コード	都道府県コード。
4	CODE	道路点ID、道路区間ID	道路点ID、道路区間ID
5	NAME	交差点／道路名称	交差点名称、道路名称、または料金所名称
6	TIME	所要時間	ルート情報ファイルの一番最初の発地点からの累積所要時間(1/10秒単位)
7	DIST	道のり	ルート情報ファイルの一番最初の発地点からの累積距離(m単位)
8	FareSS	二輪・軽自動車通行料金	当該料金所で支払う料金 但し、最終レコードは、ルート情報ファイルの一番最初の発地点からの累積料金
9	FareS	普通車通行料金	
10	FareM	中型車通行料金	
11	FareL	大型車通行料金	
12	FareLL	特大車通行料金	

(注)「FareSS」から「FareLL」までのカラムは、unFlag に ACTLIB_ROUTENAME_FARE フラグをセットしたときのみ出力されます。

例

実際に出力されたルート情報ファイルは下記のようになります。尚、出力情報指定フラグには (ACTLIB_ROUTENAME_NODE | ACTLIB_ROUTENAME_LINK) が指定されています。

```
"SE","LINK","PREF","CODE","NAME","TIME","DIST"
1,0,"13",5267,"",0,0
0,1,"13",6193,"六郷橋",0,0
0,0,"13",4147,"競馬場前",1095,608
0,1,"14",102734,"大師道",0,0
0,1,"14",105778,"国道 409 号",0,0
...
0,0,"12",80758,"木更津北IC",30741,34326
0,1,"12",118331,"小櫃川大橋",0,0
0,1,"12",118328,"大曽根高架橋",0,0
0,0,"12",81528,"姉崎袖ヶ浦IC",34995,41417
1,0,"12",81541,"",38576,47385
```

BOOL WINAPI ACT_DIS_IsDetailRouteExist(void)

詳細ルート表示が可能かどうかチェックします。

パラメータ	説明
-------	----

なし	
----	--

戻り値

詳細ルート表示が可能な場合は TRUE(1)。不可能な場合は FALSE(0)。

備考

本関数は、ACT_DIS_SetWorkDir 関数で指定されている計算用道路データフォルダ内の計算用道路データに詳細ルートデータが存在するかどうかをチェックします。

詳細ルートデータを使用する関数(ACT_DIS_GetCalcRouteDetail 関数等)をコールする前に、本関数を用いて詳細ルートデータの有無をチェックしてください。

```
int WINAPI ACT_DIS_GetCalcRouteDetail( short nToPrefCode, long lnToNodeID,
                                     long * lpBufSize, double * lpRouteBuf, BOOL bContinue )
short nToPrefCode   着地点都道府県コード
long lnToNodeID     着地点道路点ID
long * lpBufSize    ルートバッファサイズ
double * lpRouteBuf ルートバッファ
BOOL bContinue      継続フラグ
```

指定された点までの詳細ルート情報(XY座標の系列)を取得します。

パラメータ	説明
nToPrefCode	着地点都道府県コード。2桁の数値。
lnToNodeID	着地点道路点ID
lpBufSize	ルートバッファサイズを指定する long 値へのポインタ。関数コール時に lpRouteBuf が格納できる最大の要素数を指定します。この値は 1024 以上でなくてはなりません。関数リターン時には、ルートバッファにセットされたルートの要素数がセットされます。
lpRouteBuf	ルートバッファを示すポインタ。ルートバッファのサイズは最低でも 1024 要素 (1024×8×2 バイト)を格納できるだけの大きさが必要です。
bContinue	継続フラグ。初回コール時には FALSE(0)。継続コール時には TRUE(1)を指定します。

戻り値

- 0: 正常終了。すべてのルートデータがバッファにセットされた。
- 1: 正常終了。途中までのルートデータがバッファにセットされた。
- 2: エラー。到着地点に到達するルートが存在しない場合。

備考

本関数の実行前に、距離計算関数が正常に終了している必要があります。

本関数は Version 3.0 以降で詳細ルートデータをサポートする形式で作成された計算用道路データでのみ正常に動作します。計算用道路データに詳細ルートデータが存在するかどうかは ACT_DIS_IsDetailRouteExist 関数でチェックします。

本関数を最初にコールする場合 bContinue の値は、FALSE(0)を設定してください。ルートのシーケンスを格納するバッファ(lpRouteBuf)のサイズが実際のルートが辿るノード数より小さい場合は、バッファのサイズまでルートのシーケンスが格納され、戻り値=1となります。この場合、bContinue に TRUE(1)を設定して、本関数を再度コールしてください。

尚、2回目以降に本関数をコールして取得したルートデータの最初の要素は、前回コール時に取得したルートデータの最後の要素に等しくなります。

ルートバッファに格納されるデータは、ルートが辿るノードのX座標(経度)、Y座標(緯度)の系列で、それぞれ度単位、小数点以下6桁の倍精度浮動小数点数値となります。尚、ルートデータが格納されていないルートバッファの領域の値は不定となります。

尚、本関数で使用している、ルートの「要素」とは、一組のX座標とY座標を示しており、ルートバッファの構造体の要素とは異なります。

すなわち、ルートバッファのサイズは、

ルートデータの要素数 × 8【sizeof(double)】 × 2【経度と緯度】

となります。

**BOOL WINAPI ACT_DIS_GetCalcRouteDetailMIF(short nToPrefCode, long lnToNodeID,
LPSTR szMifFileName)**

short nToPrefCode 着地点都道府県コード
long lnToNodeID 着地点道路点ID
LPSTR szMifFileName 出力する MIF ファイルのフルパス

指定された点までの詳細ルート情報を MIF ファイルに出力します。

パラメータ	説明
nToPrefCode	着地点都道府県コード。2桁の数値。
lnToNodeID	着地点道路点ID
szMifFileName	出力する MIF ファイルのフルパスを格納する文字列へのポインタ。

戻り値

MIF ファイルへの出力が完了したときは TRUE(1)。失敗した場合は FALSE(0)。

備考

本関数の実行前に、距離計算関数が正常に終了している必要があります。

本関数は Version 3.0 以降で詳細ルートデータをサポートする形式で作成された計算用道路データのみ正常に動作します。計算用道路データに詳細ルートデータが存在するかどうかは ACT_DIS_IsDetailRouteExist 関数でチェックします。

MIF ファイルには、詳細ルートオブジェクトが Poly Line として1レコードだけ出力されます。詳細ルートの要素数が 32,763 個を超える場合には、オブジェクトの頂点数が 32,763 個以下になるように要素が間引かれます。

MID ファイルには、nToPrefCode と lnToNodeID で着地点として指定された道路点識別子が 14 桁の文字列として出力されます。

10. リンク速度管理関数

計算に使用する道路(リンク)の速度は、計算モード終了までメモリ上に保持されている「一時的速度」と、ディスク上に保存されている「恒久的速度」の2種類の速度があり、距離計算コアライブラリは、メモリ上に保持されている「一時的速度」を用いて演算を行います。

本節のリンク速度管理関数は、この「一時的速度」「恒久的速度」のどちらに対しても取得／変更操作が行えるようになっています。

尚、計算モードを開始した直後の状態では「一時的速度」＝「恒久的速度」の関係が成立しています。

距離計算パッケージでは、各リンクについて「順方向」と「逆方向」2方向の速度を持っています。道路区間データの「接続点ID1」から「接続点ID2」への方向が「順方向」で、その逆が「逆方向」となります。

BOOL WINAPI ACT_DIS_ShowAllSpeed(HWND hWnd, UINT nFlag)

HWND hWnd 親ウィンドウのハンドル

UINT nFlag 取得対象フラグ

計算用道路データに設定されている道路速度の一覧表を表示します。

パラメータ	説明
hWnd	ダイアログの親ウィンドウのハンドル
nFlag	取得対象フラグ。速度の取得対象を指定します。下記の2つの値のいずれかまたは両方を指定することができます。 ACTLIB_SPEED_TEMP(0x0010): 一時的速度を取得します。 ACTLIB_SPEED_PARAM(0x0020): 恒久的速度を取得します。

戻り値

速度変更が正常に行われた場合は TRUE。失敗した場合は FALSE。

備考

現在計算対象として指定されている計算用道路データに設定されている道路速度の一覧表を表示します。

表示される速度は、各道路区間に設定されている道路速度の単純平均値です。道路区間の順方向速度と逆方向速度が異なる場合には、速度が大きい方の値がその道路区間の速度として平均されます。「-」が表示される場合には、集計対象の道路がないことを示しています。

一時的速度を表示するには、ACT_DIS_LoadRoadNetworks 関数がコールされて計算用道路データの一部がメモリにロードされている状態でのみ可能です。従って、ACTLIB_SPEED_TEMP フラグのみを指定する場合には、計算用道路データの一部がメモリにロードされている状態でもってなりません。一方、計算用道路データの一部がメモリにロードされている状態でも、ACTLIB_SPEED_PARAM フラグを指定することによって、恒久的速度を表示することができます。

ACTLIB_SPEED_TEMP と ACTLIB_SPEED_PARAM の両方をフラグに指定した場合は、一時的速度が優先されます。一時的速度が取得できない場合にのみ、恒久的速度が取得されます。

尚、本関数は、Version 2.0 以降で作成された計算用道路データに対してのみ正常に動作します。

```
int WINAPI ACT_DIS_ChangeBothSpeedEx( short nLinkPrefCode, long lnLinkID,
                                     int nSpeedAB, int nSpeedBA, int nFlag )
```

```
short nLinkPrefCode  対象道路の都道府県コード
long lnLinkID        対象道路の道路区間ID
int nSpeedAB         順方向の速度
int nSpeedBA         逆方向の速度
int nFlag            変更フラグ
```

