JRA－VAN　Data　Lab. SDK

開発ガイド

2022年2月22日

第4. 2. 2版

JRAシステムサービス株式会社

修正履歴

日付	版	項番	種類	内容
2022/2/22	4.2.2	ー	修正	スタートキット(CD-ROM)の提供終了に伴い、関連する記述を修正
		3.4.2	修正	「プロジェクトの作成」手順の誤りを修正
2021/5/26	4.2.1	ー	修正	VisualStudio2019 へのバージョン更新
2021/5/13	4.2.0	2.1	修正	JVーLink動作環境を更新(OS・ブラウザ)
		3.3.2	追記	セットアップデータ取得についての説明を追記
		3.3.4	追記	FROM タイム指定についての説明を追記
2018/9/4	4.1.0	2.1	修正	JVーLink動作環境を更新
2017/6/6	4.0.0	ー	修正	VisualStudio2015 へのバージョン更新
2008/6/17	3.0.0	ー	修正	Windows Vista 対応によりバージョン更新
2008/2/26	2.4.0	1	修正	現在のサービス内容に合わせ説明を修正
2008/2/26	2.4.0	2.1	修正	JVーLink動作環境を修正
2006/6/6	2.1.4.1	3.3.4	追記	セットアップ時のFROMタイムについての説明を追記
		3.4.6	修正	JV-Data 構造体について説明を追加
2004/3/2	1.1.4	3.4.5	修正	・サンプルソース 1 の JVSetUIProperties 判定処理のコメントを修正 誤:「-1」→「-100」 サンプルソース 1 の解説の誤りを修正 誤:「-1」→「-100」
2003/7/11	1.0.7β	3.4.5	修正	・JVSetUIProperties のエラー判定条件を修正 誤:「-1」→「-100」
		3.4.6	修正	VB.net 版 JV-Data 構造体の修正に伴い、レコードヘッダ構造体へのデータ展開メソッドが変更になったため、サンプルソースのレコード種別IDの判定方法を変更。
2003/5/2	1.0.3β	3.4.6	修正	サンプルソース2の解説の誤りを修正 ・誤:「オプションに 1 をセットします」→正:「オプションに 2 をセットします」
2003/4/22	1.0.1β	1.2.1	修正	「1．3 JRAーVAN Data Lab. SDKとは」→「1．4 JRAーVAN Data Lab. SDKとは」
		1.2.3	修正	「5 JV-Data(JRA-VAN 提供データ)仕様」→「JV-Data 仕様書」
		3.1.1	追加 修正	・開発環境に Delphi 6.0 を追加 ・「VC＋＋、VB、Delphi用」→「主な開発環境用」
		3.1.2	追加	・開発環境製品名にメーカー名を追加 ・推奨開発環境について説明を追加
		3.2	追加	JVーLinkのインストール手順を追加
		3.4.7	修正	サンプルソース3の不具合を修正
		A.1	修正	・Delphi 6.0 の製品名の誤りを修正 ・C++ Builder サンプルプログラム追加 ・説明文の修正
		A.2.4	修正	コントロールの構成を修正
		A.2.5	修正	注意事項を修正
		A.3	修正	サンプルプログラム修正
2003/4/1	1.0β	ー	新規	初版作成

目次

APPENDIX

はじめに

平素よりJRA－VANをご利用いただき誠にありがとうございます。

弊社では JRA 公式データをより一層活用できる仕組みとして、「JRA－VAN Data Lab.」サービスを提供しております。本サービスは、データリンクモジュールのJV－Link（※）を通じて、様々な競馬データの利用が可能になるサービスです。
エンドユーザー様、およびソフトウェア作者の皆様には、この「JRA－VAN Data Lab.」サービスの仕組みをご理解いただき、JRA－VANのデータ提供サービスを活用していただきたいと願っております。

※JV－Linkについては「1．2 JV－Linkとは」を参照して下さい。

1 JRA－VAN Data Lab. 概要

1．1 JRA－VAN Data Lab. とは

JRA－VAN Data Lab. は、JRA 公式データの配信サービスです。Data Lab. は次に挙げる項目を実現することにより、エンドユーザー様、ソフトウェア作者様、JRA-VAN それぞれがメリットを受けられることを目標にしたサービスです。

・必要なデータのダウンロードを競馬ソフトが自動的に行うことでエンドユーザー様の利便性を向上
・競馬ソフトの目的により様々なデータ取得単位を提供
・競馬ソフト開発情報やサンプルプログラムの提供による競馬ソフトの開発生産性の向上
・データ不正利用の防止

上記からも分かる通り、JRA－VAN Data Lab. はエンドユーザー様が競馬データを直接取得、閲覧するサービスではなく、競馬ソフトを通じて競馬データを閲覧するサービスです。

エンドユーザー様は、それぞれの目的や好みで競馬ソフトを選び、利用することができます。

ソフトウェア作者様は、JV－Linkを利用して、競馬データを直接、また簡単にソフトに取り込むことが可能です。また、当ドキュメントでは説明を割愛しますが、報奨制度にエントリーすることで報奨金を受け取ることも可能です。

※報奨制度は報奨金規約に定める手続きを経てエントリーを行う必要があります。
詳細は JRA-VAN サイトの各種手続き・確認（作者サポート） ページ https://jra-van.jp/dlb/sdv/support.html をご参照ください。

1．2 JV－Linkとは

先ほど、「JRA－VAN Data Lab. は、エンドユーザー様が競馬データを直接取得、閲覧するサービスではなく、競馬ソフトを通じて競馬データを閲覧するサービス」であると書きましたが、この仕組みを競馬ソフトに提供するのがJV－Linkです。

JV－Linkは競馬ソフトがJRA－VAN Data Lab. サーバーへアクセスする際のインターフェースモジュールになります。ATLで作成されたActiveX COMサービスとして提供され、JRA－VAN Data Lab. 対応の競馬ソフトを利用するには、競馬ソフトのインストールとともにこのJV－Linkをインストールする必要があります。

また競馬ソフトを開発する場合はJV－Linkが提供する各インターフェースを利用することでJRA－VANのサーバーへのアクセスを意識することなく、データを読み込むことが出来ます。（※）

※JV－Linkの詳しい使用方法については「3 開発ガイド」を参照してください。

JV－Linkは以下の機能を競馬ソフトに提供します。

・通常データの自動ダウンロード機能と読み出し機能
・今週分データの自動ダウンロード機能と読み出し機能
・セットアップデータの自動ダウンロード機能と読み出し機能
・データのキャッシュ機能

1．2．1 JV－Linkによるデータ取得方法

JRA－VAN Data Lab. ユーザーの具体的なデータ取得方法は以下のようなイメージとなります。

・競馬ソフトのユーザーインターフェースからデータ取り込み機能を呼び出す事によってデータを取得（取得元はJRA－VANサイト）

エンドユーザー様はどんな競馬データを取得する必要があるのかなどを意識せず、競馬ソフトを立ち上げて取り込み機能を呼び出すだけで、データが利用可能になります。

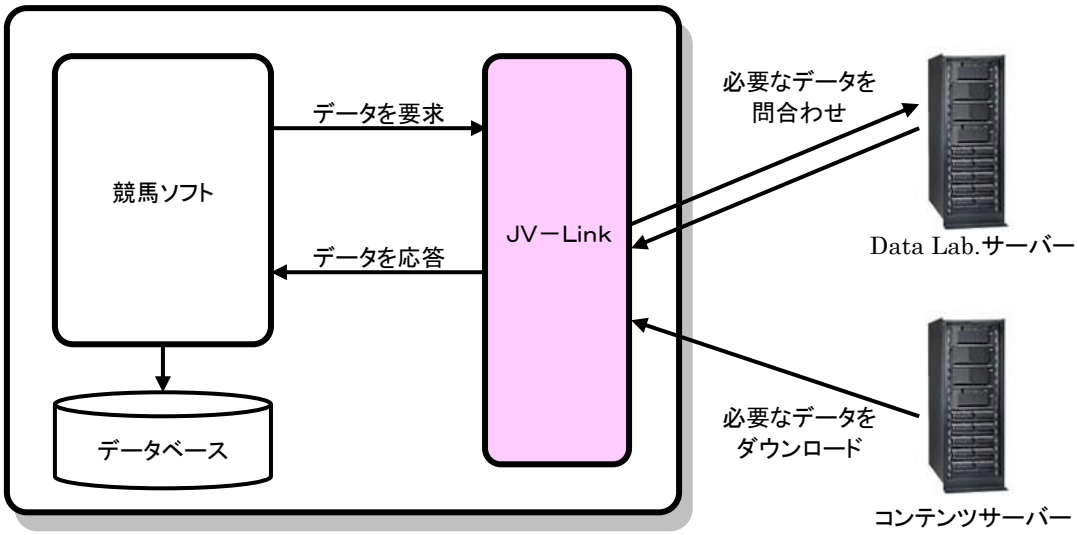
JRA－VANでは、このようなユーザー環境をより簡単に各ソフトに実装していただけるよう、競馬ソフトの開発ツールとして「JRA－VAN Data Lab. SDK」（※）をソフトウェア作者様にご提供します。

※「JRA－VAN Data Lab. SDK」については「1．3 JRA－VAN Data Lab. SDKとは」を参照して下さい。

「JRA－VAN Data Lab. SDK」にはJRA－VANサーバーのインターフェースモジュールである「JV－Link」が含まれており、競馬ソフトはこの「JV－Link」の機能を呼び出してデータを取得することになります。基本的に競馬ソフトは必要とする**データの種類**と**期間**を指定すればよく、そのデータを取得するために必要な手続きは全て「JV－Link」が行います。

ただし「JV－Link」はオンラインデータ照会のための機能ではなく、あくまでも競馬ソフトが持つデータベースへ最新データを取り込むための機能となっています。したがって競馬データの画面表示やレースの予想計算は競馬ソフトが持つデータベースを利用してください。

データ取得概念図



1．2．2 データの暗号化・復号化

JRA－VAN Data Lab. では提供するデータを圧縮するとともに暗号化を施して送信します。ただし、競馬ソフトはJV－Linkから復号化済のデータを受け取るため、復号化処理を意識する必要がありません。

1．2．3 データ仕様について

JRA－VAN Data Lab. では、JRA－VAN競馬データを総称して「JV－Data」と標記します。

提供されているデータの種類、データフォーマット、提供タイミングなどについては、

「JV－Data仕様書」を参照してください。

1．3 JRA－VAN Data Lab. SDKとは

「JRA－VAN Data Lab. SDK（Software Development Kit）」はData Lab. 対応競馬ソフトを作成するための開発キットとなります。「JRA－VAN Data Lab. SDK」は JRA-VAN サイトのソフトウェア開発キット(SDK)提供コーナー(https://jra-van.jp/dlb/sdv/sdk.html）で配布しております。

