

LogisticaTRUCKServer- II (SQLServer 版)距離計算 DB・アプリケーションサーバ・API・クライアント構成

LogisticaTRUCKServer- II (SQLServer 版)距離計算 DB・アプリケーションサーバ

Microsoft SQLServer 上に距離計算 DB サーバを構築します

MicrosoftSQLServer2012/2008R2/2008 をベースに、距離計算 DB サーバを構築します。距離計算 DB サーバは、住所基本DB(JIS 市区町村一般郵便番号・個別事業所郵便番号)・PowerAtlas 全国デジタル道路地図・街区位置 DB・距離計算・位置検索 DB より構成されます。距離計算アプリケーションサーバは、**2014年版全国デジタル道路地図**(住友電工システムソリューション(株)製 PowerAtlas 使用)をベースに JIS 市区町村コード・郵便番号・住所文字列地点間の距離を計算します。

住所データの自動更新

住所データは出荷時点での最新の市区町村・郵便番号情報が導入されます。その後の更新データは距離計算サーバ側で自動で更新します。**導入 2 年目以降継続利用は有償となります。**住所文字列の検索位置精度は**番地・号数レベル**です。

LogisticaTRUCKServer- II (SQLServer 版)距離計算クライアントアプリケーション

クライアント PC 側に、クライアントアプリケーションとして、**クライアント基本**を導入することにより、クライアント/サーバ形式で距離計算機能が利用できます。

LogisticaTRUCKServer- II (SQLServer 版)距離計算サーバ API

Micorsoft の VisualStudio で作成したアプリケーションについては、距離計算.NET DLL T-SQL 距離計算スタアドプロシージャ、Active-X コントロール/DLL を利用して、距離計算機能を組み込みます。それ以外の言語で作成するアプリケーションには、ソケット通信を利用して、距離計算機能を組み込みます。

距離計算ソケット通信により、Windows UNIX Linux のあらゆる OS または、OS 上のあらゆる言語で距離計算機能が実装できます。

LogisticaTRUCKServer- II (SQLServer 版)距離計算サーバ API 一覧

| API 分類 I | API 分類 II | 利用プログラム言語 | 参照 Doc / 参照 PDF |
|------------|------------------|-------------|--|
| .NET DLL | WindowsForm | VisualBasic | LogisticaTRUCKServer- II (SQLServer 版) 距離計算サーバ API_NETDLLWinForm |
| | | C# | |
| | WebForms ASP.NET | VisualBasic | LogisticaTRUCKServer- II (SQLServer 版) 距離計算サーバ API_NETDLLASPNET |
| | | C# | |
| スタアドプロシージャ | | T-SQL | T-SQL 距離計算スタアドプロシージャ |
| ソケット通信 | | Java | LogisticaTRUCKServer- II (SQLServer 版) 距離計算サーバ API_Socket |
| | | Perl | |
| | | C# | |
| | | VisualBasic | |
| | | PHP | |
| | | COBOL ほかの言語 | |
| Active-X | Active-X コントロール | VB6 | LogisticaTRUCKServer- II (SQLServer 版) 距離計算サーバ API_Active-XControl |
| | | Access200X | |
| | Active-X DLL | VB6 | LogisticaTRUCKServer- II (SQLServer 版) 距離計算サーバ API_Active-XDLL |
| | | VBScript | |
| | | ASP | |
| | | | |

LogisticaTRUCKServer- II (SQLServer 版)距離計算クライアントアプリケーション一覧

クライアントアプリケーションとして以下を提供します。

| クライアントアプリケーション名 | 機能 | 参照ページ |
|--|----------------------|-------|
| LogisticaTRUCKServer- II (SQLServer 版)距離計算クライアント基本 | 距離計算の基本 4 パターンから距離計算 | |

LogisticaTRUCKServer- II (SQLServer 版)距離計算サーバの動作環境

| | |
|--------|--|
| 必須案件 | 準備するソフトウェア製品 |
| サーバ OS | Microsoft WindowsServer2012R2/2012/2008R2/2008 |
| DB サーバ | MicrosoftSQLServer2014/2012/2008R2/2008 MicrosoftSQLServer2014 は弊社でも販売します。価格などはお問い合わせください。 動作確認用に SQLServer2008/Express を同梱 |