指定されたリンクの速度(両方向)を設定します。

パラメータ	説明
nLinkPrefCode	対象道路の都道府県コード。2桁の数値。
lnLinkID	対象道路の道路区間ID
nSpeedAB	順方向の速度。1/10Km/h 単位で指定します。e.g.100=10km/h
nSpeedBA	逆方向の速度。1/10Km/h 単位で指定します。e.g.100=10km/h
nFlag	変更フラグ。速度変更方法および変更対象を指定します。下記の2つの値のいずれかまたは両方を指定することができます。 ACTLIB_SPEED_TEMP(0x0010): 一時的速度を変更する。 ACTLIB_SPEED_ZERORESERVE(0x0040): 速度ゼロの方向の速度は変更しない。

戻り値

速度変更が正常に行われた場合は1。失敗した場合は0。

備考

nSpeedAB、nSpeedBA に負値を指定した場合には、当該方向の速度は変更されません。たとえば、nSpeedAB=300、nSpeedBA=-1 と指定すると、この道路区間の順方向の速度は30km/hに変更されますが、逆方向の速度は変更されません。

ACTLIB_SPEED_ZERORESERVE を指定すると、変更対象の道路区間で現在速度がゼロに指定されている方向の速度は変更しません。これによって、一方通行が指定されている道路の速度の変更を変更する際に誤って一方通行情報を解除してしまうのを防ぐことができます。

一時的速度の変更は、ACT_DIS_LoadRoadNetworks 関数がコールされて計算用道路データの一部がメモリにロードされている状態でのみ可能です。従って、ACTLIB_SPEED_TEMPフラグを指定する場合には、計算用道路データの一部がメモリにロードされている状態でなくてはなりません。

ACTLIB_SPEED_TEMP フラグを指定しない場合には、一時的速度と恒久的速度の両方の速度が変更されます。

```
int WINAPI ACT_DIS_ChangeSpeedBatch( HWND hWnd,
                                     LPSTR szLinkFileName, int nFlag )
```

HWND hWnd 進捗状況を表示するダイアログの親ウィンドウはハンドル
LPSTR szLinkFileName 変更対象の道路区間識別子と変更速度を設定したテキストファイルのフルパス
int nFlag 変更フラグ

指定されたリンクの速度(両方向)を設定します。

パラメータ	説明
hWnd	進捗状況を表示するダイアログの親ウィンドウはハンドル
szLinkFileName	変更対象の道路区間識別子と変更速度を設定したテキストファイルのフルパスを格納した文字列へのポインタ。
nFlag	変更フラグ。速度変更方法および変更対象を指定します。下記の2つの値のいずれかまたは両方を指定することができます。 ACTLIB_SPEED_TEMP(0x0010): 一時的速度を変更する。 ACTLIB_SPEED_ZERORESERVE(0x0040): 速度ゼロの方向の速度は変更しない。

戻り値

速度変更が正常に行われた場合は1。失敗した場合は0。

備考

szLinkFileName で指定されるテキストファイルはカンマ区切りのCSVファイルで、下表のカラムを持つものとして。尚、ファイルの第1レコードからデータとして認識されます。

項番	フィールド名称	備考
1	都道府県コード	速度変更対象の道路区間の識別子。
2	区間ID	速度変更対象の道路区間の識別子。
3	速度A→B	順方向の速度。1/10Km/h 単位で指定します。 負値を指定するとこの方向の速度は変更しません。
4	速度B→A	逆方向の速度。1/10Km/h 単位で指定します。 負値を指定するとこの方向の速度は変更しません。

nFlag に ACTLIB_SPEED_ZERORESERVE を指定すると、変更対象の道路区間で現在速度がゼロに指定されている方向の速度は変更しません。これによって、一方通行が指定されている道路の速度の変更を変更する際に誤って一方通行情報を解除してしまうのを防ぐことができます。

一時的速度の変更は、ACT_DIS_LoadRoadNetworks 関数がコールされて計算用道路データの一部がメモリにロードされている状態でのみ可能です。従って、ACTLIB_SPEED_TEMPフラグを指定する場合には、計算用道路データの一部がメモリにロードされている状態でなくてはなりません。

ACTLIB_SPEED_TEMP フラグを指定しない場合には、一時的速度と恒久的速度の両方の速度が変更されます。

```
int WINAPI ACT_DIS_GetSpeed( short nFromPrefCode, long lnFromNodeID,
                             short nToPrefCode, long lnToNodeID, int nFlag )
```

```
short nFromPrefCode  起点の道路点の都道府県コード
long lnFromNodeID    起点の道路点の道路点ID
short nToPrefCode    終点の道路点の都道府県コード
long lnToNodeID      終点の道路点の道路点ID
int nFlag            取得対象フラグ
```

指定されたリンクの速度(片方向)を取得します。

パラメータ	説明
short nFromPrefCode	起点の道路点の都道府県コード。2桁の数値。
long lnFromNodeID	起点の道路点の道路点ID
short nToPrefCode	終点の道路点の都道府県コード。2桁の数値。
long lnToNodeID	終点の道路点の道路点ID
int nFlag	取得対象フラグ。一時的速度または恒久的速度を取得することを指定します。次の値のいずれかまたは両方を指定することができます。 ACTLIB_SPEED_TEMP(0x0010): 一時的速度を取得します。 ACTLIB_SPEED_PARAM(0x0020): 恒久的速度を取得します。

戻り値

関数が正常に終了した場合は取得した速度(1/10km/h 単位)。失敗した場合は-1。

備考

nFromPrefCode と lnFromNodeID で指定された起点と、nToPrefCode と lnToNodeID で指定された終点を結ぶ道路区間の、起点から終点方向の速度を取得します。起点と終点が道路区間の両端にない場合、関数は失敗します。

一時的速度を取得するには、ACT_DIS_LoadRoadNetworks 関数がコールされて計算用道路データの一部がメモリにロードされている状態でのみ可能です。従って、ACTLIB_SPEED_TEMP フラグのみを指定する場合には、計算用道路データの一部がメモリにロードされている状態でもってなりません。一方、計算用道路データの一部がメモリにロードされている状態でも、ACTLIB_SPEED_PARAM フラグを指定することによって、恒久的速度を取得することができます。

ACTLIB_SPEED_TEMP と ACTLIB_SPEED_PARAM の両方をフラグに指定した場合は、一時的速度の取得が優先されます。一時的速度が取得できない場合にのみ、恒久的速度が取得されます。

```
int WINAPI ACT_DIS_GetSpeedEx( short nLinkPrefCode, long lnLinkID, int nFlag,
                               int * nSpeedAB, int * nSpeedBA )
short nFromPrefCode  対象道路区間の都道府県コード
long lnFromLinkID    対象道路区間の道路点ID
int nFlag            取得対象フラグ
int * nSpeedAB       順方向速度を受け取る整数値へのポインタ
int * nSpeedBA       逆方向速度を受け取る整数値へのポインタ
```

指定されたリンクの速度(両方向)を取得します。

パラメータ	説明
nFromPrefCode	対象道路区間の都道府県コード。2桁の数値。
lnFromLinkID	対象道路区間の道路点ID
int nFlag	取得対象フラグ。一時的速度または恒久的速度を取得することを指定します。次の値のいずれかまたは両方を指定することができます。 ACTLIB_SPEED_TEMP(0x0010): 一時的速度を取得します。 ACTLIB_SPEED_PARAM(0x0020): 恒久的速度を取得します。
nSpeedAB	順方向速度を受け取る整数値へのポインタ
nSpeedBA	逆方向速度を受け取る整数値へのポインタ

戻り値

関数が正常に終了した場合は1。失敗した場合は0。

備考

一時的速度を取得するには、ACT_DIS_LoadRoadNetworks 関数がコールされて計算用道路データの一部がメモリにロードされている状態でのみ可能です。従って、ACTLIB_SPEED_TEMP フラグのみを指定する場合には、計算用道路データの一部がメモリにロードされている状態でなくてはなりません。一方、計算用道路データの一部がメモリにロードされている状態でも、ACTLIB_SPEED_PARAM フラグを指定することによって、恒久的速度を取得することができます。

ACTLIB_SPEED_TEMP と ACTLIB_SPEED_PARAM の両方をフラグに指定した場合は、一時的速度の取得が優先されます。一時的速度が取得できない場合にのみ、恒久的速度が取得されます。


```
long WINAPI ACT_DIS_CountTempSpeed( void )
```

一時的速度が設定されているリンク数を取得します

パラメータ	説明
-------	----

なし	
----	--

戻り値

一時的速度が設定されている道路区間の本数。関数が失敗した場合は-1。

備考

本関数は一時的速度が設定されている道路区間の数を取得します。

一時的速度は、ACT_DIS_LoadRoadNetworks 関数がコールされて計算用道路データの一部がメモリにロードされている状態でのみ有効な速度です。従って、計算用道路データの一部がメモリにロードされていない状態で本関数を実行すると、関数はエラー(-1)を返します。

```
int WINAPI ACT_DIS_CommitTempSpeed( void )
```

一時的速度を恒久的速度として保存します。

パラメータ	説明
-------	----

なし	
----	--

戻り値

関数が正常に終了した場合は1。関数が失敗した場合は0。

備考

本関数は一時的速度を恒久的速度としてディスクに保存します。

一時的速度は、ACT_DIS_LoadRoadNetworks 関数がコールされて計算用道路データの一部がメモリにロードされている状態でのみ有効な速度です。従って、計算用道路データの一部がメモリにロードされていない状態で本関数を実行すると、関数はエラー(-1)を返します。

```
int WINAPI ACT_DIS_ClearTempSpeed( void )
```

一時的速度変更を取り消し元の速度(恒久的速度)に戻します。

パラメータ	説明
-------	----

なし	
----	--

戻り値

関数が正常に終了した場合は1。関数が失敗した場合は0。

備考

本関数は一時的速度をすべて取り消し、恒久的速度に変更します。

一時的速度は、ACT_DIS_LoadRoadNetworks 関数がコールされて計算用道路データの一部がメモリにロードされている状態でのみ有効な速度です。従って、計算用道路データの一部がメモリにロードされていない状態で本関数を実行すると、関数はエラー(-1)を返します。

```
int WINAPI ACT_DIS_ResumeOneWayInfo( HWND hWnd )
```