SDKには以下のものが含まれます。

- **「JRA－VAN Data Lab. SDK開発ガイド」（当ドキュメント）**
- **「JV－Data仕様書」**
- **「JV－Linkインターフェース仕様書」**
- **最新版JV－Linkモジュール**
- **サンプルプログラム**
- **JV－Data構造体ソース**
- **Data Lab. 検証ツール**

２ システム要件

２．１ JV－Link動作環境

オペレーティングシステム

Windows 8.1、 10

※Windows 8.1、 10 では 32bit・64bit 版ともにご利用いただけます。

※Windows 10 Mobile、Windows RT については、非対応となります。

※いずれの OS も日本語版のみの対応となります。

ブラウザ

Microsoft Edge

Google Chrome

Internet Explorer ※非推奨

CPU

1GHz 以上の 32 ビットまたは 64 ビットの CPU

メモリ

1GB 以上、 推奨：2GB 以上

その他

インターネット接続環境が必要です。（ブロードバンド接続を推奨）

3 開発ガイド

3．1 開発環境

「JRA－VAN Data Lab. SDK」を使って競馬ソフトを開発するにはVisualStudioなどのプログラム開発環境が必要です。開発環境には様々なものが存在しますが、ここでは「JRA－VAN Data Lab. SDK」を使って開発するのに必要な開発環境について記述します。

3．1．1 JRA－VAN Data Lab. SDKが利用可能な開発環境

「JRA－VAN Data Lab. SDK」にはプログラム開発に必要なJV－Linkというモジュールが含まれています。JV－Linkは多くの開発環境で効率的な開発が可能となるようActiveX COMサービスとして作成されており、また、よりコンパクトで高速な動作をさせるためATL（Active Template Library）を利用して作成されています。そのため、JV－Linkは ActiveX COM サービスが利用可能な（あるいは Windows の DLL が呼び出せる）開発環境であれば利用可能です。

ActiveX COMサービスについて

ActiveX COMサービスが何であるか分からない方のために簡単な説明をします。

VC＋＋やVB、Delphiでプログラム開発を行なったことのある方は、ActiveXコントロールについての知識はお持ちだと思います。これらの開発環境でダイアログなどを作成する場合にはボタンコントロールやリストボックスコントロールなどを直接ダイアログボックス(フォーム)にドラッグ &ドロップで貼り付けて作成することが可能です。これはプログラムのGUIの要素であるボタンやリストボックスの描画や振る舞いを自分でプログラミングするよりもはるかに開発効率がよく、開発されたプログラムを使う側のユーザーにとっても見た目や振る舞いが統一され使い易いプログラムの開発が可能になります。**JV－Link**もこれらのコントロールの一種になります。したがってフォームにドラッグ &ドロップで貼り付けて機能を呼び出すことが可能です。しかしながらこれらコントロールには様々な種類があり、その呼ばれ方もOLEコントロールやOCXなどと様々であるためその分類はまぎらわしいものとなっています。ただ、通常プログラム開発をするにあたってそれぞれの分類までを意識する必要はあまり無いので開発者は便利なプログラミングパーツとしてひとくくりに考えている場合がほとんどだと思います。

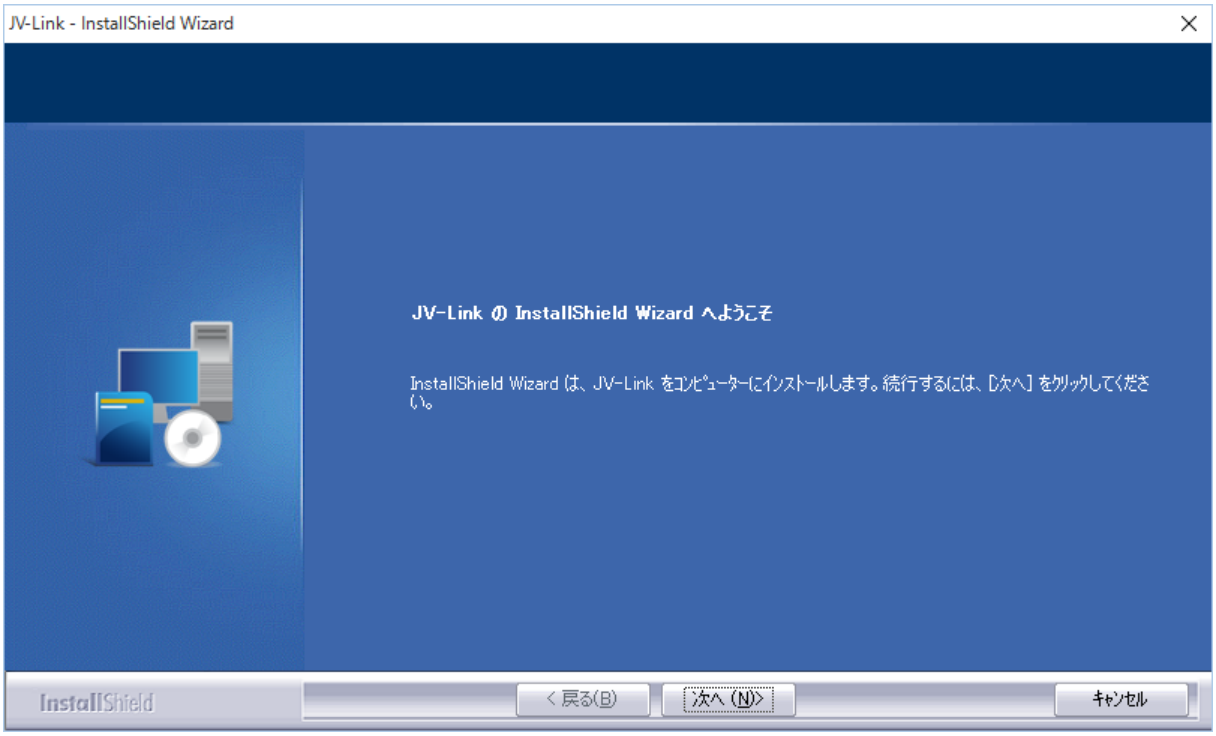
JVLinkはActiveX COMサービスとして提供されており、リストボックスコントロールなどと同様の使い方をしますが、「コントロール」と言わず「サービス」と言っているのはボタンやリストボックスのように目に見えるGUIを持たない場合に「サービス」という分類に入るからです。また、「COM」と付くのは Microsoft の Component Object Model：COMの仕様に基づいて作成されているからです。COM仕様について記述しようとすれば一冊の本になってしまいますので、ここではCOM仕様で作成することによってJV－Linkの汎用性が劇的に高まるという利点があるということにとどめておきます。

3．2 JRA－VAN Data Lab. SDKのインストール

「JRA－VAN Data Lab. SDK」を使って競馬ソフトを開発するには開発環境を用意した後に「JRA－VAN Data Lab. SDK」を開発用のマシンにインストールする必要があります。