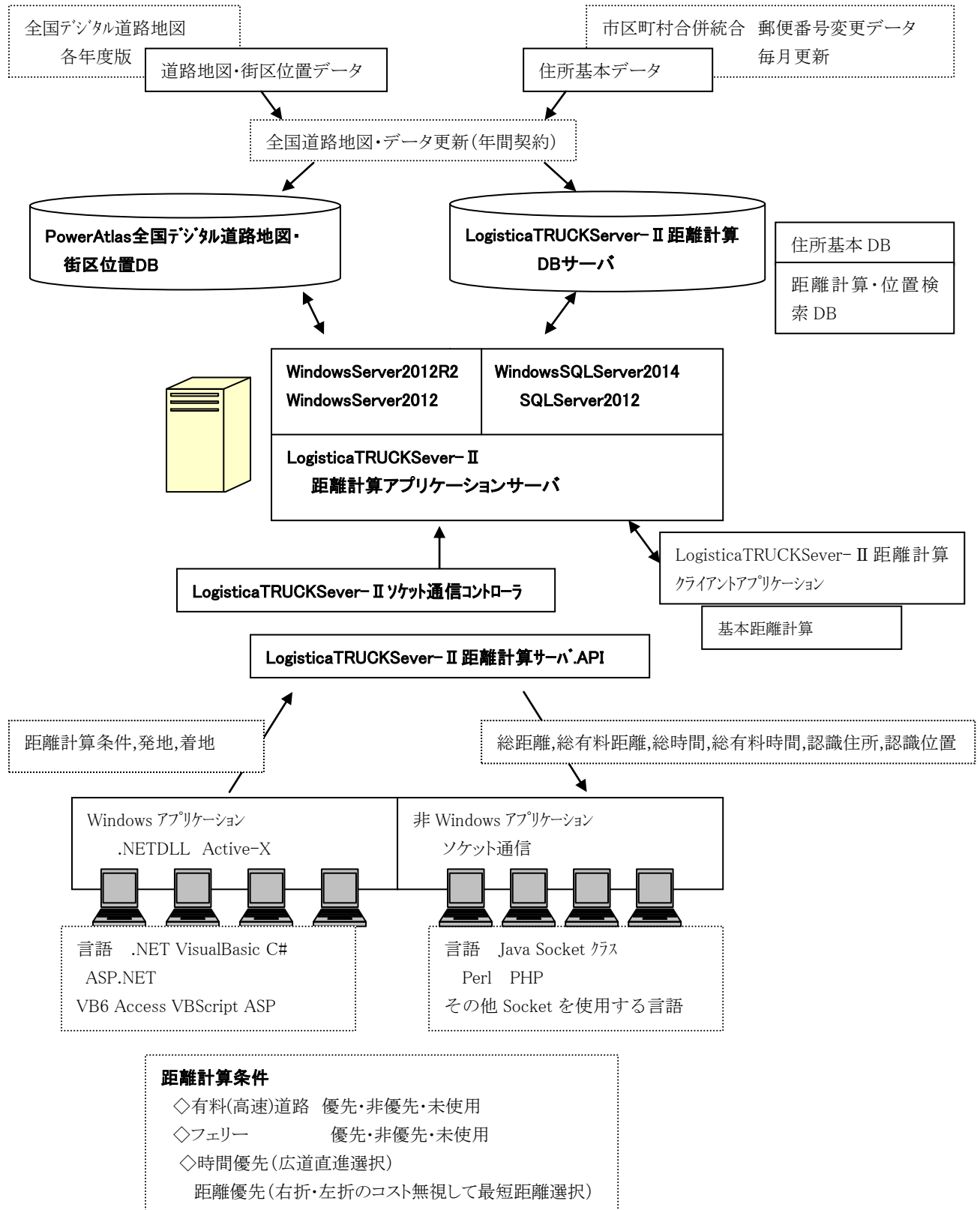
そのほかの動作環境はインストール CD より導入時に自動的に設定します。

LogisticaTRUCKServer- II (SQLServer 版)距離計算 DB・アプリケーションサーバ・API・クライアント システム構成図

LogisticaTRUCKServer- II (SQLServer 版)距離計算 DB・アプリケーションサーバ・API・クライアント システム構成図

住友電気システムソリューション(株)より
定期的に更新

(株)はるかシステム研究所 HP より
更新データをロードして自動更新



LogisticaTRUCKServer- II (SQLServer 版)距離計算サーバ API

LogisticaTRUCKServer- II (SQLServer 版)距離計算サーバ API .NET DLL ASP.NET VisualBasic 利用例