HWND hWnd 親ウィンドウのハンドル

右左折禁止情報から一方通行情報を抽出し、通行不可能な方向の速度をゼロに設定します。

パラメータ

説明

hWnd 進捗状況を表示するダイアログの親ウィンドウのハンドル

戻り値

関数が正常に終了した場合は1。関数が失敗した場合は0。

備考

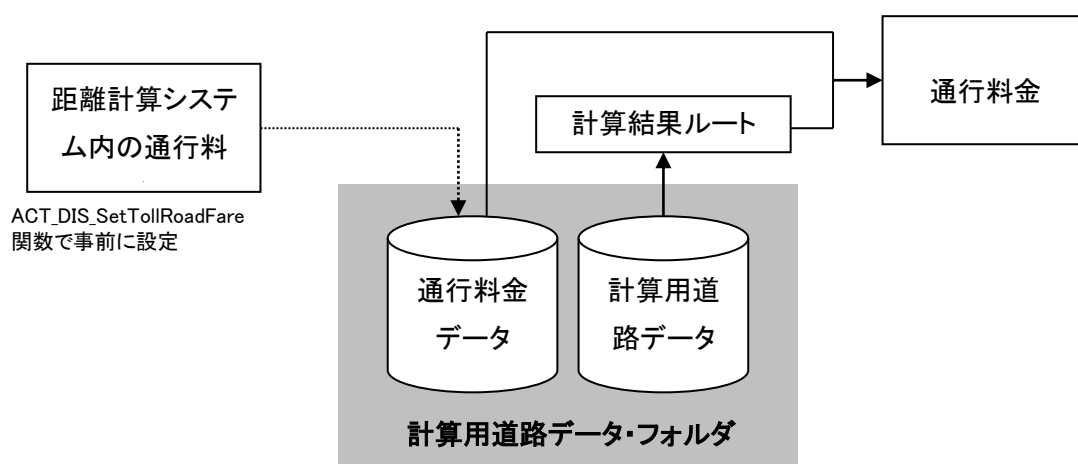
本関数は、計算用道路データを作成する時に指定された右左折禁止情報から一方通行情報を抽出し、通行不可能な方向の速度をゼロに設定します。

右左折禁止情報を指定して、計算用道路データを作成すると、一方通行情報が自動的に抽出されて当該道路区間の通行不可能な方向の速度はゼロになっていますが、道路速度変更を繰り返していくうちにこの情報が失われることがあります。本関数によって、道路速度変更によって失われた一方通行情報を再設定することができます。

11. 通行料金計算関数

通行料金計算関数は、計算結果のルートが辿るリンク情報と、計算用道路データに設定されている通行料金データから高速道路、有料道路の通行料金を算出します。このため、通行料金を算出するには、計算用道路データを作成した道路地図のバージョンに合う通行料金データが、計算用道路データに事前に設定されていなくてはなりません。

計算用道路データに通行料金データを設定するには、ACT_DIS_GetNumFareVersion 関数、ACT_DIS_GetFareVersionText 関数、および ACT_DIS_SetTollRoadFare 関数を使用します。



BOOL WINAPI ACT_DIS_SetTollRoadFare(LPSTR lpszFareVersionText)

LPSTR lpszFareVersionText 通行料金バージョンテキストを格納する文字列へのポインタ

通行料金データを計算用道路データに設定します。

パラメータ	説明
-------	----

lpszFareVersionText	通行料金バージョンテキストを格納する文字列へのポインタ
---------------------	-----------------------------

戻り値

関数が正常に終了した場合は TRUE(1)。関数が失敗した場合は FALSE(0)。

備考

本関数は、指定された通行料金バージョンテキストが示すバージョンの通行料金データを、ACT_DIS_SetWorkDir 関数で現在指定されている計算用道路データに設定します。

本関数で指定することができる通行料金バージョンテキストは、ACT_DIS_GetFareVersionText 関数で取得します。

通行料金バージョンテキストで示される通行料金データと、計算用道路データの作成に使用した道路地図のバージョンは一致しない場合には、算出される通行料金の値は不正確なものとなります。

`long WINAPI ACT_DIS_GetNumFareVersion(void)`

利用可能な通行料金データの数を取得します。

パラメータ	説明
-------	----

なし	
----	--

戻り値

利用可能な通行料金データの数。

備考

本関数は、システムに格納されている通行料金データの数を取得します。

この値は、ACT_DIS_GetFareVersionText 関数で取得することができる通行料金バージョンテキストの数と一致します。

BOOL WINAPI ACT_DIS_GetFareVersionText(short nIndex, LPSTR lpszFareVersionText)

short nIndex 取得するバージョンテキストのインデックス番号
LPSTR lpszFareVersionText 取得したバージョンテキストを格納する文字列へのポインタ

通行料金データのバージョンテキストを取得します。

パラメータ	説明
nIndex	取得するバージョンテキストのインデックス番号。 この番号には 1 から ACT_DIS_GetNumFareVersion 関数で取得した数までを指定します。
lpszFareVersionText	取得した通行料金バージョンテキストを格納する文字列へのポインタ。 本バッファサイズは 64 バイト以上必要です。

戻り値

通行料金バージョンテキストが取得できた場合は TRUE(1)。できなかった場合は FALSE(0)。

備考

本関数で取得した通行料金バージョンテキストを ACT_DIS_SetTollRoadFare 関数に渡します。
通行料金バージョンテキストで示される通行料金データと、計算用道路データの作成に使用した道路地図のバージョンは一致しない場合には、算出される通行料金の値が不正確なものとなります。

たとえば、本関数で取得できた通行料金バージョンテキストが下記の表の場合、

nIndex=1 lpszFareVersionText : “なし”
nIndex=2 lpszFareVersionText : “V1999(MSR)”
nIndex=3 lpszFareVersionText : “V2000(MSR)”

通行料金データを設定する計算用道路データの作成に使用した道路地図が三井造船システム技研株式会社の標準版道路地図 V2000 であれば、

ACT_DIS_SetTollRoadFare(“V2000(MSR)”)

と指定します。

BOOL WINAPI ACT_DIS_IsFareExist(void)

通行料金計算が可能かどうかチェックします。

パラメータ	説明
-------	----

なし	
----	--

戻り値

通行料金計算が可能な場合は TRUE(1)。不可能な場合は FALSE(0)。

備考

本関数は、ACT_DIS_SetWorkDir 関数で指定されている計算用道路データに通行料金データが設定されているかどうかをチェックします。


```

BOOL WINAPI ACT_DIS_CalcTollRoadFare( short nToPrefID, long lnToNodeID,
                                     LPTOLLROADFARE lpTollRoadFare )
short nToPrefCode           着地点都道府県コード
long lnToNodeID            着地点道路点ID
LPTOLLROADFARE lpTollRoadFare 通行料金構造体へのポインタ

```

指定された点までの通行料金を求めます。

パラメータ	説明
nToPrefCode	着地点都道府県コード。2桁の数値。
lnToNodeID	着地点道路点ID
lpTollRoadFare	通行料金構造体へのポインタ

戻り値

通行料金が計算できた場合は TRUE(1)。計算に失敗した場合は FALSE(0)。

備考

本関数をコールする前に距離計算実行関数が正常に終了していることが必要です。

また、計算用道路データに通行料金データが設定されていることが必要です。計算用道路データに通行料金データが設定されているかどうかは、ACT_DIS_IsFareExist 関数でチェックします。

本関数が正常に終了した場合には、通行料金構造体(TOLLROADFARE)の各メンバに通行料金が設定されます(下記参照)。本関数が正常に終了しなかった場合の通行料金構造体のメンバの値は不定となります。

```

typedef struct tagTOLLROADFARE{
    long    lnFareSS;           // 二輪・軽自動車料金(単位:円)
    long    lnFareS;           // 普通車料金(単位:円)
    long    lnFareM;           // 中型車料金(単位:円)
    long    lnFareL;           // 大型車料金(単位:円)
    long    lnFareLL;          // 特大車料金(単位:円)
    long    lnFareReg;         // 予備
    double  ldDistance;        // 距離(現バージョンでは未使用)
} TOLLROADFARE;
typedef TOLLROADFARE *      LPTOLLROADFARE;

```

**BOOL WINAPI ACT_DIS_CalcTollRoadFareDetail(short nToPrefID, long lnToNodeID,
LPSTR lpOutputFile, UINT unOutputFlag)**

short nToPrefCode 着地点都道府県コード
long lnToNodeID 着地点道路点ID
LPSTR lpOutputFile 出力先ファイルのフルパス
UINT unOutputFlag 出力指定フラグ(常に0)

指定された点までの詳細な通行料金データをファイルに出力します。

パラメータ	説明
nToPrefCode	着地点都道府県コード。2桁の数値。
lnToNodeID	着地点道路点ID
lpOutputFile	出力先ファイルのフルパスを格納する文字列へのポインタ
unOutputFlag	出力指定フラグ(常に0)

戻り値

通行料金データが出力できた場合は TRUE(1)。出力に失敗した場合は FALSE(0)。

備考

本関数をコールする前に距離計算実行関数が正常に終了していることが必要です。

また、計算用道路データに通行料金データが設定されていることが必要です。計算用道路データに通行料金データが設定されているかどうかは、ACT_DIS_IsFareExist 関数でチェックします。