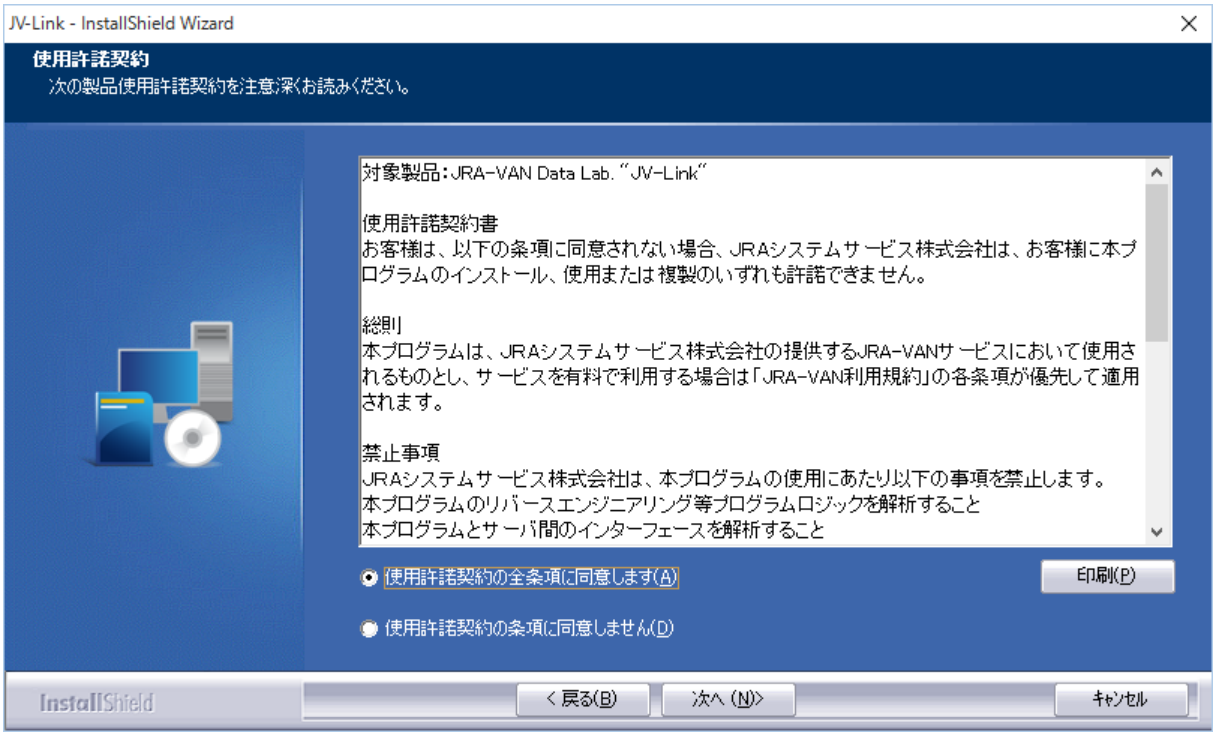
3．2．1 インストーラーの起動

JV－Link.exe をダブルクリックしインストーラーを起動し、「次へ」ボタンをクリックします。



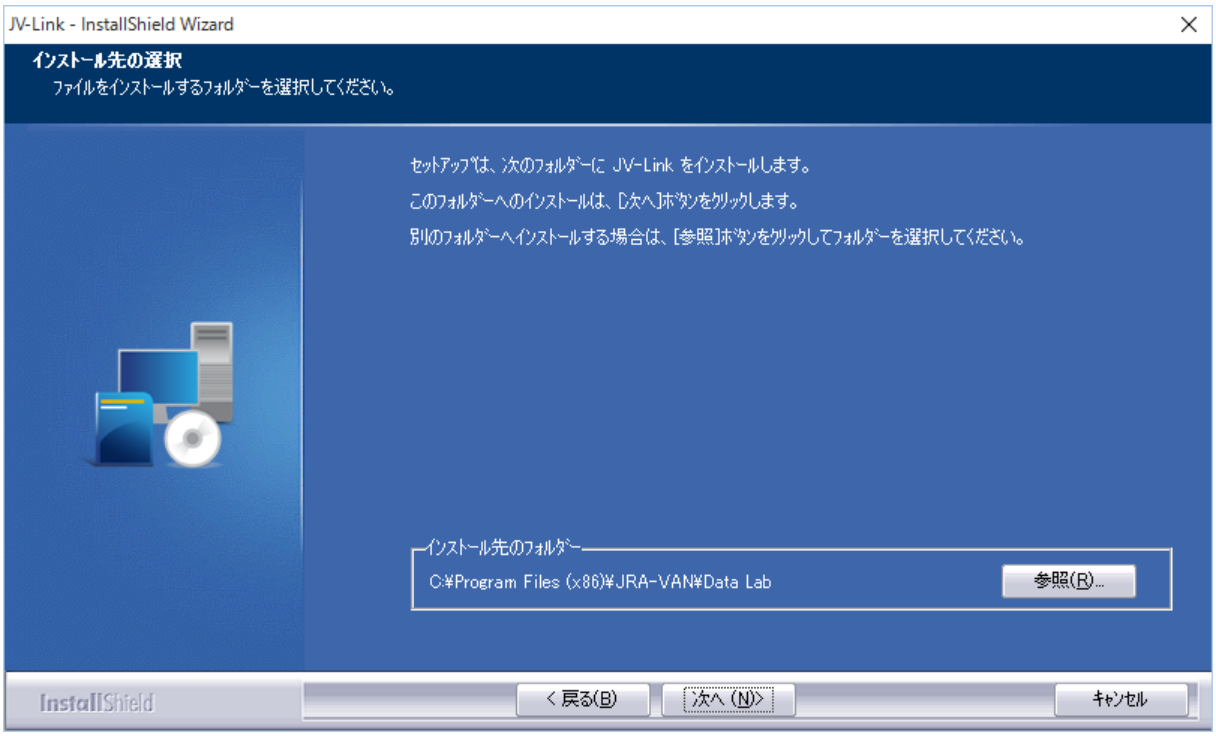
3．2．2 使用許諾契約

使用許諾契約書を読んで「同意します」にチェックを入れ、「次へ」ボタンをクリックします。



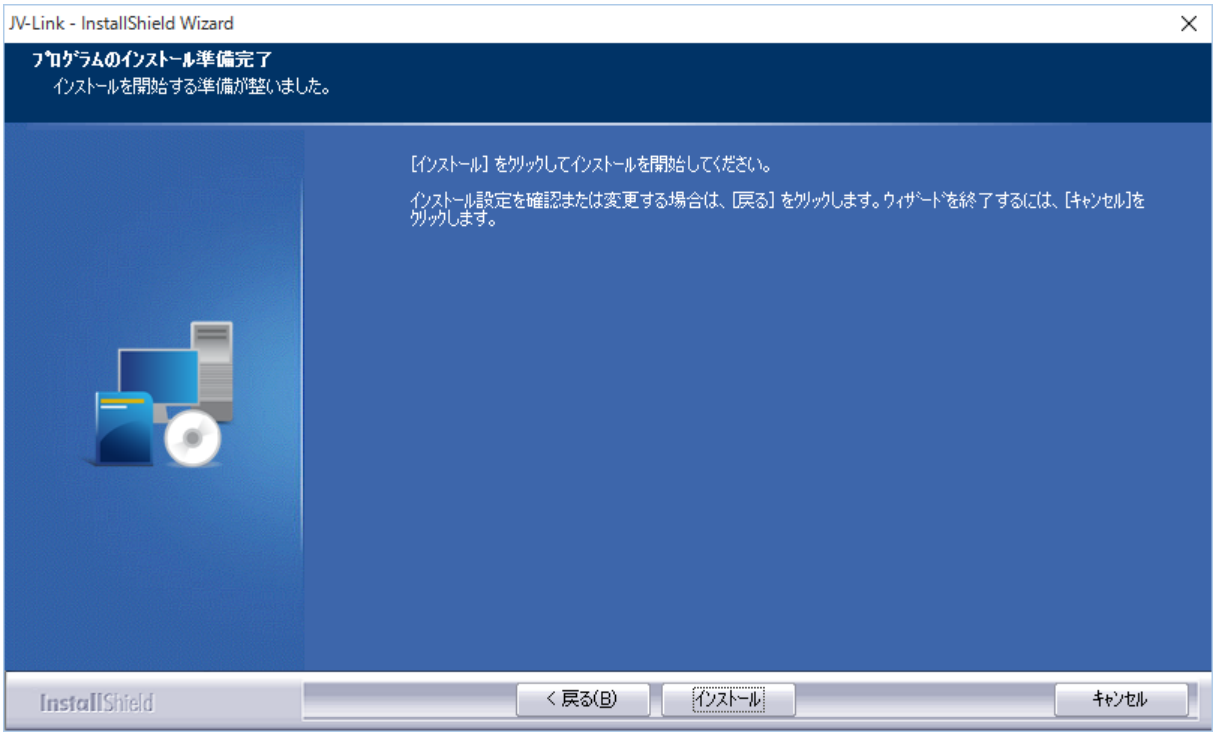
3．2．3 インストール先の指定

JV－Linkのインストール先を指定して「次へ」ボタンをクリックします。

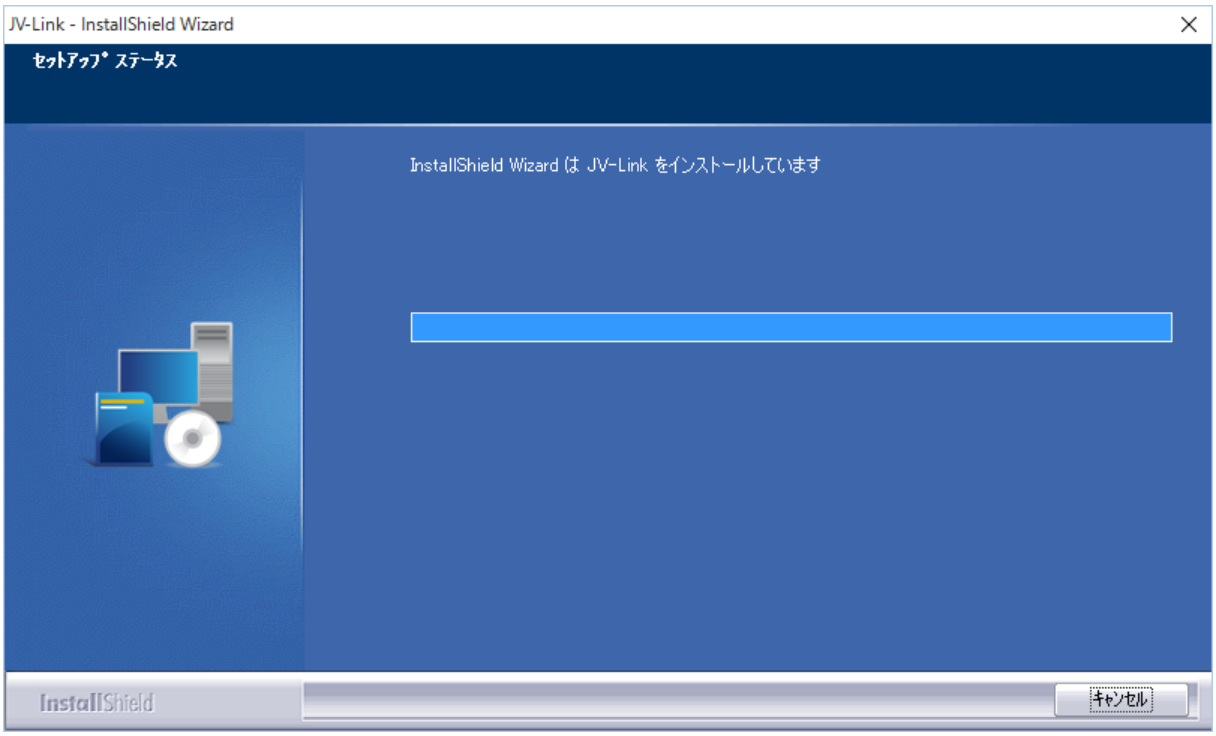


3．2．4 インストールの開始

「インストール」ボタンをクリックしてインストールを開始します。

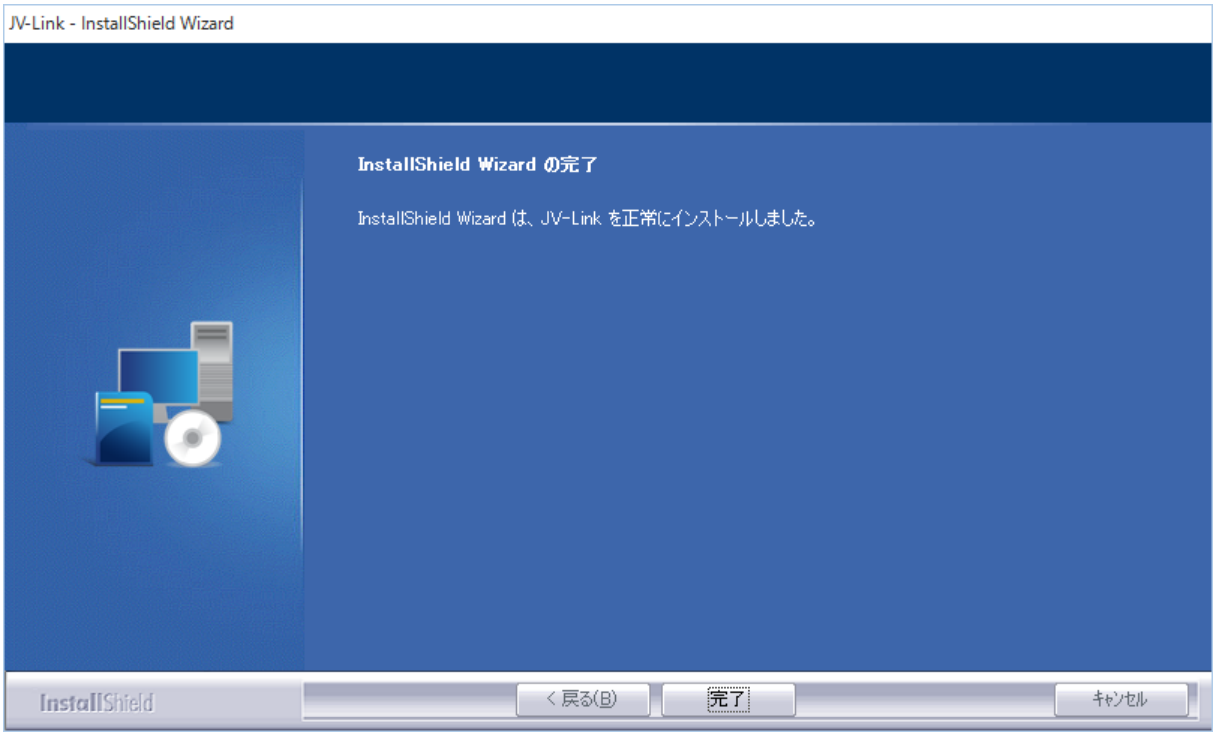


3．2．5 ファイルコピーの実行



3．2．6 インストールの完了

「完了」ボタンをクリックすると、インストールが完了します。



3．3 JV－Data取得方法の概念

3．3．1 競馬ソフトが必要とするデータの2つのパターン

JRA－VANのデータを利用する競馬ソフトが必要とするデータの保持方法は大きく2つのパターンに分けることができます。またこの2つのパターンに相対して競馬ソフトを2つの性格に分類することができます。JRA－VAN Data Lab. ではこの分類を明確に意識したデータ種別、データ取得の仕組みを用意しています。

一つは過去のある時点から現在までの完全なデータを保持するパターンで、このパターンを必要とするソフトはJRA－VANのデータをデータベースとして利用し様々な角度からの分析や検索を主機能とするソフトになります。JRA－VAN Data Lab. ではこのような性格を持つソフトを**「蓄積系ソフト」**と呼びます。もちろん従来のサービスではエンドユーザーがデータを選択してソフトに組み込んでいましたので、そこにはデータの抜けが発生する可能性もあるので従来の競馬ソフトはデータの抜けを考慮した作りになっているはずですが、JRA－VAN Data Lab. では抜けが発生しないデータ取得が可能になっています。

もう一つは今週開催されるレースに関する情報だけを必要とするパターンで、主に競馬新聞に代わるものとして利用されるソフトや予想を行なうソフトなどがこれにあたります。JRA－VAN Data Lab. ではこのような性格を持つソフトを**「非蓄積系ソフト」**と呼びます。「非蓄積系ソフト」はその名の通りデータを完全に蓄積する必要はなく、データに期間的な抜けがあっても今週開催されるレースに関するデータにさえ抜けが無ければ主たる機能に影響がないソフトということになります。JRA－VAN Data Lab. では今週の開催に関するデータだけを漏れなく取得することが可能になっています。

したがって新しくJRA－VAN Data Lab. 対応の競馬ソフトを作成したり、既存の競馬ソフトをJRA－VAN Data Lab. 対応に変更する場合にどちらのパターンを選択するかを決定する必要があります。