サンプルプログラム

```
Imports Microsoft.VisualBasic
Imports System.Threading.Thread

Partial Class _Default
    Inherits System.Web.UI.Page

    Private TRKSVR2NET As lgsTrkSvr2NETX1.TRKSVR2NETX

    Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs)
        Handles Me.Load
            If IsPostBack = False Then
                Me.TextBox起点.Text = "11214"
                Me.TextBox終点.Text = "27201"
            End If
        End Sub

    Protected Sub Button_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles Button.Click
        Dim blnCond As Boolean
        Dim lngIraiNo As Long
        Dim blnReceive As Boolean
        Dim blnKeisanStatus As Boolean
        TRKSVR2NET = New lgsTrkSvr2NETX1.TRKSVR2NETX

        Me.TextBox距離.Text = ""
        Me.TextBox時間.Text = ""
        Me.TextBox有料距離.Text = ""

        TRKSVR2NET.ServerComputerName = "192.139.11.2"
        blnCond = TRKSVR2NET.lgsServerConnect

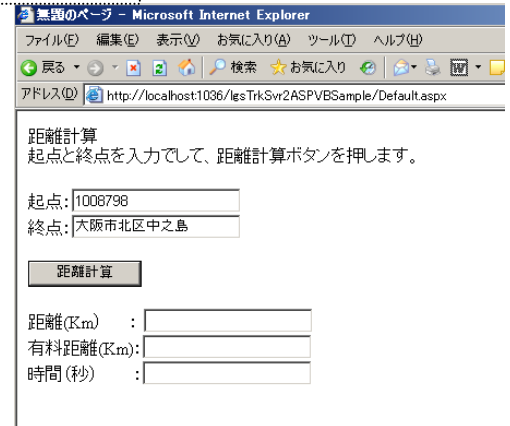
        If blnCond = True Then
            TRKSVR2NET.Kiten = Me.TextBox起点.Text
            TRKSVR2NET.Shuuten = Me.TextBox終点.Text
            TRKSVR2NET.KeisanJouken = TRKSVR2NET.JoukenType.時間優先
            lngIraiNo = TRKSVR2NET.Request
            blnReceive = False

            Do While (blnReceive = False)
                blnReceive = TRKSVR2NET.Reply(lngIraiNo, blnKeisanStatus)
                Sleep(10)
            Loop

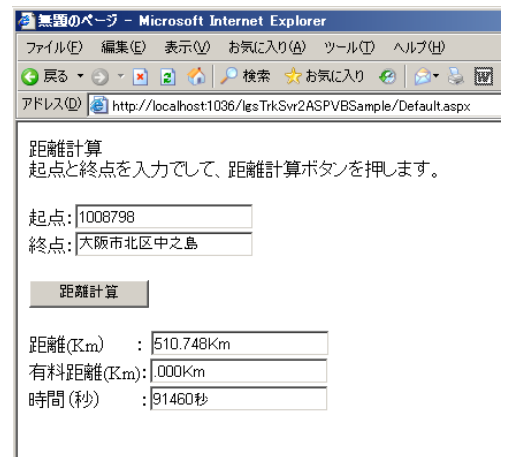
            If blnKeisanStatus = True Then
                Me.TextBox距離.Text = Format(TRKSVR2NET.KyoriM / 1000, "####.000Km")
                Me.TextBox有料距離.Text = Format(TRKSVR2NET.YuuryouKyoriM / 1000, "####.000Km")
                Me.TextBox時間.Text = Format(TRKSVR2NET.JikanS, "####0秒")
            End If
        End If

        TRKSVR2NET = Nothing
    End Sub
End Class
```

実行画面



起点と終点を入力し、距離計算ボタンを押すと、右下の画面になります。



LogisticaTRUCKServer- II (SQLServer 版)距離計算サーバ API ソケット通信 Java 利用例

```

import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.net.Socket;
import java.net.SocketException;
public class SocketSample01 {
    public static void main(String[] args) {
        try {
            String strServer = "192.139.11.6";
            int intServerPort = 44963;
            // ソケット作成
            Socket socket = new Socket(strServer, intServerPort);
            System.out.println("Connected to Server");
            InputStream in = socket.getInputStream();
            OutputStream out = socket.getOutputStream();
            String strSoushin = "272,東京都千代田区大手町 1-1-1,大阪市北区中之島 1-1-2";
            byte[] byteSoushinBuffer = strSoushin.getBytes();
            out.write(byteSoushinBuffer);
            // 距離計算結果を受信する
            int intLength = 4096;
            int intReceived;
            byte[] byteJushinBuffer = new byte[4096];
            if ((intReceived = in.read(byteJushinBuffer, 0, intLength)) == -1)
                throw new SocketException("Connection closed prematurely");
            System.out.println("Received: " + new String(byteJushinBuffer, 0, intReceived));
            // Split
            String strS = new String(byteJushinBuffer, 0, intReceived);
            String[] strStrings = strS.split(",", -1);
            System.out.println("[Length]" + strStrings.length);
            for (int i = 0; i < strStrings.length; i++) {
                System.out.println("[ " + i + "]" + strStrings[i]);
            }
            socket.close();
        } catch (SocketException e) {
            System.err.println("Socket Error");
            System.exit(-1);
        } catch (IOException e) {
            System.err.println("IO Error");
            System.exit(-1);
        }
    }
}

