

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

MicrosoftSQLServer2019/2017/2016/2014/2012 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(住友電工システムソリューション(株)製 PowerAtlas 使用)・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、**2021 年版全国デジタル道路地図**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。

市区町村・郵便番号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、**クライアント基本地図**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 距離計算ストアードプロシージャ、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#	
ストアードプロシージャ		T-SQL	T-SQL 距離計算ストアードプロシージャ
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	

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

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	動作環境
LogisticaTRUCKServer- II (SQLServer 版)距離計算クライアント基本地図	基本距離計算とルート地図表示機能	Windows10/8.1/7 IE(インターネットクスプローラ) 11 必須

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 WindowsServer2019/2016/2012R2/2012
サーバ DB	MicrosoftSQLServer2017/2016/2014/2012 動作確認用に SQLSe2019/rver2012/Express を同梱

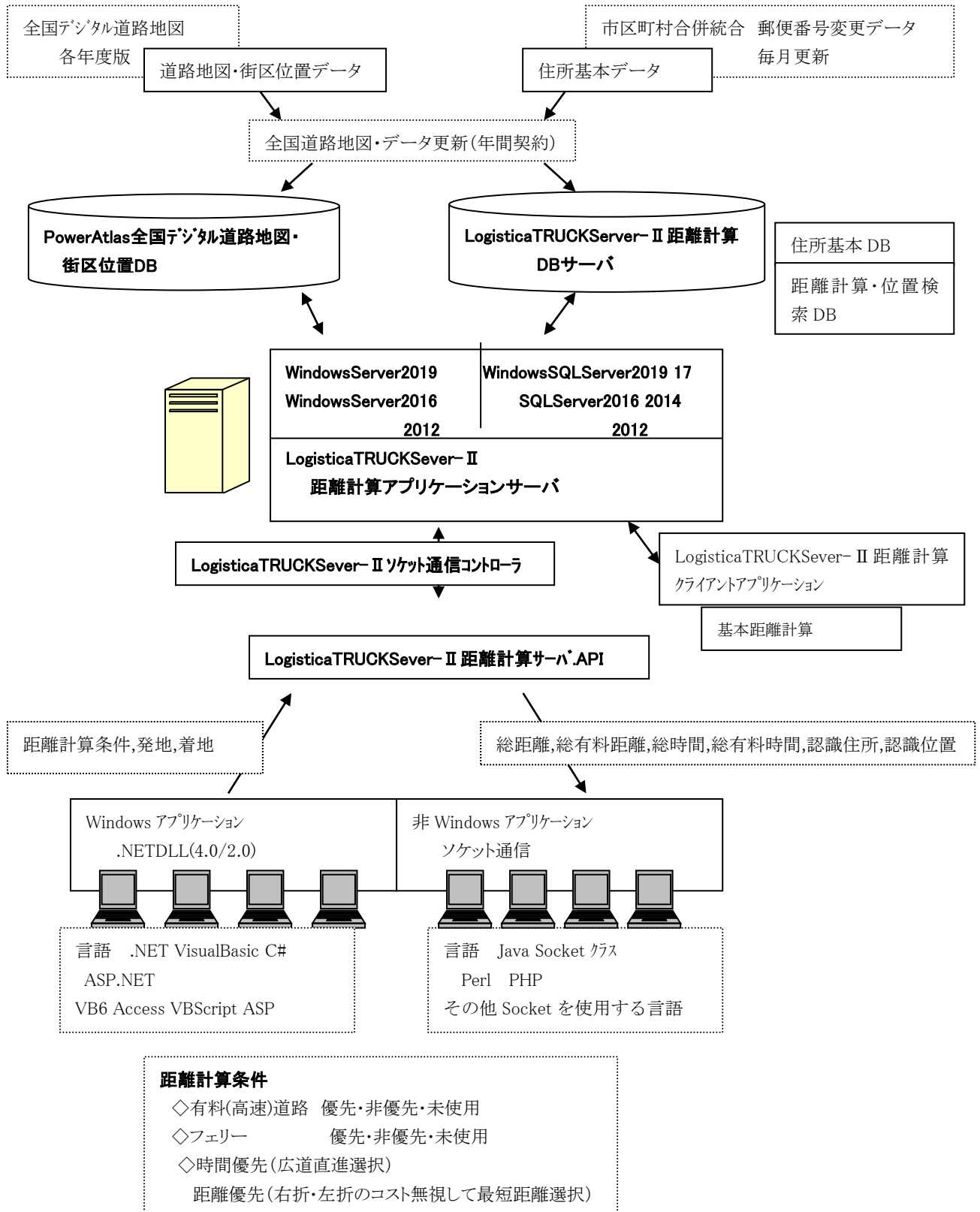
そのほかの動作環境はインストール CD より導入時に自動的に設定します。

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
```

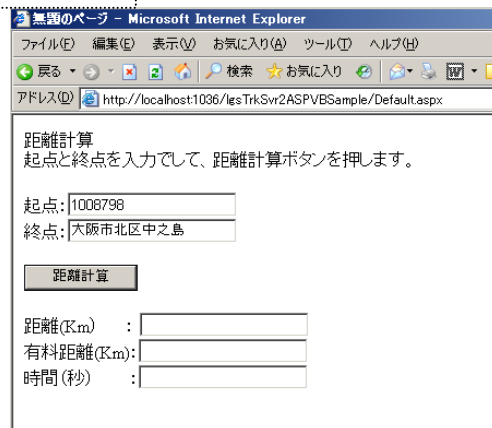
```
    End If
```

```
    TRKSVR2NET = Nothing
```

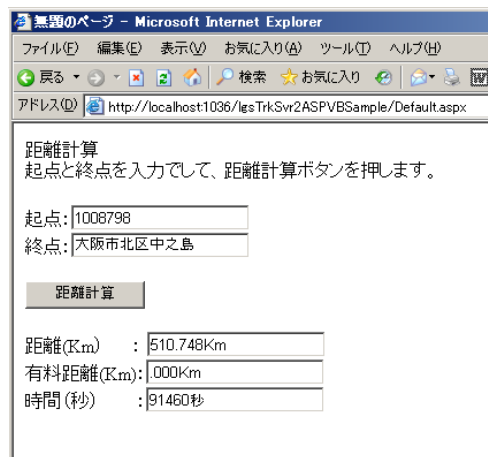
```
End Sub
```

```
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。



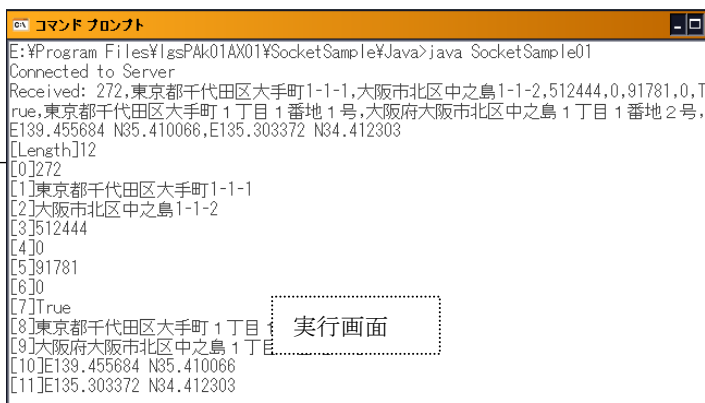
LogisticaTRUCKServer-II (SQLServer 版)距離計算サーバ API ソケット通信 Java 利用例

```
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);
        }
    }
}
```



```
コマンド プロンプト
E:\Program Files\Igs\PAK01\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 1-1-2,512444,0,91781,0,1
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]true
[8]東京都千代田区大手町 1 丁目
[9]大阪府大阪市北区中之島 1 丁目
[10]E139.455684 N35.410066
[11]E135.303372 N34.412303
```


起点一括距離計算

拠点(起点)から全国の市区町までのキロ程表を作成します。計算結果は Excel ファイルに出力できます。

起点一括距離

起点 終点 計算条件 計算 出力

計算条件
【距離優先】【有料道路使用(非優先)】【出発点一般道路上】

Excel xlsx
CSV xls

JIS市区町村コード	JIS市区町村	JIS市区町村かな	現旧	検索結果1	検索結果2	検索結果3
13113	東京都渋谷区	しぶや	1	13113	東京都渋谷区	しぶや