3．3．2 蓄積系ソフトのデータ取得の方法

蓄積系ソフトは過去のある時点からの完全なデータを保持しますが、一般的に言って過去数年分、あるいは全過去データを持たないと機能的に面白みがありません。もちろんこれはエンドユーザーの考え方次第ではありますが、少なくとも過去のレース結果を照会することが出来るソフトにとって過去データは多ければ多いほどその機能を発揮することができると言えるでしょう。

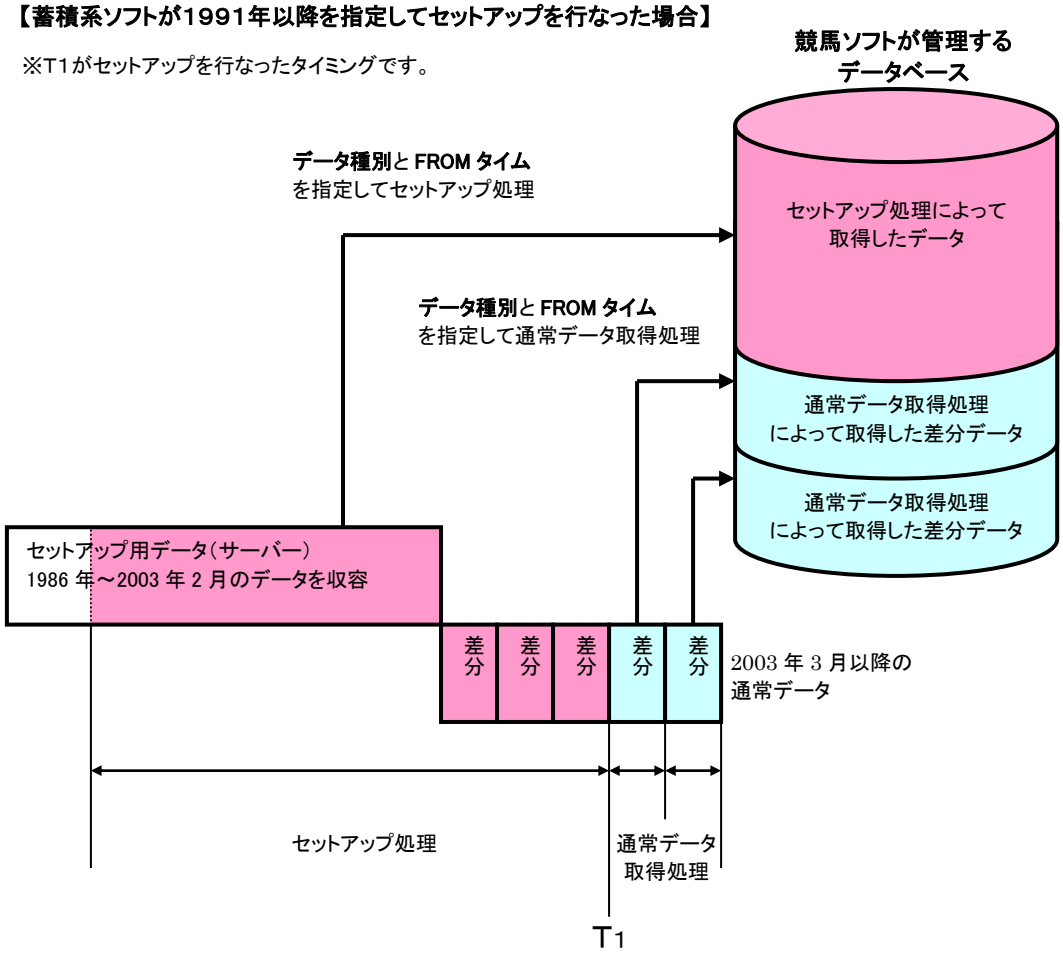
したがって「蓄積系ソフト」はインストールされた後、まず初めに過去データの取り込みを行なう必要があります。JRA－VAN Data Lab. ではこの処理を**「セットアップ」**と呼び、JV－Linkはこのためにセットアップ機能を提供します。セットアップはJRA－VANのサーバーからデータをダウンロードして行います。

セットアップによって過去から現在までの情報を漏れなく取り込んだ後は、日々発生する変更データ（差分データ）を漏れなく取得し続ければデータベースは完全性を保ちます。このための変更データの取得処理をJRA－VAN Data Lab. では**「通常データ取得」**と呼び、JV－Linkは通常データ取得機能を提供します。

「セットアップ」処理と「通常データ取得」処理によるデータベース更新を下図に示します。

【蓄積系ソフトが1991年以降を指定してセットアップを行なった場合】

※T1がセットアップを行なったタイミングです。



セットアップによる取り込みは、過去のある時点から現在までの完全なデータを取り込むため、対象ファイルが多くなり処理が遅くなる場合がございます。JRA－VAN Data Lab. では過去のある時点から現在までのデータを取り込む方法と、過去のある時点からある時点までのデータを取り込む方法を提供します。
期間の詳細な指定方法については「JV－Linkインターフェース仕様書」を参照して下さい。

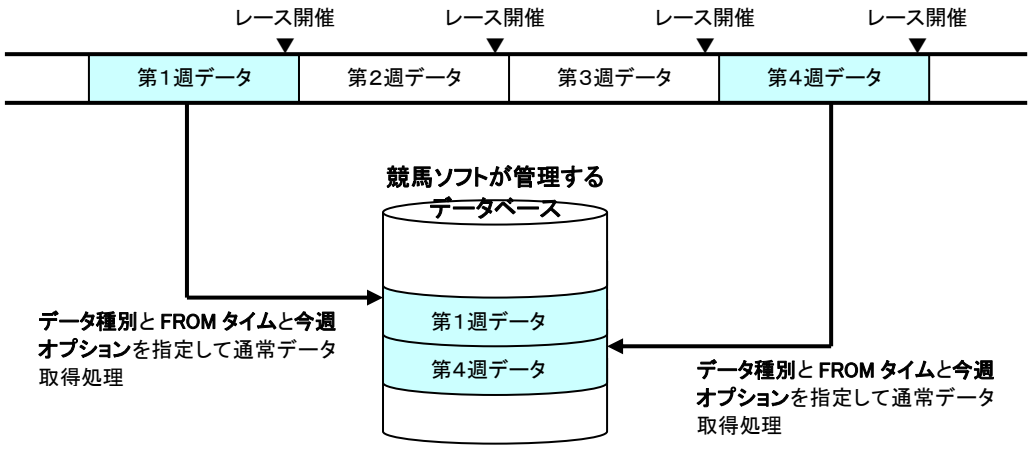
３．３．３ 非蓄積系ソフトのデータ取得の方法

非蓄積系ソフトは基本的に今週開催されるレースのデータに対して処理を行ないます。過去に行なった処理を振り返る（例えば過去に行なったレース予想を再表示する）といった機能を持たなければ、過去に取り込んだデータを必ずしも保持する必要はありません。ソフトを起動するごとにその時点で次に開催されるレースに関する必要なデータを取り込めばデータベースとしては充分ということになります。JRA－VAN Data Lab. はその週のレースに関して処理を行なうために必要と思われるデータのセットを用意しています。

したがって「非蓄積系ソフト」は蓄積系ソフトと違いセットアップ処理を行なう必要がありません。JV－Linkに用意された**「通常データ取得」**機能の**「当週のみ」**のオプションを使って断片的なデータを取得することになります。ただし JRA-VAN ではその週に開催されるレースに関する情報についてもその週の中で更新されますので、最新の状態に保つにはやはり更新データを取得する必要があります。