```

実行画面

```

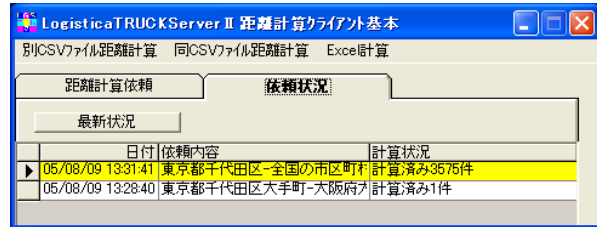
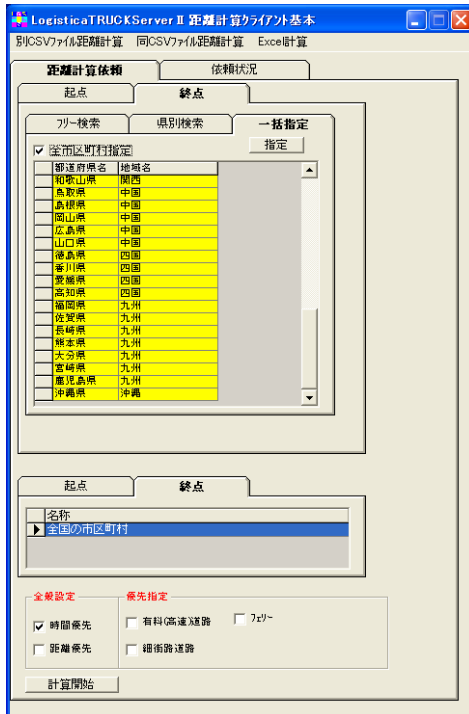
C:\Program Files\IgsPAK01\AX01\SocketSample\Java>java SocketSample01
Connected to Server
Received: 272,東京都千代田区大手町 1-1-1,大阪市北区中之島 1-1-2,512444.0,91781.0,
rue,東京都千代田区大手町 1丁目 1番地 1号,大阪府大阪市北区中之島 1丁目 1番地 2号,
E139.455684 N35.410066,E135.303372 N34.412303
[Length]12
[0]272
[1]東京都千代田区大手町 1-1-1
[2]大阪市北区中之島 1-1-2
[3]512444
[4]0
[5]91781
[6]0
[7]rue
[8]東京都千代田区大手町 1丁目 1番地 1号
[9]大阪府大阪市北区中之島 1丁目 1番地 2号
[10]E139.455684 N35.410066
[11]E135.303372 N34.412303