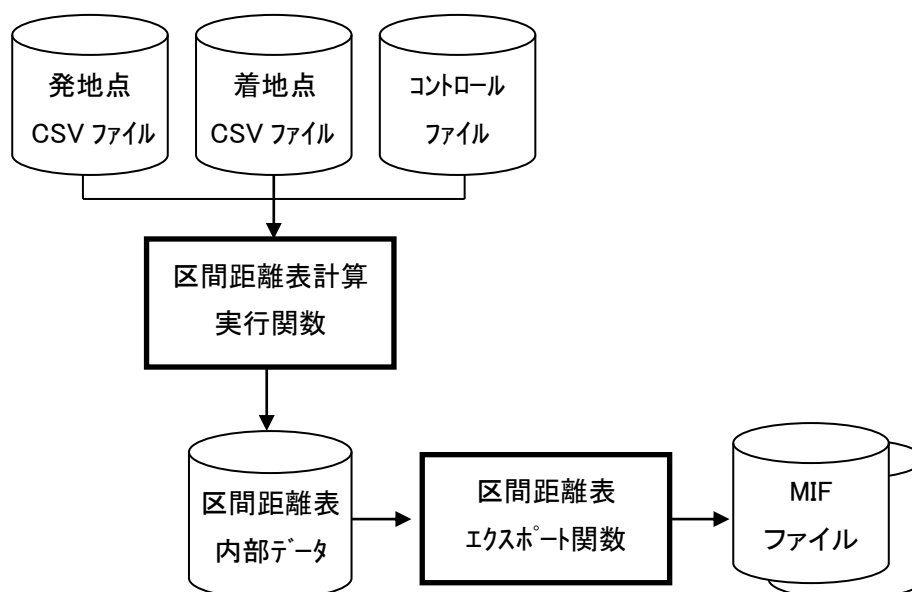
本関数は、既存のファイルに新しいデータをアペンドしますので、本関数コール前に既存のファイルを削除するようにしてください。

出力されるファイルはカンマ区切りのCSVファイルで、第1レコードはフィールド名称となっています。最終レコードは通行料金の合計値となります。各フィールドの値は下記のとおりです。

項番	フィールド名	内容
1	入道路名	入口道路の名称。
2	入口IC	入口料金所の名称。
3	出道路名	出口道路の名称。
4	出口IC	出口料金所の名称。
5	二輪・軽	各車種の通行料金(単位:円)
6	普通車	
7	中型車	
8	大型車	
9	特大車	
10	距離	入口から出口までの距離。但し、この値は、距離計算コアが計算する距離とは一致しません。

12. 区間距離表計算関数

区間距離計算は、複数の発地点と複数の着地点の間の所要時間・道のり・ルートของマトリックスを作成します。下図のように発・着地をCSVファイルに出力し、計算条件をコントロールファイルに設定した後、区間距離表関数をコールすると、計算が実行され区間距離表内部データが作成されます。計算結果は、区間距離表内部データから MIF 形式で取り出すことができます。



```
int WINAPI ACT_DIS_MTRX_ExecuteCalc(HWND hWnd, LPSTR lpszCalcInf)
```

HWND hWnd 進捗状況を表示するダイアログの親ウィンドウのハンドル

LPSTR lpszCalcInf 計算条件を設定したコントロールファイル

指定した条件で区間距離表計算を実行し、区間距離表計算結果内部データを作成します。

パラメータ	説明
hWnd	進捗状況を表示するダイアログの親ウィンドウのハンドル
lpszCalcInf	計算条件を設定したコントロールファイルのフルパスを格納する文字列へのポインタ

戻り値

計算に成功した場合は 1。失敗した場合は 0 を返します。

備考

本関数の実行前に、lpszCalcInf が示すコントロールファイルに次表の項目を設定します。

ファイル形式は、Windows の初期化ファイル(*.INI ファイル)と同様です。

[CALCOPTION]セクション

エントリ	指定例	摘要
CALCID	0	計算方法を指定します。 0:時間最短 1:距離最短
CALCTOLL	1	通行料計算を行うか否かを指定します 0:行わない 1:行う
TRF	0	交通規制情報の考慮フラグ 0:考慮しない 1:考慮する
NOHIGHTWAY	1	高速道路非使用フラグ 0:高速道路を使用する 1:高速道路を使用しない
THRESHOLD	0	到達圏計算のしきい値 0:しきい値なし、全ルート計算 >0有効値 単位:時間は分 距離はm なお TRF の時は無効です

[POINTINFO]セクション

エントリ	指定例	摘要
NUMFROMPOINT	500	発地のポイント数
FROMFILE	C:¥FROM.CSV	発地のポイントのファイルパス ^(注1)
NUMTOPOINT	1000	着地ポイント数
TOFILE	C:¥TO.CSV	発地のポイントのファイルパス ^(注1)

[OUTPUT]セクション

エントリ	指定例	摘要
FOLDER	C:¥Result	出力フォルダ
TIME	1	時間データの出力有無 0:なし、1:出力
DIST	1	距離データの出力有無
FARE	1	通行料データの出力有無
ROUTE	1	ルート情報の出力有無
OBJTYPE	1	ルートオブジェクトのタイプ ^(注2) 0:ルートなし 1:簡易ルート 2:詳細ルート
OBJPEN	Pen(3, 2, 16711680)	ルートオブジェクトのペンタイプ

(注1)発・着地点CSVファイルは、次のカラムを持ちます。第1レコードからデータとして認識されます(第1レコードにカラム名を出力しないでください)。

項番	項目名称	内容
1	ポイント識別子	地点の識別子。30バイト以内の文字列である必要があります。30バイト以上の文字列を指定した場合は、各区間表の出力時に先頭より30バイトに縮められます。ダブルクォーテーションで囲んで出力します。
2	都道府県コード	ポイントの最寄道路点の識別子。ダブルクォーテーションで囲んで出力します。
3	,道路点ID	ポイントの最寄道路点の識別子。整数値。

下記は発・着地点CSVファイルの例です。

e.g.
 "深川倉庫","13",124576
 "市ヶ谷倉庫","13",74629
 "麻布倉庫","13",20704

(注2)本関数で、ルートオブジェクトの要素数が 32,763 個を超える場合には、オブジェクトの頂点数が 32,763 個以下になるように要素が間引かれます。

例

コントロールファイルの例を以下に示します。

[CALCOPTION]

CALCID=1
 CALCTOLL=1
 TRF=0
 NOHIGHTWAY=0
 THRESHOLD=-1

[OUTPUT]

FOLDER=E:¥_Work
 TIME=1
 DIST=1
 FARE=1
 ROUTE=1
 OBJTYPE=2
 OBJPEN=Pen(3, 2, 16711680)

[POINTINFO]

NUMFROMPOINT=3
 FROMFILE=E:¥_Work¥FROM.CSV
 NUMTOPOINT=10
 TOFILE=E:¥_Work¥TO.CSV

int WINAPI ACT_DIS_MTRX_ExportCoreToMif(HWND hWnd, LPSTR lpszExportInfo)

HWND hWnd 進捗状況を表示するダイアログの親ウィンドウのハンドル

LPSTR lpszExportInfo エクスポート条件を設定したコントロールファイルのフルパス

区間距離表計算結果内部データを変換し、MIF ファイルに出力します。

パラメータ

説明

パラメータ	説明
hWnd	進捗状況を表示するダイアログの親ウィンドウのハンドル
lpszExportInf	エクスポート条件を設定したコントロールファイルのフルパスを格納する文字列へのポインタ

戻り値

エクスポートに成功した場合は 1。失敗した場合は 0 を返します。

備考

本関数の実行前に ACT_DIS_MTRX_ExecuteCalc 関数が正常に終了している必要があります。

本関数の実行前に、lpszExportInf が示すコントロールファイルに次表の項目を設定します。

ファイル形式は、Windows の初期化ファイル(*.INI ファイル)と同様です。

[FOLDER]セクション

エントリ	指定例	摘要
COREDATA	C:%Result	内部データの格納先フォルダを指定します。 ACT_MBL_MTRX_ExecuteCalc の出力先フォルダを指定します。
EXPORT	C:%Mif	Mif の出力先フォルダを指定します。

[OPTION]セクション

エントリ	指定例	摘要
NUMCOLUMN	200	カラム数の最大値 250 以下の値を指定することができます。
FROMISCOLUMN	1	発地を列として表出力するか否かを設定します。 0=着地が列 1=発地が列

[OUTPUT]セクション

エントリ	指定例	摘要
TIME	1	時間表を出力する場合は1を指定
DIST	1	距離表を出力する場合は1を指定
FARE	1	通行料金を出力する場合は1を指定
TWOPOINT	7	区間表に出力する項目を指定します。 1=時間、2=距離、4=通行料金、8=ルート (すべてを出力する場合は15を指定) (時間、距離、通行料金の場合は7を指定)
TIMEUNIT	1	時間の出力単位 1=時間:分、2=分
DISTUNIT	1	距離の出力単位 1=Km、2=m
FARECARTYPE	3	区間通行料金を出力する場合の車種を指定します。 1=二輪・軽、2=普通車、3=中型車、4=大型車、5=特大車

例

コントロールファイルの例を以下に示します。

[FOLDER]

COREDATA=E:¥_Work

EXPORT=E:¥_Work

[OPTION]

NUMCOLUMN=200

FROMISCOLUMN=1

[OUTPUT]

TIME=1

DIST=1

FARE=1

TWOPOINT=15

TIMEUNIT=1

DISTUNIT=1

FARECARTYPE=4

[FILENAME]

NUMFILE=1

TIME0001=Time

DIST0001=Dist

FARE0001=Fare

TWOPOINT=Route

```
int WINAPI ACT_DIS_MTRX_DeleteCoreResult(HWND hWnd, LPSTR lpszResultFolder);
```

HWND hWnd 親ウィンドウのハンドル

LPSTR lpszResultFolder 区間距離表計算結果内部データの出力先フォルダ

区間距離表計算結果内部データを削除します。

パラメータ	説明
hWnd	親ウィンドウのハンドル
lpszResultFolder	区間距離表計算結果内部データの出力先フォルダのフルパスを格納する文字列へのポインタ

戻り値

削除に成功した場合は 1。失敗した場合は 0 を返します。

備考

本関数は指定フォルダ内にある区間距離表計算結果内部データを削除します。

13. 2点間距離計算関数

本節では、2点間距離計算機能で使用する関数を説明します。

BOOL WINAPI ACT_DIS_CalcFromToRoute(HWND hWnd)