非蓄積系ソフトのデータ取得を下図に示します。

【レース予想ソフトが第１週と第４週に処理を行なった場合】

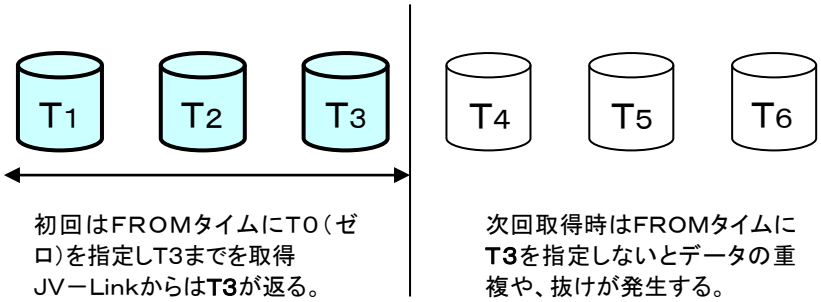


３．３．４ 競馬ソフトによる FROM タイムの管理

JV－Linkは呼び出される際に「FROM タイム」を受け取ることを想定しています。JV－Linkが「FROM タイム」を受け取るとその時刻から現在時刻、または指定した時刻までに発生したデータを競馬ソフトに返します。したがってデータの連続性を保つためには競馬ソフトがJV－Linkに渡す「FROM タイム」は前回取得したデータの最終時刻である必要があります。JV－Linkは競馬ソフトにデータを渡す際にデータの最終時刻（次回「FROM タイム」として使用するべき時刻）を同時に返しますので、競馬ソフトはこの時刻を次回取り込み処理まで保存しておく必要があります。

【蓄積系ソフト通常データ取得における FROM タイム指定】

蓄積系ソフトはデータの連続性が重要なので FROM タイムを確実に管理する必要があります。

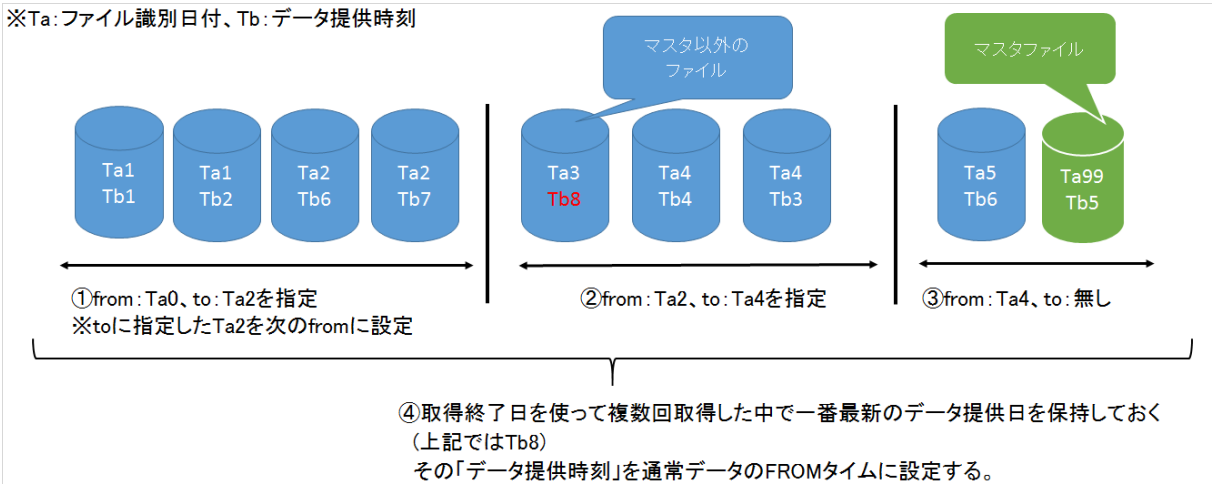


蓄積系ソフトセットアップ時のFROMタイムについて

通常データ取得の場合のFROMタイムは、「データの連続性を保つ」ことを目的としたものです。これに対してセットアップ時のFROMタイムは、「指定年以降のデータを取得する」ことが目的となります。そのため、セットアップ時のFROMタイムは、レースの開催日などが対象となります。例えば 1991 年以降のレースデータを取得対象とする場合、セットアップのFROMタイムには” 19910101000000”を指定することになります。この場合でも競走馬や騎手などのマスタ情報は全データが取得対象となります。

【蓄積系ソフトセットアップデータ取得における FROM タイム指定】

セットアップ時の FROM タイムは、レースの開催日などが対象となるため、過去のある時点からある時点までのデータを取り込む場合、データの重複や抜けが発生します。
通常データ取得時と同様に FROM タイムに指定する時刻を確実に管理する必要があります。



- JV-Link から返却される最新のデータ提供時刻ではなく、取得終了日に設定した日時を次の FROM タイムに設定する。（上記①、②）
- セットアップデータ取得の最後は必ず取得終了日なしで取得を行う。（上記③）
- セットアップデータで取得終了日を使って複数回取得を行う際は、通常データ取得時に使用するために、JV-Link から返却される最新のデータ提供時刻を保持しておく。（上記④）

【非蓄積系ソフトの今週データ取得における FROM タイム指定】

非蓄積系ソフトは長期においてのデータの連続性は不要ですが、今週のデータの中では連続性を保つ必要があるので FROM タイムの管理は蓄積系ソフトと同じです。

ある週に開催されるレースに関係したデータは下の表のようなサイクルで提供され、約1週間で完結します。

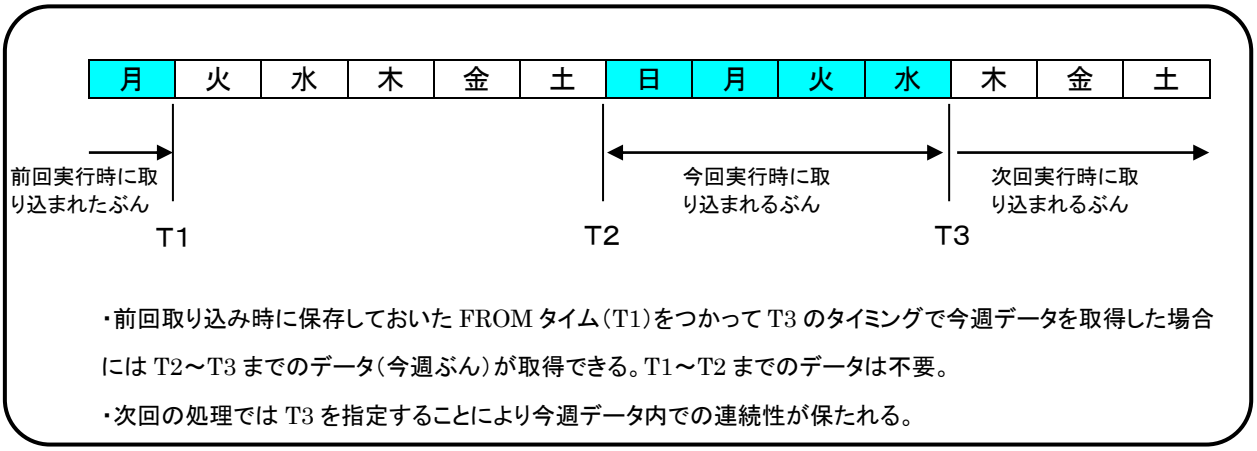
今週データとはこの1サイクル（約1週ぶん）のデータを示します。ただし日曜日と月曜日は次の週のサイクルと重なるので今週データ取得では2週ぶんのデータを読むことになります。

曜日	提供データ	提供データ内容
日曜	特別登録馬データ	次の土日の特別レースに関する情報と特別レースに登録されている馬の情報が提供されます。（ハンデなし）※1
月曜	特別登録馬関連データ	次の土日の特別レースに関する情報と特別レースに登録されている馬の情報が提供されます。（ハンデあり）※1 また、登録されている馬に関する補てん情報も同時に提供されます。
火曜		
水曜		
木曜	出走馬名表関連データ	次の土日の全レースに関する情報とレースごとに出走の決まった馬の情報が提供されます。この時点では馬番および枠番が未定の状態です。※2 また、出走する馬に関する補てん情報も同時に提供されます。
金曜	出馬表関連データ	次の土日の全レースに関する情報と土曜日の馬番、枠番付きの出走馬情報、および日曜日の馬番、枠番無しの出走馬情報が提供されます。※2
土曜 開催日	出馬表関連データ	次の日曜日の全レースに関する情報と日曜日の馬番、枠番付きの出走馬情報が提供されます。
日曜 開催日		
月曜	成績関連データ	レースの結果に関する情報が提供されます。

注：上の表は一般的なモデルですが、レース開催日が土曜、日曜以外の場合は例外となります。

※1：翌々週の G1 レースの情報が含まれている場合があります。

※2：G1 レースの馬番、枠番だけ他のレースより早く提供される場合があります。



・前回取り込み時に保存しておいた FROM タイム（T1）をつかって T3 のタイミングで今週データを取得した場合には T2～T3 までのデータ（今週ぶん）が取得できる。T1～T2 までのデータは不要。