```

LogisticaTRUCKServer- II (SQLServer 版)距離計算クライアント基本

4° ターン距離計算方式で地図表示しない

① 市区町村コード・郵便番号、または県別リストから選択して距離計算する。終点側は全国市区町村を一括指定可能



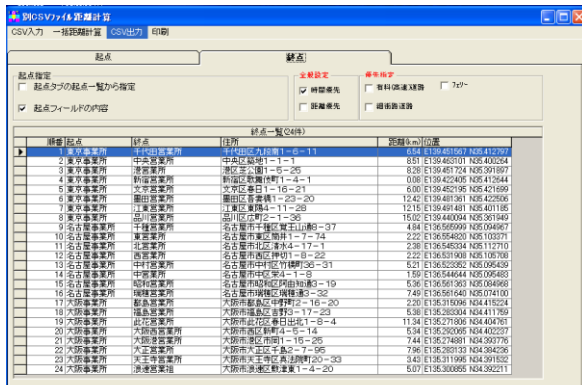
東京葛千代田区-全国の市区町村 計算結果

CSV出力 印刷

| 順番 | 起点コード | 起点名 | 起点かな | 終点コード | 終点名 | 終点かな | 距離(m) | 時間 |
|----|--------|-----------|------|-------|-----------|------|---------|----------|
| 1 | 113101 | 北海道札幌市東区 | 札幌 | 01100 | 北海道札幌市 | 札幌 | 1039444 | 時間46分22秒 |
| 2 | 113101 | 北海道札幌市中央区 | 札幌 | 01101 | 北海道札幌市中央区 | 札幌 | 1037704 | 時間45分30秒 |
| 3 | 113101 | 北海道札幌市北区 | 札幌 | 01102 | 北海道札幌市北区 | 札幌 | 1043042 | 時間46分38秒 |
| 4 | 113101 | 北海道札幌市南区 | 札幌 | 01103 | 北海道札幌市南区 | 札幌 | 1041444 | 時間46分26秒 |
| 5 | 113101 | 北海道札幌市白石区 | 札幌 | 01104 | 北海道札幌市白石区 | 札幌 | 1040328 | 時間46分19秒 |
| 6 | 113101 | 北海道札幌市豊平区 | 札幌 | 01105 | 北海道札幌市豊平区 | 札幌 | 1037476 | 時間46分23秒 |
| 7 | 113101 | 北海道札幌市厚別区 | 札幌 | 01106 | 北海道札幌市厚別区 | 札幌 | 1033587 | 時間45分11秒 |
| 8 | 113101 | 北海道札幌市手稲区 | 札幌 | 01107 | 北海道札幌市手稲区 | 札幌 | 1043232 | 時間46分49秒 |
| 9 | 113101 | 北海道札幌市東区 | 札幌 | 01108 | 北海道札幌市東区 | 札幌 | 1047224 | 時間46分46秒 |
| 10 | 113101 | 北海道札幌市西区 | 札幌 | 01109 | 北海道札幌市西区 | 札幌 | 1054280 | 時間46分44秒 |
| 11 | 113101 | 北海道札幌市南区 | 札幌 | 01110 | 北海道札幌市南区 | 札幌 | 1043003 | 時間46分02秒 |

② 起点の CSV ファイルと終点の CSV ファイルを組み合わせて距離計算する

③ 起点と終点を含む CSV ファイルから距離を計算

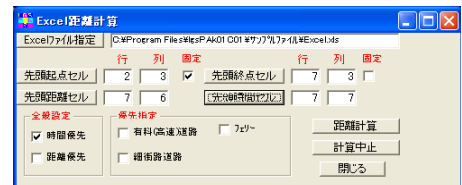
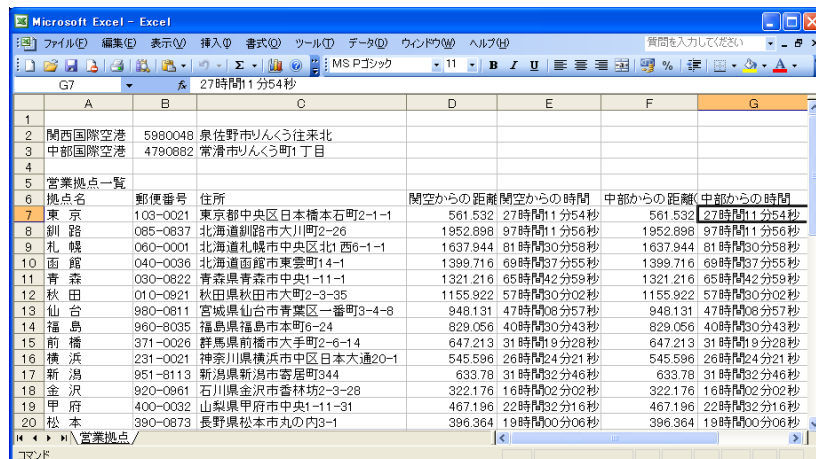