HWND hWnd 親ウィンドウのハンドル

2点間距離計算を実行し、計算結果ファイルを作成します。

パラメータ	説明
-------	----

hWnd	親ウィンドウのハンドル
------	-------------

戻り値

削除に成功した場合は TRUE(1)。失敗した場合は FALSE(0)を返します。

備考

本関数は DistCalc.ini の [FROMTO_SYSTEM] セクションの指定に従って2点間距離計算を実行します(下表)。なお、本関数の計算実行時の距離計算モードの設定は DistCalc.ini の [OPERATION] セクションの設定を使用します。

エントリ	摘要
MAKEROUTEOBJECT	ルートオブジェクトを作成する場合は 1。 作成しない場合は 0。
EXPORTMIF	本関数が読み込むコントロールデータ(MIF ファイル)のフルパス
IMPORTMIF	本関数が出力する計算結果データ(MIF ファイル)のフルパス

(EXPORTMIF)本関数が読み込む MID ファイルのカラム定義

項番	カラム名	内容
1	発地点コード	発地点識別子。最大 15 バイトの文字列。
2	発地点都道府県コード	発地点の最寄道路点識別子。
3	発地点道路点ID	
4	着地点コード	着地点識別子。最大 15 バイトの文字列。
5	着地点都道府県コード	発地点の最寄道路点識別子。
6	着地点道路点ID	

(IMPORTMIF)計算結果データの MID ファイルのカラム定義

項番	カラム名	内容
1	発地点コード	発地点識別子。最大 15 バイトの文字列。
2	発地点都道府県コード	発地点の最寄道路点識別子。
3	発地点道路点ID	
4	着地点コード	着地点識別子。最大 15 バイトの文字列。
5	着地点都道府県コード	発地点の最寄道路点識別子。
6	着地点道路点ID	
7	時間	発地点から着地点までの所要時間。単位: 1/10 秒。 計算に失敗した場合は-1。
8	距離	発地点から着地点までの道のり。単位: m。 計算に失敗した場合は-1。
9	通行料金	発地点から着地点までの通行料金。単位: 円。 計算に失敗した場合は-1。

14. 最適拠点選択計算関数

本節では、最適拠点選択計算で使用する関数を説明します。

BOOL WINAPI ACT_DIS_ CalcOptSite (HWND hWnd)

HWND hWnd 親ウィンドウのハンドル

最適拠点選択計算を実行し、計算結果ファイルを作成します。

パラメータ	説明
hWnd	親ウィンドウのハンドル

戻り値

削除に成功した場合は TRUE(1)。失敗した場合は FALSE(0)を返します。

備考

本関数は DistCalc.ini の[OPTSITE_SYSTEM]セクションの指定に従って最適拠点選択計算を実行します (下表)。なお、本関数の計算実行時の距離計算モードの設定は DistCalc.ini の[OPERATION]セクションの設定を使用します。

(ファイル指定部)

エントリ	摘要
EXPORTSITEMIF	本関数を読み込む拠点候補地点データ(MIF ファイル)のフルパス
EXPORTUSERMIF	本関数を読み込む納品先データ(MIF ファイル)のフルパス
IMPORTSITEMIF	本関数が出力する拠点データ(MIF ファイル)のフルパス
IMPORTUSERMIF	本関数が出力する納品先データ(MIF ファイル)のフルパス

(拠点候補データ部)

エントリ	摘要
SITWEIGHTCOL	拠点候補データの最大荷量カラム名称 最大荷量を指定しない場合は SITWEIGHTCOL=[なし]と記述
SITERANGECOL	拠点候補データの最大範囲カラム名称 最大範囲を指定しない場合は SITERANGECOL=[なし]と記述

(納品先データ部)

エントリー	摘要
USERWEIGHTCOL	納品先データの荷量カラム名称 荷量を指定しない場合は SITEWEIGHTCOL=[なし]と記述
USERSITECODECOL	納品先データの最寄拠点コード記入カラム名称 指定しない場合は USERSITECODECOL=[なし]と記述
USERTIMECOL	納品先データの所要時間記入カラム名称 指定しない場合は USERTIMECOL=[なし]と記述
USERDISTCOL	納品先データの道のり記入カラム名称 指定しない場合は USERDISTCOL=[なし]と記述
USERTONKMCOL	納品先データのトンキロ記入カラム名称 指定しない場合は USERTONKMCOL=[なし]と記述

(パラメータ部)

エントリー	摘要
SITERANGEUNIT	拠点の最大範囲の単位 1=分、2=km
NUMOPTSITE	算出する拠点の数
TONKMTYPE	トンキロの計算方法 1=荷量×時間、2=荷量×距離
MAXDEVIATION	荷量の許容乖離率。単位:%

本関数がインポートする拠点候補地点データ(MID ファイル)、納品先データ(MID ファイル)のカラムは下表のとおりです。

なお、拠点候補地点データ、納品先データの MIF ファイルには、有効なポイントデータが出力されていることが必要です。

(拠点候補地点データ)

項番	カラム名	内容
1	拠点コード	拠点候補地点の識別子。最大 15 バイトの文字列。
2	拠点都道府県コード	拠点候補地点の最寄道路点識別子。
3	拠点道路点ID	
4	固定フラグ	固定拠点を示すフラグ(1=固定)。
5	最大荷量	拠点候補地点の荷量の最大値。 指定しない場合はカラム自体が存在しないものとする。
6	最大範囲	拠点候補地点の範囲の最大値。単位:分またはkm 指定しない場合はカラム自体が存在しないものとする。

(納品先データ)

項番	カラム名	内容
1	都道府県コード	納品先の最寄道路点識別子。
2	道路点ID	
3	荷量	納品先の荷量。 指定しない場合はカラム自体が存在しないものとする。

本関数がエクスポートする拠点データ(MID ファイル)、納品先データ(MID ファイル)のカラムは下表のとおりです。

なお、拠点データ MIF ファイルの奇数行は拠点のポイント、偶数行は拠点の範囲を示すポリゴン図形が保存されます。

(拠点データ)

項番	カラム名	内容
1	拠点コード	拠点候補地点の識別子。最大 15 バイトの文字列。
2	拠点都道府県コード	拠点候補地点の最寄道路点識別子。
3	拠点道路点ID	
4	総件数	拠点到割り当てられた納品先の件数。
5	総荷量	拠点到割り当てられた納品先の荷量の合計値。 納品先の荷量を設定しない場合は、拠点到割り当てられた納品先の件数。
6	総トンキロ	拠点到割り当てられた納品先のトンキロの合計値。

(納品先データ)

項番	カラム名	内容
1	都道府県コード	納品先の最寄道路点識別子。
2	道路点ID	
3	荷量	納品先の荷量。 指定しない場合はカラム自体が存在しないものとする。
4	拠点コード	割り当て先の拠点の識別子。 指定しない場合はカラム自体が存在しないものとする。
5	時間	割り当て先の拠点からの所要時間。単位:分。 指定しない場合はカラム自体が存在しないものとする。
6	距離	割り当て先の拠点からの道のり。単位:km。 指定しない場合はカラム自体が存在しないものとする。
7	トンキロ	割り当て先の拠点からのトンキロ。 指定しない場合はカラム自体が存在しないものとする。

int WINAPI ACT_DIS_CountCombination (int nN, int nR)

int nN 組み合わせの基数
int nR 組み合わせの選択数

最適拠点選択計算の組み合わせ数を求めます。

パラメータ	説明
nN	組み合わせの基数
nR	組み合わせの選択数

戻り値

組み合わせ数。組み合わせ数が LONG_MAX 以上の場合は -1。

BOOL WINAPI ACT_DIS_VerifyCombination (HWND hWnd,

int nNumSite, int nNumResult, int nNumUser)
HWND hWnd 親ウィンドウのハンドル
int nNumSite 拠点候補地点数
int nNumResult 選択する拠点の数
int nNumUser 納品先件数

最適拠点選択計算の計算回数、パラメータを表示し、計算実行の可否を尋ねるメッセージボックスを表示します。

パラメータ	説明
HWND hWnd	親ウィンドウのハンドル
int nNumSite	拠点候補地点数
int nNumResult	選択する拠点の数
int nNumUser	納品先件数

戻り値

計算を実行する場合は TRUE(1)。

パラメータにエラーがある場合、オペレータがキャンセルボタンを押下した場合は FALSE(0)。

備考

本関数をコールする前に DistCalc.ini の[OPTSITE_SYSTEM]セクション、[OPERATION]セクションを設定する必要があります。

BOOL WINAPI ACT_DIS_VerifyCombinationEx(HWND hWnd,
int nNumSite, int nNumFixedSite, int nNumResult, int nNumUser)

HWND hWnd 親ウィンドウのハンドル
int nNumSite 拠点候補地点数
int nNumFixedSite 固定地点数
int nNumResult 選択する拠点の数
int nNumUser 納品先件数

最適拠点選択計算の計算回数、パラメータを表示し、計算実行の可否を尋ねるメッセージボックスを表示します。

パラメータ	説明
HWND hWnd	親ウィンドウのハンドル
int nNumSite	拠点候補地点数
int nNumFixedSite	固定地点数
int nNumResult	選択する拠点の数
int nNumUser	納品先件数

戻り値

計算を実行する場合は TRUE(1)。

パラメータにエラーがある場合、オペレータがキャンセルボタンを押下した場合は FALSE(0)。

備考

本関数をコールする前に DistCalc.ini の[OPTSITE_SYSTEM]セクション、[OPERATION]セクションを設定する必要があります。

15. 最適拠点立地計算関数

本節では、最適拠点立地計算で使用する関数を説明します。

BOOL WINAPI ACT_DIS_ CalcOptPnt(HWND hWnd)

HWND hWnd 親ウィンドウのハンドル

最適拠点立地計算を実行し、計算結果ファイルを作成します。

パラメータ	説明
hWnd	親ウィンドウのハンドル

戻り値

削除に成功した場合は TRUE(1)。失敗した場合は FALSE(0)を返します。

備考

本関数は DistCalc.ini の[OPTPNT_SYSTEM]セクションの指定に従って最適拠点立地計算を実行します（下表）。なお、本関数の計算実行時の距離計算モードの設定は DistCalc.ini の[OPERATION]セクションの設定を使用します。