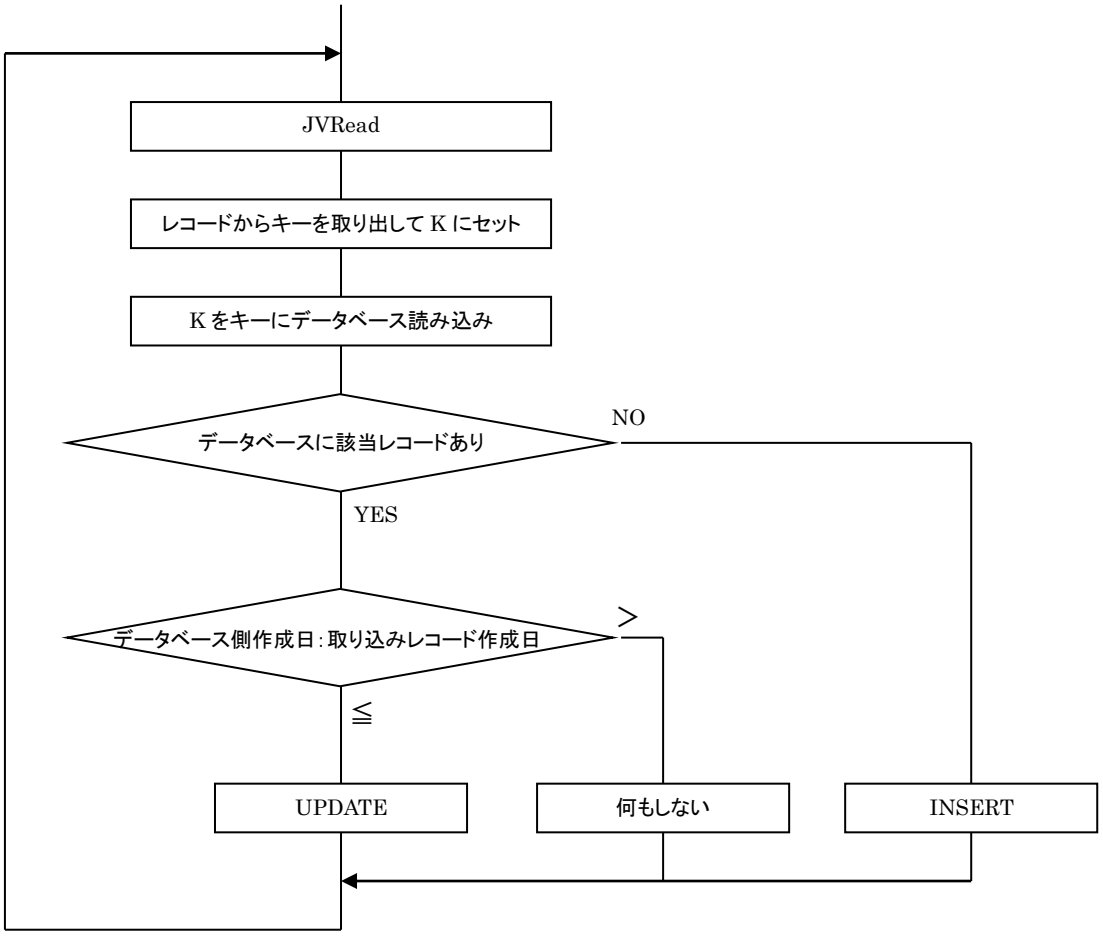
・次回の処理では T3 を指定することにより今週データ内での連続性が保たれる。

3．3．5 データベースへの取り込み

JV－Linkは複数のデータファイルをまとめて競馬ソフトに提供しますが、その読み込み順はレコード単位に完全な時系列順を保証していません。原則としてJV－Linkはレコード種別ごとに提供日順にファイルを読み出しますが、データの作成時刻順に並んでいるとはかぎりません。

したがって競馬ソフトは読み込んだ各レコードが持つ作成日付がデータベースに既に存在するレコードの作成日付と比べて新しい場合には更新を、古い場合にはレコードを無視、データベースに該当のレコードが存在しない場合には挿入をといった考慮が必要です。

以下に簡単なフローチャートを示します。(EOF やデータの終わりを考慮していません)

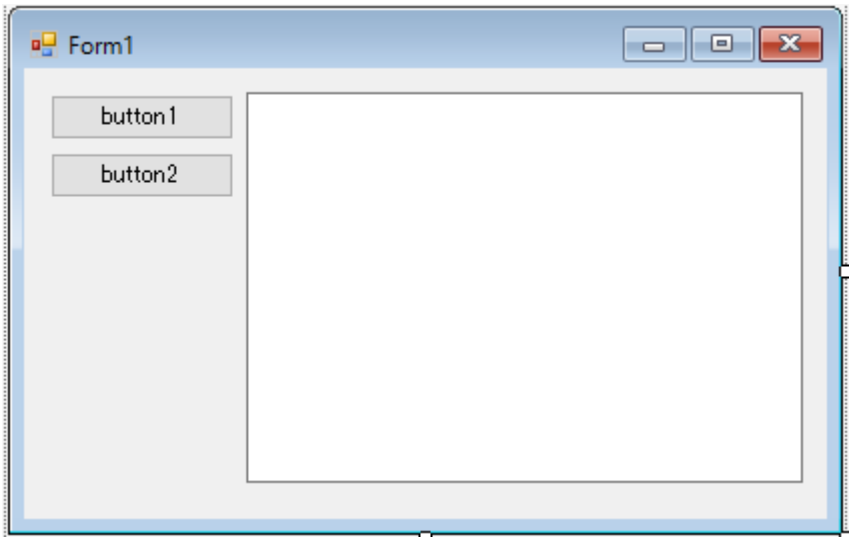


3．4 JV－Linkを使ったプログラミング1（通常データ）

JV－Linkを使ってJV-Dataを取り込む方法について具体的な例を示すことによってJV－Linkの使い方を解説していきたいと思います。ここでは VisualStudio2019 を例に順を追ってプログラミング方法を説明していきます。JV－Linkのインターフェースの詳細については触れませんので、インターフェースの詳細については**「JV－Linkインターフェース仕様書」**を参照して下さい。

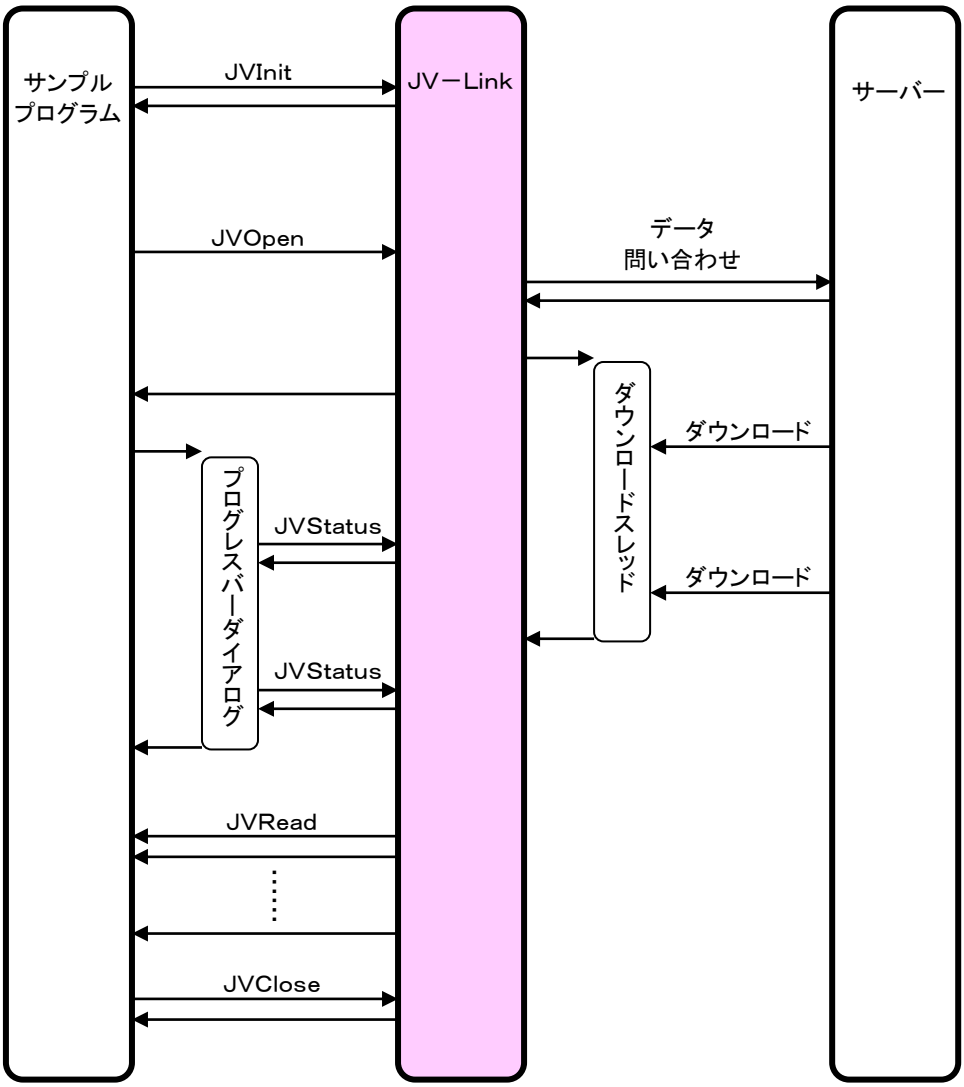
3．4．1 プログラミングのゴール

ここではJV－Linkを使って今週分のレース情報を取得して表示する簡単なプログラム（JVSample1）を作成します。



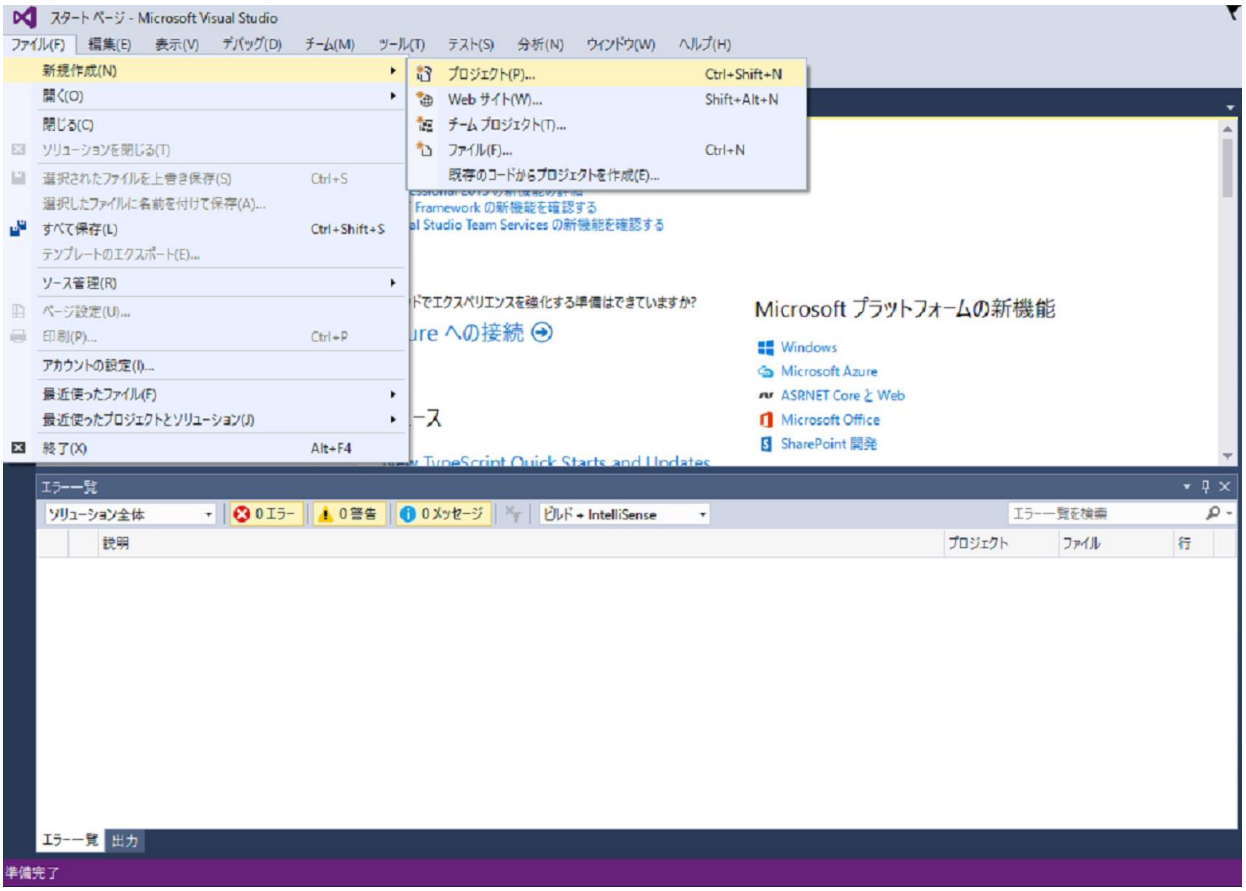
このプログラムはボタンが2つとテキストボックスが一つのシンプルな画面を持っています。「設定」ボタンを押すとJV－Linkの**JVSetUIProperties**インターフェースを使ってJV－Linkの各種設定を行なうことができます。「今週開催レース」ボタンを押すとJV－Linkの一連のインターフェースを使って今週ぶんの出馬表データを取得し、テキストボックスに内容を表示します。また、データのダウンロード中にはプログレスバー付きダイアログも表示します。