同CSVファイル距離計算

CSV入力 一括距離計算 CSV出力 印刷

| 順番 | 起点住所 | 終点住所 | 距離(m) |
|----|-------------|---------------------|----------|
| 1 | 2 811-3436 | 北海道札幌市中央区北1条西2 | 2,160.02 |
| 2 | 3 170-0847 | 北海道紋別郡九軒町中町115-2 | 1,965.27 |
| 3 | 4 722-0045 | 宮城県気仙沼市八戸町1-1-1 | 1,259.10 |
| 4 | 5 640-8156 | 佐賀県唐津市今泉町1-6-1 | 1,152.17 |
| 5 | 6 620-0035 | 福島県若狭町石町大字鏡田字不持沼345 | 793.81 |
| 6 | 7 500-8822 | 茨城県高萩市本町1-100 | 558.23 |
| 7 | 8 102-0074 | 群馬県太田市浜町2-35 | 83.50 |
| 8 | 9 336-0000 | 北海道札幌市中央区北1条西2 | 1,026.60 |
| 9 | 10 080-0001 | 東京都千代田区九段南1-6-11 | 964.80 |
| 10 | 11 009-0203 | 埼玉県埼玉市中央区1-3 | 1,608.89 |
| 11 | 12 988-0084 | 京都府福知山市宇治池13-1 | 1,054.86 |
| 12 | 13 018-0402 | 和歌山県和歌山市七番丁23 | 1,148.71 |
| 13 | 14 969-0401 | 徳島県尾道市久保1-6-1 | 1,004.05 |
| 14 | 15 318-0033 | 徳島県徳島市幸町2-5 | 847.90 |
| 15 | 16 373-0853 | 福岡県宗像市大字東郷955 | 1,117.63 |
| 16 | 17 336-0000 | 沖縄県那覇市泉崎1-1-1 | 1,518.99 |

④ Excel シート上で距離計算



LogisticaTRUCKServer- II (SQLServer 版)距離計算 DB・アプリケーションサーバ・API・クライアント構成と価格

価格体系

| 製品群 | 製品名 | 構成 | 価格 (税込) |
|--|---|---|----------|
| 【A】 LogisticaTRUCKServer- II (SQLServer 版)距離計算 DB・アプリケーションサーバ | LogisticaTRUCKServer- II (SQLServer 版)距離計算DBサーバ | 住所基本DB(JIS 市区町村・一般郵便番号・個別事業所郵便番号) | ¥432,000 |
| | | PowerAtlas 全国デジタル道路地図・街区位置 DB | |
| | | 距離計算・位置検索 DB | |
| | LogisticaTRUCKServer- II (SQLServer 版)距離計算アプリケーションサーバ | アプリケーションサーバ LogisticaTRUCKServer- II (SQLServer 版)距離計算クライアント基本 1 クライアント | |
| 【B】 LogisticaTRUCKServer- II (SQLServer 版)距離計算サーバ API | API モジュール | .NET DLL | ¥432,000 |
| | | Active-X コントロール | |
| | | Active-X DLL | |
| | ソケット通信コントローラ | TCP ソケット通信コントローラ | |
| | ストアドプロシージャ | T-SQL 距離計算ストアドプロシージャ | |
| | 各言語サンプルプログラム | WindowForms VisualBasic C# | |
| | | WebForms ASP.NET VisualBasic C# | |
| | | Active-X コントロール VB6 Access200X | |
| | | Active-X DLL VB6 VBScript ASP | |
| ソケット通信サンプルプログラム | T-SQL 距離計算ストアドプロシージャ ソケット通信 Java Perl PHP ほか | | |
| 【C】 LogisticaTRUCKServer- II (SQLServer 版)距離計算クライアントアプリケーション | クライアントアプリケーション 【C-1】 | LogisticaTRUCKServer- II (SQLServer 版)距離計算クライアント基本 1 クライアント | ¥43,200 |

製品の組み合わせ

【A】LogisticaTRUCKServer- II (SQLServer 版)距離計算 DB・アプリケーションサーバは必須。そのほかは用途に応じて、【B】LogisticaTRUCKServer- II (SQLServer 版)距離計算サーバ API、【C-1】を組み合わせご利用いただきます。
製品群【A】には、LogisticaTRUCKServer- II (SQLServer 版)距離計算クライアント基本の1クライアントが含まれます。
製品群【B】LogisticaTRUCKServer- II (SQLServer 版)距離計算サーバ API を利用するには、製品群【A】は必須です。

年間データ更新料

住友電工フィールドシステムズ(株)提供 PowerAtlas 全国デジタル道路地図 各年度版を含む。導入後、2 年目以降、**全国道路地図・データ更新料 年間¥108,000**
2 年目以降、郵便番号・JIS 市区町村コードのみの**データ更新料 年間¥64,800**

評価試用版 CD の無料貸出

導入以前に、動作確認・評価試用のための評価試用版 CD の無料貸し出しがあります。

説明・デモのご案内

説明・デモをご希望の方は、ご連絡ください。

(株)はるかシステム研究所 〒344-0041 埼玉県春日部市増富 437-7 TEL048(763)8165
電子メール: info@harukasystem.co.jp URL : <http://www.harukasystem.co.jp>