（ファイル指定部）

エントリ	摘要
EXPORTUSERMIF	本関数が読み込む納品先データ(MIF ファイル)のフルパス
IMPORTPNTMIF	本関数が出力する最適地点データ(MIF ファイル)のフルパス
IMPORTUSERMIF	本関数が出力する納品先データ(MIF ファイル)のフルパス

（納品先データ部）

エントリ	摘要
USERWEIGHTCOL	納品先データの荷量カラム名称 荷量を指定しない場合は USERWEIGHTCOL=[なし]と記述
USERTIMECOL	納品先データの所要時間記入カラム名称 指定しない場合は USERTIMECOL=[なし]と記述
USERDISTCOL	納品先データの道のり記入カラム名称 指定しない場合は USERDISTCOL=[なし]と記述
USERTONKMCOL	納品先データのトンキロ記入カラム名称 指定しない場合は USERTONKMCOL=[なし]と記述

(パラメータ部)

エントリ	摘要
TONKMTYPE	トンキロの計算方法 1=荷量×時間、2=荷量×距離
OUTPUTALLNODES	全道路点情報出力指定 0=道路点を出力しない(デフォルト)、1=出力する

(最適地点ポイントシンボル指定部)

エントリ	摘要
NODE_KIND	シンボル種別 1=MapInfo 3.0 互換、2=TrueType、3=カスタムシンボル
NODE_SHAPE	MapInfo 3.0 互換シンボルスタイル
NODE_COLOR	シンボルの色 (MapInfo BGR 値)
NODE_SIZE	シンボルのサイズ
NODE_FONT_NAME	TrueType フォント名称
NODE_FONT_STYLE	TrueType スタイル
NODE_ANGLE	TrueType ローテーション角度
NODE_CUSTOM_NAME	カスタムシンボル名称
NODE_CUSTOM_STYLE	カスタムシンボルスタイル

(納品先ポリゴン指定部)

エントリ	摘要
PEN_WIDTH	ポリゴン境界線の幅
PEN_PATTERN	ポリゴン境界線の種類
PEN_COLOR	ポリゴン境界線の色 (MapInfo BGR 値)
BRUSH_PATTERN	ポリゴン内部パターン
BRUSH_FORECOLOR	ポリゴン内部前景色 (MapInfo BGR 値)
BRUSH_BACKCOLOR	ポリゴン内部背景色 (MapInfo BGR 値)

(注) 最適地点ポイントシンボル指定部、納品先ポリゴン指定部の各エントリの値は、MapInfo Interexchange File Format の定義に準拠しています。

本関数がインポートする納品先データ(MID ファイル)のカラムは下表のとおりです。

納品先データの MIF ファイルには、有効なポイントデータが出力されていることが必要です。

項番	カラム名	内容
1	都道府県コード	納品先の最寄道路点識別子。
2	道路点ID	
3	荷量	納品先の荷量。 指定しない場合はカラム自体が存在しないものとする。

本関数がエクスポートする最適地点データ(MID ファイル)、納品先データ(MID ファイル)のカラムは下表のとおりです。

なお、最適地点データ MIF ファイルの奇数行は最適地点のポイント、偶数行は最適地点から到達可能な納品先を示すポリゴン図形が保存されます。

(最適地点データ)

項番	カラム名	内容
1	地点都道府県コード	最適地点の道路点識別子。
2	地点道路点ID	
3	総件数	最適地点から到達可能な納品先の件数。
4	総荷量	最適地点から到達可能な納品先の荷量の合計値。 納品先の荷量を設定しない場合は、納品先の件数。
5	総トンキロ	最適地点にから到達可能な納品先のトンキロの合計値。

(納品先データ)

項番	カラム名	内容
1	都道府県コード	納品先の最寄道路点識別子。
2	道路点ID	
3	荷量	納品先の荷量。 指定しない場合はカラム自体が存在しないものとする。
4	時間	最適地点からの所要時間。単位:分。 指定しない場合はカラム自体が存在しないものとする。
5	距離	最適地点からの道のり。単位:km。 指定しない場合はカラム自体が存在しないものとする。
6	トンキロ	最適地点からのトンキロ。 指定しない場合はカラム自体が存在しないものとする。

BOOL WINAPI ACT_DIS_VerifyOptPnt (HWND hWnd,, int nNumUser)

HWND hWnd 親ウィンドウのハンドル

int nNumUser 納品先件数

最適拠点立地計算の納品先数、距離計算回数、道路点数を表示し、計算実行の可否を尋ねるメッセージボックスを表示します。

パラメータ	説明
HWND hWnd	親ウィンドウのハンドル
int nNumUser	納品先件数

戻り値

計算を実行する場合は TRUE(1)。

パラメータにエラーがある場合、オペレータがキャンセルボタンを押下した場合は FALSE(0)。

備考

本関数は最適拠点立地計算実行直前にコールします。

16. その他の関数

本節では、最短巡回順序を計算する関数、到達圏ポリゴンを算出する関数、ノードの座標を取得する関数、座標の最寄りノードを取得する関数を説明します。

最短巡回順序を計算する関数と到達圏ポリゴンを算出する関数は、計算モードが開始されている状態で実行する必要がありますが、ノードの座標を取得する関数は、計算モードの開始／終了状態に関係なく実行することができます。ただし、いずれの場合でも計算用道路データの場所(フォルダ)が事前に指定されている必要があります。

ノードの座標を取得する関数は、指定されたノードが計算用道路データに存在するかどうかを判断する用途にも利用できます。

```

long WINAPI ACT_DIS_CalcOptDSecTime( HWND hWnd, long lnCounter,
                                     long * lpRecNo, short * lpPrefID, long * lpNodeID,
                                     short * lpNoHighWay )

```

HWND	hWnd	親ウィンドウのハンドル
long	lnCounter	地点数
long *	lpRecNo	レコード番号バッファへのポインタ
short *	lpPrefID	都道府県コードバッファへのポインタ
long *	lpNodeID	道路点IDバッファへのポインタ
short *	lpNoHighWay	高速道路未使用フラグバッファへのポインタ

複数地点を巡回する最短時間経路を求めます。

パラメータ	説明
hWnd	親ウィンドウのハンドル
lnCounter	地点数
lpRecNo	レコード番号バッファへのポインタ。1から始まるレコード番号を指定します。関数リターン時には、最適な順番でレコード番号が格納されます。
lpPrefID	都道府県コードバッファへのポインタ
lpNodeID	道路点IDバッファへのポインタ
lpNoHighWay	高速道路未使用フラグバッファへのポインタ。高速道路を使用する区間には0(ゼロ)を、使用しない区間には1を格納します。

戻り値

関数が正常に終了した場合は1。失敗した場合は0。

備考

指定された地点の系列を、最短時間経路を用いて巡回時間が最短になるような順番に変更します。

巡回する地点の情報は、lpRecNo, lpPrefID, lpNodeID, lpNoHighWay に指定します。計算終了後にはこれらの系列は最適な順番に並べ替えられます。

本関数は、ACT_DIS_LoadRoadNetworks 関数がコールされて、計算用道路データの一部がメモリにロードされている状態で実行します。

```
long WINAPI ACT_DIS_CalcOptDSecTimeEx ( HWND hWnd, long InCounter,
                                         long * lpRecNo, short * lpPrefID, long * lpNodeID,
                                         short * lpNoHighWay, UINT unFlag )
```

HWND	hWnd	親ウィンドウのハンドル
long	InCounter	地点数
long *	lpRecNo	レコード番号バッファへのポインタ
short *	lpPrefID	都道府県コードバッファへのポインタ
long *	lpNodeID	道路点IDバッファへのポインタ
short *	lpNoHighWay	高速道路未使用フラグバッファへのポインタ
UINT	unFlag	発着地固定フラグ

複数地点を巡回する最短時間経路を求めます。

パラメータ	説明
hWnd	親ウィンドウのハンドル
InCounter	地点数
lpRecNo	レコード番号バッファへのポインタ。1から始まるレコード番号を指定します。関数リターン時には、最適な順番でレコード番号が格納されます。
lpPrefID	都道府県コードバッファへのポインタ
lpNodeID	道路点IDバッファへのポインタ
lpNoHighWay	高速道路未使用フラグバッファへのポインタ。高速道路を使用する区間には0(ゼロ)を、使用しない区間には1を格納します。
unFlag	下記の値の1つまたは複数指定します。 ANLB_OPT_FIXNONE(0x0000) : 発着地点を固定しない ANLB_OPT_FIXFIRST(0x0100) : 発地点(最初の点)を固定する ANLB_OPT_FIXLAST(0x0001) : 着地点(最後の点)を固定する

戻り値

関数が正常に終了した場合は1。失敗した場合は0。

備考

指定された地点の系列を、最短時間経路を用いて巡回時間が最短になるような順番に変更します。

巡回する地点の情報は、lpRecNo, lpPrefID, lpNodeID, lpNoHighWay に指定します。計算終了後にはこれらの系列は最適な順番に並べ替えられます。

本関数は、ACT_DIS_LoadRoadNetworks 関数がコールされて、計算用道路データの一部がメモリにロードされている状態で実行します。

```
long WINAPI ACT_DIS_CalcOptDistance( HWND hWnd, long lnCounter,
                                     long * lpRecNo, short * lpPrefID, long * lpNodeID,
                                     short * lpNoHighWay )
```

HWND	hWnd	親ウィンドウのハンドル
long	lnCounter	地点数
long *	lpRecNo	レコード番号バッファへのポインタ
short *	lpPrefID	都道府県コードバッファへのポインタ
long *	lpNodeID	道路点IDバッファへのポインタ
short *	lpNoHighWay	高速道路未使用フラグバッファへのポインタ