「今週開催レース」ボタンを押した際にプログラム、JV－Link、サーバー間で行なわれる処理は以下のようになります。



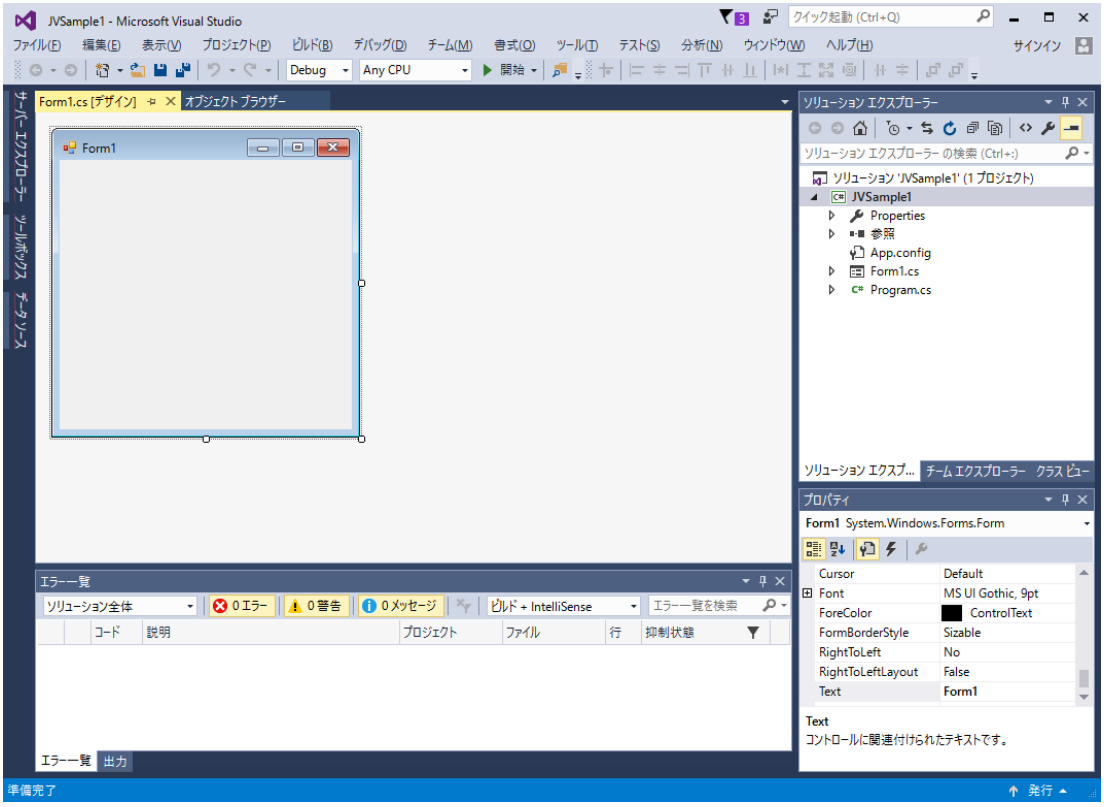
3．4．2 プロジェクトの作成

まず VisualStudio2019 を起動し、スタートページで「ファイル」タブの「新規作成」⇒「プロジェクト」ボタンをクリックします。



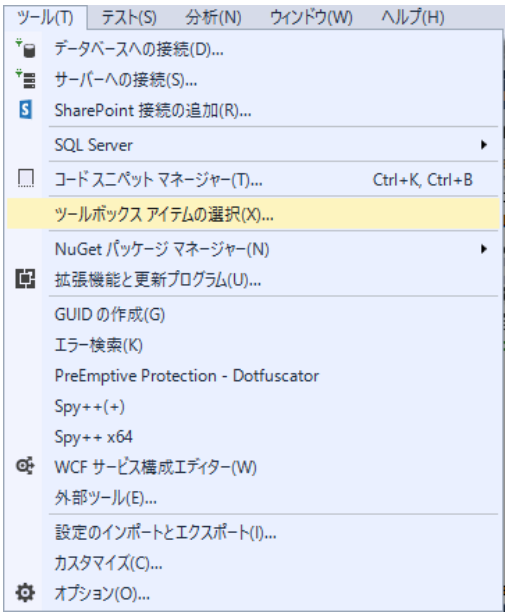
次にプロジェクトのテンプレートから「Windows フォーム アプリケーション (.NET Framework)」を選択して「次へ」ボタンをクリックします。

新しいプロジェクトが作成され下の画面になります。

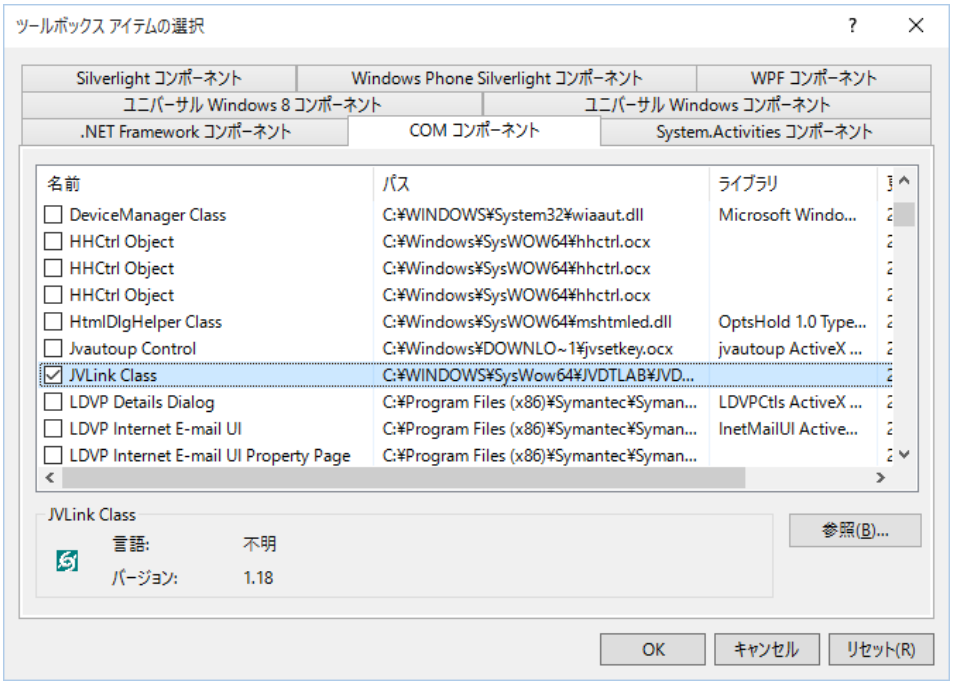


3．4．3 JVLinkコントロールの追加

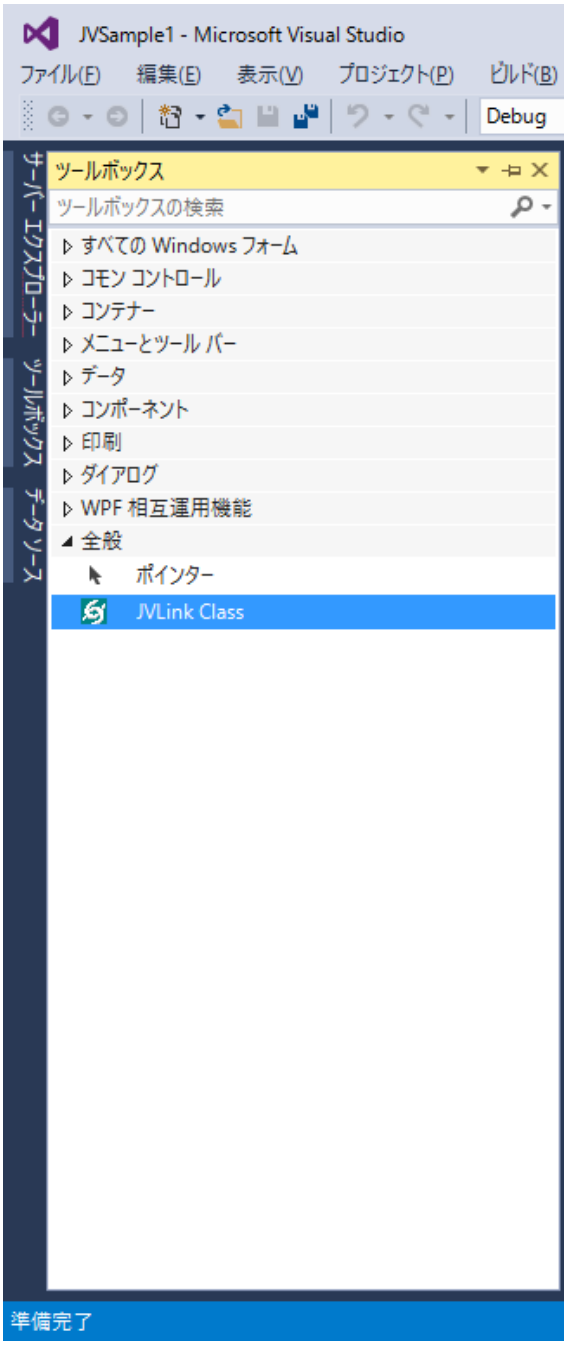
この開発環境(VisualStudio 2019)でJV－Linkを使用できるように設定を行なう必要があります。
メニューバーの**ツール(T)→ツールボックスアイテムの選択(X)…**を選択する。