終点
茨城県 栃木県 群馬県 埼玉県 千葉県 東京都 神奈川県

市区町村検索
 現 現旧

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Km	時間	有料距離Km	有料時間
1	1 13113	東京都渋谷区	しぶや	08201	茨城県水戸市	みと	115.14	003:28:28	0.00	000:00:00
2	2 13113	東京都渋谷区	しぶや	08202	茨城県日立市	ひたち	147.40	004:16:28	0.00	000:00:00
3	3 13113	東京都渋谷区	しぶや	08203	茨城県土浦市	つちうら	71.49	002:13:14	0.00	000:00:00
4	4 13113	東京都渋谷区	しぶや	08204	茨城県古河市	こが	67.19	002:03:12	0.00	000:00:00
5	5 13113	東京都渋谷区	しぶや	08205	茨城県石岡市	いしおか	88.09	002:39:08	0.00	000:00:00
6	6 13113	東京都渋谷区	しぶや	08207	茨城県結城市	ゆうき	85.04	002:21:07	0.00	000:00:00
7	7 13113	東京都渋谷区	しぶや	08208	茨城県龍ヶ崎市	りゅうがさき	59.29	001:49:26	0.00	000:00:00

スプレッド距離計算

スプレッド距離

入力 計算条件 計算 出力 指定列

列名	開始セル	終了セル
起点住所	B2	B7
終点住所	D2	D7
距離	E2	E7
時間		
有料距離		
有料時間		

計算条件
【距離優先】【有料道路使用(非優先)】【フェリー使用(非優先)】【細街路道路(非優先)】【出発点一般道路上】

入力ファイル
C:\%ls%\lsPAk01C14\Sample\距離\Excel\県庁間.xlsx

	A	B	C	D	総距離 (km)	F
1	都道府県名	県庁所在地	都道府県名	県庁所在地		
2	北海道	北海道札幌市中央区北3条西6丁目	東京都	東京都新宿区西新宿2-8-1	190.677	
3	宮城県	宮城県仙台市青葉区本町3丁目8番1号	神奈川県	神奈川県横浜市中区日本大通1	377.376	
4	兵庫県	兵庫県神戸市中央区下山手通5丁目10番1号	埼玉県	埼玉県さいたま市浦和区高砂3-15-1	552.872	
5	広島県	広島県広島市中区基町10-52	千葉県	千葉県千葉市中央区市場町1番1号	870.104	
6	高知県	高知県高知市丸ノ内1丁目2番20号	愛媛県	愛媛県松山市一番町4丁目4-2	113.448	
7	鹿児島県	鹿児島県鹿児島市鴨池新町10番1号	福岡県	福岡県福岡市博多区東公園7番7号	285.385	
8						

Excel2016 などの Excel シート・CSV データを取り込んで距離計算をします。

LogisticaTRUCKServer- II (SQLServer 版)距離計算 DB 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) 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	¥440,000
	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地図 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 (2.0/4.0)	¥440,000
	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#	
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地図 1 クライアント	¥44,000

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¥110,000**

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¥66,000**

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>

LogisticaTRUCKServer-Ⅱ 距離計算サーバ 試用版 CD 送付 FAX 申込書

必要事項を御記入の上、下記 FAX 番号へFAXしてください。

0 4 8 - 7 6 3 - 1 8 2 8

試用版CDのお申し込み

LogisticaTRUCKServer-Ⅱ 距離計算サーバ 試用版 CD (無料貸出)送付申し込み。

御送付先(名刺貼り付けでもかまいません)

送付先住所	〒
御社名	
部署名	
御担当者名	
電話番号	
FAX番号	
電子メール	

御使用予定のパソコン環境について

メーカー名・機種名		メモリ容量	GB
使用OS	<input type="checkbox"/> Windows10 <input type="checkbox"/> Windows8.1 <input type="checkbox"/> Windows7 <input type="checkbox"/> WindowsServer2012 <input type="checkbox"/> 2016 <input type="checkbox"/> 2019	HD容量	GB

商品名および社名などは、各社の商標または登録商標です。