複数地点を巡回する最短距離経路を求めます。

パラメータ	説明
hWnd	親ウィンドウのハンドル
lnCounter	地点数
lpRecNo	レコード番号バッファへのポインタ。1から始まるレコード番号を指定します。関数リターン時には、最適な順番でレコード番号が格納されます。
lpPrefID	都道府県コードバッファへのポインタ
lpNodeID	道路点IDバッファへのポインタ
lpNoHeighWay	高速道路未使用フラグバッファへのポインタ。高速道路を使用する区間には0(ゼロ)を、使用しない区間には1を格納します。

戻り値

関数が正常に終了した場合は1。失敗した場合は0。

備考

指定された地点の系列を、最短距離経路を用いて巡回距離が最短になるような順番に変更します。

巡回する地点の情報は、lpRecNo, lpPrefID, lpNodeID, lpNoHighWay に指定します。計算終了後にはこれらの系列は最適な順番に並べ替えられます。

本関数は、ACT_DIS_LoadRoadNetworks 関数がコールされて、計算用道路データの一部がメモリにロードされている状態で実行します。


```
long WINAPI ACT_DIS_CalcOptDistanceEx ( HWND hWnd, long InCounter,
                                       long * lpRecNo, short * lpPrefID, long * lpNodeID,
                                       short * lpNoHighWay,
```

```
UINT unFlag )
```

HWND	hWnd	親ウィンドウのハンドル
long	InCounter	地点数
long *	lpRecNo	レコード番号バッファへのポインタ
short *	lpPrefID	都道府県コードバッファへのポインタ
long *	lpNodeID	道路点IDバッファへのポインタ
short *	lpNoHighWay	高速道路未使用フラグバッファへのポインタ
UINT	unFlag	発着地固定フラグ

複数地点を巡回する最短距離経路を求めます。

パラメータ	説明
hWnd	親ウィンドウのハンドル
InCounter	地点数
lpRecNo	レコード番号バッファへのポインタ。1から始まるレコード番号を指定します。関数リターン時には、最適な順番でレコード番号が格納されます。
lpPrefID	都道府県コードバッファへのポインタ
lpNodeID	道路点IDバッファへのポインタ
lpNoHeighWay	高速道路未使用フラグバッファへのポインタ。高速道路を使用する区間には0(ゼロ)を、使用しない区間には1を格納します。
unFlag	下記の値の1つまたは複数を指定します。 ANLB_OPT_FIXNONE(0x0000) : 発着地点を固定しない ANLB_OPT_FIXFIRST(0x0100) : 発地点(最初の点)を固定する ANLB_OPT_FIXLAST(0x0001) : 着地点(最後の点)を固定する

戻り値

関数が正常に終了した場合は1。失敗した場合は0。

備考

指定された地点の系列を、最短距離経路を用いて巡回距離が最短になるような順番に変更します。

巡回する地点の情報は、lpRecNo, lpPrefID, lpNodeID, lpNoHighWay に指定します。計算終了後にはこれらの系列は最適な順番に並べ替えられます。

本関数は、ACT_DIS_LoadRoadNetworks 関数がコールされて、計算用道路データの一部がメモリにロードされている状態で実行します。

```

long WINAPI ACT_DIS_MakePolygon(short nStartPrefCode, long lnStartNodeID,
                                long lnThreshold, long bTRF,
                                long bDistanceKey, long bDistanceRange,
                                long * lnNumBuffer, LONLAT * lpPolygonBuffer )
short      nStartPrefCode      発地点の都道府県コード
long       lnStartNodeID       発地点の道路点ID
long       lnThreshold          到達時間または距離
long       bTRF                 TRF1 型指定フラグ
long       bDistanceKey        最短距離計算指定フラグ
long       bDistanceRange      到達距離指定フラグ
long       * lnNumBuffer        XY座標を格納するバッファサイズを示す値へのポインタ
LONLAT     * lpPolygonBuffer    XY座標を格納するバッファへのポインタ

```

指定された到達時間／距離の外縁をなす簡易型(凸型)ポリゴンを作成します。

パラメータ	説明
nStartPrefCode	発地点(中心点)の都道府県コード。2桁の数値。
lnStartNodeID	発地点(中心点)の道路点ID
lnThreshold	到達時間(1/10 秒単位)または距離(m単位)
bTRF	TRF1 型指定フラグ。TRF1 型の計算を行っている場合には1。ACT1 型の計算を行っている場合には0。
bDistanceKey	最短距離計算指定フラグ。距離を最短とする計算を行っている場合には1。時間を最短とする計算を行っている場合には0。
bDistanceRange	到達距離指定フラグ。lnThreshold を距離で指定した場合には1。時間で指定した場合には0。
lnNumBuffer	XY座標を格納するバッファサイズを示す値へのポインタ。関数コール時には格納可能なXY座標の数を設定します。関数リターン時には、実際に格納されたXY座標の数が返されます。尚、ここでの「XY座標の数」はX座標とY座標の一組で1つと数えます。
lpPolygonBuffer	XY座標を格納するバッファへのポインタ。関数リターン時にはXY座標(度単位)が返されます。

戻り値

関数が正常に終了した場合には1。失敗した場合には0。

備考

本関数を実行する前に、本コアライブラリの計算実行関数が正常に終了している必要があります。

【Version 8.0 機能変更関数】

long WINAPI ACT_DIS_MakeSuperPolygon(short nStartPrefCode, long lnStartNodeID,		long lnThreshold, long bTRF,
		long bDistanceKey, long bDistanceRange,
		long * lnNumBuffer, LONLAT * lpPolygonBuffer,
		long lnLevel)
short	nStartPrefCode	発地点の都道府県コード
long	lnStartNodeID	発地点の道路点ID
long	lnThreshold	到達時間または距離
long	bTRF	TRF1 型指定フラグ
long	bDistanceKey	最短距離計算指定フラグ
long	bDistanceRange	到達距離指定フラグ
long	* lnNumBuffer	XY座標を格納するバッファサイズを示す値へのポインタ
LONLAT	* lpPolygonBuffer	XY座標を格納するバッファへのポインタ
long	lnLevel	ポリゴンレベル

指定された到達時間／距離の外縁をなす標準型(凹凸型)ポリゴンを作成します。

パラメータ	説明
nStartPrefCode	発地点(中心点)の都道府県コード。2桁の数値。
lnStartNodeID	発地点(中心点)の道路点ID
lnThreshold	到達時間(1/10 秒単位)または距離(m単位)
bTRF	TRF1 型指定フラグ。TRF1 型の計算を行っている場合には1。ACT1 型の計算を行っている場合には0。
bDistanceKey	最短距離計算指定フラグ。距離を最短とする計算を行っている場合には1。時間を最短とする計算を行っている場合には0。
bDistanceRange	到達距離指定フラグ。lnThreshold を距離で指定した場合には1。時間で指定した場合には0。
lnNumBuffer	XY座標を格納するバッファサイズを示す値へのポインタ。関数コール時には格納可能なXY座標の数を設定します。関数リターン時には、実際に格納されたXY座標の数が返されます。尚、ここでの「XY座標の数」はX座標とY座標の一組で1つと数えます。
lpPolygonBuffer	XY座標を格納するバッファへのポインタ。関数リターン時にはXY座標(度単位)が返されます。
lnLevel	ポリゴンレベル。0 から 20 の値を指定します。 端点拡張モードを使用する場合は 0x100(256)を加えます。

戻り値

関数が正常に終了した場合には1。失敗した場合には0。

備考

本関数を実行する前に、本コアライブラリの計算実行関数が正常に終了している必要があります。

一般的にはポリゴンレベルの数値が大きい程、作成されるポリゴンの複雑さが増大し、ポリゴン作成に要する計算時間も長くなります。ただし、描画可能なポリゴンの複雑度合いは、描画対象地域の道路ネットワークの密度にも関連しますので、ポリゴンレベルを大きくしても描画されるポリゴンの複雑度合いが増さない場合があります。また、ポリゴンレベルが小さい方が複雑なポリゴンが作成される場合もあります。

端点拡張モードを使用する場合は、ポリゴンレベルに 0x100(256)を加えます。

端点拡張モードはデフォルトでは、到達圏内のノード数が 100 万ノードを超えると自動的にオフになります。この閾値を変更する場合は、DistCalc.ini、[CALCRANGE]セクション、EXPANDTERMINALLIMIT エントリに閾値を指定してください。

Version 8.0 では、端点拡張モードがオンのとき、同時に中間点補完が行われます。

```

long WINAPI ACT_DIS_MakeHyperPolygon( short nStartPrefCode, long lnStartNodeID,
                                     long lnThreshold, long bTRF,
                                     long bDistanceKey, long bDistanceRange,
                                     long * lnNumBuffer, LONLAT * lpPolygonBuffer,
                                     long lnLevel )
short      nStartPrefCode      発地点の都道府県コード
long       lnStartNodeID      発地点の道路点ID
long       lnThreshold         到達時間または距離
long       bTRF               TRF1 型指定フラグ
long       bDistanceKey       最短距離計算指定フラグ
long       bDistanceRange     到達距離指定フラグ
long       * lnNumBuffer      XY座標を格納するバッファサイズを示す値へのポインタ
LONLAT     * lpPolygonBuffer  XY座標を格納するバッファへのポインタ
long       lnLevel           ポリゴンレベル(常に 0)

```

指定された到達時間／距離の外縁をなす詳細型(凹凸型)ポリゴンを作成します。

パラメータ	説明
nStartPrefCode	発地点(中心点)の都道府県コード。2桁の数値。
lnStartNodeID	発地点(中心点)の道路点ID
lnThreshold	到達時間(1/10 秒単位)または距離(m単位)
bTRF	TRF1 型指定フラグ。TRF1 型の計算を行っている場合には1。ACT1 型の計算を行っている場合には0。
bDistanceKey	最短距離計算指定フラグ。距離を最短とする計算を行っている場合には1。時間を最短とする計算を行っている場合には0。
bDistanceRange	到達距離指定フラグ。lnThreshold を距離で指定した場合には1。時間で指定した場合には0。
lnNumBuffer	XY座標を格納するバッファサイズを示す値へのポインタ。関数コール時には格納可能なXY座標の数を設定します。関数リターン時には、実際に格納されたXY座標の数が返されます。尚、ここでの「XY座標の数」はX座標とY座標の一組で1つと数えます。
lpPolygonBuffer	XY座標を格納するバッファへのポインタ。関数リターン時にはXY座標(度単位)が返されます。
lnLevel	未使用。常に0を指定します。

戻り値

関数が正常に終了した場合には1。失敗した場合には0。

備考

本関数を実行する前に、本コアライブラリの計算実行関数が正常に終了している必要があります。

本関数は詳細なポリゴン図形を作成しますので、指定する到達範囲が大きい場合は、多くのメモリと長い時間を要しますので注意してください。

double WINAPI ACT_DIS_GetNodeX(short nPrefCode, long lnNodeID)

short nPrefCode 道路点の都道府県コード

long lnNodeID 道路点の道路点ID

指定されたノードのX座標を取得します。

パラメータ	説明
-------	----

nPrefCode	道路点の都道府県コード。2桁の数値。
-----------	--------------------

lnNodeID	道路点の道路点ID
----------	-----------

戻り値

関数が正常に終了した場合は、X座標(度単位)。失敗した場合は-1。

備考

指定された道路点のX座標を取得します。

double WINAPI ACT_DIS_GetNodeY(short nPrefCode, long lnNodeID)

short nPrefCode 道路点の都道府県コード

long lnNodeID 道路点の道路点ID

指定されたノードのY座標を取得します。

パラメータ	説明
-------	----

nPrefCode	道路点の都道府県コード。2桁の数値。
-----------	--------------------

lnNodeID	道路点の道路点ID
----------	-----------

戻り値

関数が正常に終了した場合は、Y座標(度単位)。失敗した場合は-1。

備考

指定された道路点のY座標を取得します。

```

long WINAPI ACT_DIS_GetNearestNode( short * lpPrefCode, long * lpNodeID,
    double ldX, double ldY, double ldRange, long bFlag )
short * lpPrefCode    取得した道路点の都道府県コードを格納するバッファへのポインタ
long * lpNodeID       取得した道路点の道路点IDを格納するバッファへのポインタ
double ldX            X座標(経度)
double ldY            Y座標(緯度)
double ldRange        探索レンジ(度)
long bFlag            高速道路ノード探索拒否フラグ

```

指定されたXY座標から探索レンジ以内で最も近い道路点を取得します。

パラメータ	説明
lpPrefCode	取得した道路点の都道府県コードを格納するバッファへのポインタ
lpNodeID	取得した道路点の道路点IDを格納するバッファへのポインタ
ldX	X座標(経度)。単位:度。
ldY	Y座標(緯度)。単位:度。
ldRange	探索レンジ。探索する範囲を指定します。単位:度。
bFlag	高速道路ノード探索拒否フラグ。高速道路上のノードを探索除外する場合には1を指定します。0を指定するとすべてのノードが探索対象となります。

戻り値

道路点が見つかった場合は1。見つからなかった場合は0(ゼロ)。

備考

指定されたXY座標から、ldRange で指定された範囲以内で、最も近い道路点の都道府県コードと道路点IDを求めます。

ldRange は、度単位の小数値を指定します。たとえば ldRagne=0.01 と指定した場合、X座標の左右 0.01 度の範囲、Y座標の上下 0.01 度の範囲でノードの探索が行われます。

long WINAPI ACT_DIS_IsHighwayNode(short nPrefCode, long lnNodeID)

short nPrefCode 道路点の都道府県コード

long lnNodeID 道路点の道路点ID

指定されたノードが高速道路上のノードかどうかを判定します。

パラメータ	説明
nPrefCode	道路点の都道府県コード。2桁の数値。
lnNodeID	道路点の道路点ID

戻り値

指定したノードが高速道路上にある場合は1。高速道路上以外にある場合は0を返します。

備考

高速道路上のノードとは、当該ノードに接続しているすべての道路が高速道路であるノードを指します。すなわち、高速道路と一般道路の接続地点の交差点は、高速道路上以外のノードとなります。

void WINAPI ACT_DIS_GetDistCalcIni(LPSTR szFileName)

LPSTR szFileName 初期設定ファイルのフルパスを受け取る文字列へのポインタ

初期設定ファイル(DistCalc.ini)のフルパスを取得します。

パラメータ

説明

szFileName	初期設定ファイルのフルパスを受け取る文字列へのポインタ。 MAX_PATH+1 以上の長さが必要です。
------------	--

戻り値

なし。

備考

距離計算パッケージ Version 4.0 以降では、初期設定ファイルは、カレントユーザのローカル・アプリケーション・データフォルダの「¥Advanced Core Technologies, Inc.¥DistCalc」フォルダに保存されています。なお、インストールフォルダ下の同名のファイルは新しいユーザ用のテンプレートとして使用されますので、こちらの設定を変更しても効果のない場合があります。

```
int WINAPI ACT_DIS_AboutDistCalc( HWND hWnd, LPSTR szAddString )
```

HWND hWnd 親ウィンドウのハンドル

LPSTR szAddString ダイアログに表示する追加文字列へのポインタ

距離計算パッケージのバージョン情報を表示します。

パラメータ	説明
hWnd	親ウィンドウのハンドル
szAddString	ダイアログに表示する追加文字列 (Zero Terminated String) へのポインタ。

戻り値

関数が正常に終了した場合は 1。失敗した場合は 0。

備考

距離計算パッケージのバージョンおよびモジュールのバージョンを表示します。

【Version 8.0 機能変更関数】

BOOL WINAPI ACT_DIS_CalcMultiArea(HWND hWnd)

HWND hWnd 親ウィンドウのハンドル

複数到達圏一括計算を実行し、計算結果ファイルを作成します。

パラメータ 説明

hWnd 親ウィンドウのハンドル

戻り値

削除に成功した場合は TRUE(1)。失敗した場合は FALSE(0)を返します。

備考

本関数は DistCalc.ini の[MULTIAREA_SYSTEM]セクションの指定に従って複数到達圏一括計算を実行します(下表)。なお、本関数の計算実行時の距離計算モードの設定は DistCalc.ini の[OPERATION]セクションの設定を使用します。

(ファイル指定部)

エントリ	摘要
EXPORTCENTERMIF	本関数が読み込むセンタデータ(MIF ファイル)のフルパス
IMPORTCENTERMIF	本関数が出力する計算結果データ(MIF ファイル)のフルパス

(拠点候補データ部)

エントリ	摘要
AREARANGECOL	センタデータのレンジカラム名称 レンジを指定しない場合は AREARANGECOL=[なし]と記述

(パラメータ部)

エントリ	摘要
DEFAULTRANGE	AREARANGECOL=[なし]の場合に使用するレンジ。
AREARANGEUNIT	拠点の最大範囲の単位(1=分、2=km)
REVERSE	流入圏計算指定フラグ(0=到達圏計算、1=流入圏計算)
EXPANDTERMINAL	端点拡張指定フラグ(256=端点拡張 ^(注) を行う、0=行わない)
POLYGONLEVEL	ポリゴン凹凸レベル(0~20)
POLYGONCOLOR	ポリゴン色指定(1=自動、2=センタオブジェクトの色)

(注)Version 8.0 では、端点拡張フラグがオンのとき、同時に中間点補完が行われます。

本関数がインポートするセンタデータ(MID ファイル)のカラム構成は下表のとおりです。センタデータには、ポイントデータ(MIF ファイル)が必要が必要です。

項番	カラム名	内容
1	センタID	センタの識別子。
2	拠点都道府県コード	センタの最寄道路点識別子。
3	拠点道路点ID	
4	最大レンジ	最大レンジ。 AREARANGECOL=[なし]の場合は不要

本関数がエクスポートする計算結果データ(MID ファイル)のカラムは下表のとおりです。

MIF ファイルには到達圏／流入圏を示すポリゴン図形が保存されます。

項番	カラム名	内容
1	センタID	センタの識別子。
2	拠点都道府県コード	センタの最寄道路点識別子。
3	拠点道路点ID	
4	最大レンジ	計算に使用した最大レンジ。

17. 定数／構造体一覧

(1) 定数

リンク速度設定・取得関数、文字ルート情報ファイル出力関数のフラグには下記の値を設定します。

項番	定数	値(16進)
1	ACTLIB_ROUTENAME_NODE	0x0001
2	ACTLIB_ROUTENAME_LINK	0x0002
3	ACTLIB_ROUTENAME_MORELINK	0x0004
4	ACTLIB_ROUTENAME_FARE	0x0010
5	ACTLIB_SPEED_TEMP	0x0010
6	ACTLIB_SPEED_PARAM	0x0020
7	ACTLIB_SPEED_ZERORESERVE	0x0040

最短順序計算(最短ルート計算)関数では、発着地点の固定の有無を示す下記の値を使用します。

項番	定数	値(16進)
1	ANLB_OPT_FIXNONE	0x0000
2	ANLB_OPT_FIXFIRST	0x0100
3	ANLB_OPT_FIXLAST	0x0001

(2) 構造体

ポリゴン作成関数のXY座標格納用構造体の宣言は下記のとおりです。尚、C言語系以外の言語で上位アプリケーションを作成する場合には、倍精度浮動小数点へのポインタ(double *)でも代用できます。

```
typedef struct LONLAT
{
    double ldLon;           // 経度(X座標)
    double ldLat;          // 緯度(Y座標)
} LONLAT;
```

通行料金取得関数で使用する通行料金構造体の宣言は下記のとおりです。

```
typedef struct tagTOLLROADFARE{
    long    lnFareSS;       // 二輪・軽自動車料金(単位:円)
    long    lnFareS;       // 普通車料金(単位:円)
    long    lnFareM;       // 中型車料金(単位:円)
    long    lnFareL;       // 大型車料金(単位:円)
    long    lnFareLL;      // 特大車料金(単位:円)
    long    lnFareReg;     // 予備
    double  ldDistance;    // 距離(現バージョンでは未使用)
} TOLLROADFARE;
typedef TOLLROADFARE *      LPTOLLROADFARE;
```

以上