ツールボックスのアイテム選択ダイアログで**JVLink Class**にチェックを付け、OKボタンをクリック。



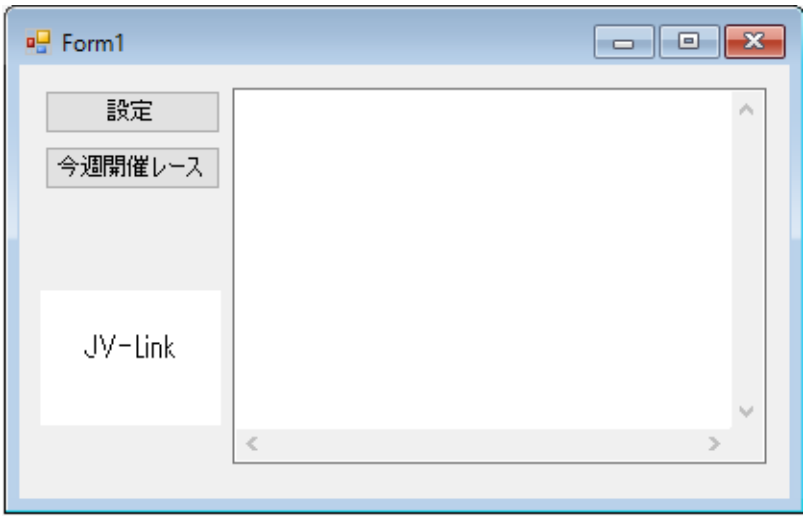
これによりツールボックスにJVLinkコントロールが追加されます。



３．４．４ フォームの作成

下図のようにボタン2個とテキストボックス、およびJVLinkコントロールを貼り付け、各プロパティーを設定する。

※このとき JV-Link のコントロールが VisualStudio の背後に隠れてしまう事象がありますが JV-Link の使用には問題ありません。



コントロール	プロパティー	選択／設定
Button1	Text	“設定“
Button2	Text	“今週開催レース”
TextBox1	Text ScrollBars Multiline Wordwrap	空白 Both True False
AxJVLink1	なし	なし

３．４．５ 設定ボタンのコーディング（JVSetUIProperties）

JVLinkコントロールには動作に関するいくつかのプロパティーを持っています。例えば作業用ディスク領域の場所やJRA－VANサーバーとの通信の際にプロキシを使用するかどうかなどです。これらの設定は **JVSetProperties** インターフェースを呼び出すことによって設定する事が可能ですが、呼出側のプログラムで設定用のユーザーインターフェースを用意する必要があります。それに対し **JVSetUIProperties** インターフェースを使用するとJVLinkが持つ設定用のダイアログを利用することができます。

JVSample1では設定ボタンをクリックすることによってこの **JVSetUIProperties** インターフェースを呼び出し、JVLinkコントロールが持つ設定ダイアログを利用することにします。

Button1 の Click イベントハンドラに以下のコードを記述します。

サンプルソース1

```
'------------------------------------------------------------------
' 設定ボタンクリック時の処理
'------------------------------------------------------------------
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    'JVSetUIPropertiesの呼出し
    If (Me.AxJVLink1.JVSetUIProperties() = -100) Then ―――――――――――――――― ①
        'レジストリへの登録に失敗すると-100が返る
        MsgBox("エラーのためＪＶ－Ｌｉｎｋの設定に失敗しました。", MsgBoxStyle.OKOnly)
    End If
End Sub
```

サンプルソース１解説

①JVSetUIPropertiesを呼び出した結果が-100 の場合は設定が何らかのエラーでレジストリに反映できなかったことを示します。

３．４．６ 今週開催レースボタンのコーディング

今週開催レースボタンが押された場合に以下の処理を行います。これがデータ読み出し処理の一般的な流れになります。

処理	内容
①JVLinkの初期化（**JVInit**）	JVLinkの初期化を行います。JVOpenを呼び出す前に最低１回呼び出す必要があります。
②JVDataのオープン（**JVOpen**）	要求に応じたデータを読み出せるように準備します。必要であればサーバーからデータをダウンロードする処理を起動します。JVOpenからの戻りとしてダウンロードを行なう予定ファイル数が返されます。
③ダウンロード中のプログレスバー表示（**JVStatus**）	JVOpenはダウンロードスレッドを起動するとダウンロード完了前に呼び出し側に制御を返しますので、呼び出し側はJVStatusを使ってダウンロード処理の進行状況を監視する必要があります。JVStatusはダウンロード済みのファイル数を返しますので、JVOpen時のダウンロード予定ファイル数と一致した場合にダウンロード処理の完了とします。
④JVDataの読み出し（**JVRead**）	データを読み出します。ダウンロードスレッドがダウンロード実行中にJVReadを呼び出すとエラーとなります。
⑤データ内容をテキストボックスに表示	一般の競馬ソフトではこの処理の代わりにデータベースへデータを反映する処理が行なわれます。
⑥JVDataのクローズ（**JVClose**）	JVOpenで確保されたリソースを開放します。

Button2 の Click イベントハンドラに以下のコードを記述します。

サンプルソース2

```
'------------------------------------------------------------------
' 今週開催レースボタンクリック時の処理
'------------------------------------------------------------------
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click

    Dim ReturnCode As Long                  'JVLinkからの戻り値
    Dim Data_Spec As String                 'JVOpen データ種別
    Dim From_Time As String                 'JVOpen From時刻
    Dim Option_Flag As Integer              'JVOpen オプション
    Dim ReadCount As Integer                'データファイルの総読み込み数
    Dim DownloadCount As Integer            'データファイルのダウンロード数
    Dim LastTime As String                  'データファイルの最終時刻
    Dim DownloadDialog As Form2             'ダウンロードダイアログ

    '************
    ' JVInit処理
    '************
    ReturnCode = Me.AxJVLink1.JVInit("UNKNOWN")  ―――――――――――― ①
```

```
If (ReturnCode <> 0) Then
    MsgBox("JVInitエラー。RC=" & CStr(ReturnCode), MsgBoxStyle.OKOnly)
End If

'*********************
' JVOpen処理
'*********************
Data_Spec = "RACE"                  'データ種別に「レース情報」を設定
From_Time = "20030101000000"        'Fromタイムに2003年1月1日を設定
Option_Flag = 2                     'オプションに「今週データ」を設定
ReturnCode = Me.AxJVLink1.JVOpen(Data_Spec, From_Time, Option_Flag, ReadCount, DownloadCount, LastTime)

' JVOpenエラー処理
If (ReturnCode < 0) Then
    MsgBox("JVOpenエラー。RC=" & CStr(ReturnCode), MsgBoxStyle.OKOnly)
    GoTo Button2_END
End If

' 該当データ無し
If (ReadCount = 0) Then
    MsgBox("該当するデータがありません。", MsgBoxStyle.OKOnly)
    GoTo Button2_END
End If

If (DownloadCount > 0) Then

    '*******************************
    ' ダウンロードプログレスバー表示
    '*******************************
    DownloadDialog = New Form2()
    'Form2からJVStatusを呼ぶためのJVLinkの参照をセット
    DownloadDialog.JVLink = Me.AxJVLink1
    DownloadDialog.DownloadCount = DownloadCount
    DownloadDialog.ShowDialog()

    ' ダウンロード処理がキャンセルかエラーで終了した場合
    If (DownloadDialog.DialogResult <> DialogResult.OK) Then
        GoTo Button2_END
    End If

End If

'*********************
' JVRead処理
'*********************
Dim RecordSpec As String            ' JV-Dataレコード種別ID
Dim Race As JV_RA_RACE              ' JV-Dataレース詳細レコード構造体

Dim buff As String
Dim buffSize As Integer = 1500
Dim fName As String

'バッファ領域確保
buff = New String(vbNullChar, buffSize)

'該当のデータ（レース情報）が1ファイル以上の場合に読み込み
If (ReadCount > 0) Then
    Do
        ReturnCode = Me.AxJVLink1.JVRead(buff, buffSize, fName)
        Select Case ReturnCode
            Case Is > 0 '正常に1レコード読み込み
                'レコード共通ヘッダ構造体にデータをセット
                RecordSpec = Mid(buff , 1 , 2)
                If ( RecordSpec= "RA") Then
                    'レース詳細レコード構造体にデータをセット
                    Race. SetData (buff)
```

②
③
④
⑤
⑥
⑦
⑧
⑨

```
                        Me.TextBox1.AppendText(Race.id.Year + Race.id.MonthDay + " " + _
                                               Race.id.JyoCD  + " " + _
                                               Race.id.Kaiji + " " + _
                                               Race.id.Nichiji + " " + _
                                               Race.id.RaceNum + " " + _
                                               Race.RaceInfo.hondai + _
                                               ControlChars.CrLf)
                    Else
                        '「レース詳細」以外は読み飛ばし
                    End If

                Case -1 'ファイルの切れ目
                    'ファイルの切り替わり時にデータは返されない

                Case 0 '全レコード読み込み終了(EOF)
                    Exit Do

                Case Else '読み込みエラー
                    MsgBox("JVReadエラー。 RC=" + CStr(ReturnCode))
                    Exit Do
            End Select
        Loop
    End If

Button2_END:
    '***************
    ' JVClose処理
    '***************
    If (DownloadCount > 0) Then
        DownloadDialog.Dispose()
        DownloadDialog = Nothing
    End If
    ReturnCode = Me.AxJVLink1.JVClose()

End Sub
```

サンプルソース2解説

①JVInitにはアプリケーションIDを渡します。このアプリケーションIDはサーバーとの通信の際にHTTPヘッダーのUser－Agentとして使用されます。ソフトの名前とバージョン番号を組み合わせてユニークなIDを指定して下さい。開発途中でアプリケーションIDを決められない場合には“UNKNOWN”を指定して下さい。

②今週ぶんだけを取得するためオプションに 2 をセットします。

③JVOpenにより返された DownloadCount が0である場合はダウンロードが行なわれないことを示していますのですぐに読み込み処理を開始することができます。DownloadCount が0より大きい場合はダウンロードのプログレスバー表示のためのダイアログを表示します。

④ダイアログ内からJVStatusを呼ぶためにJVLinkへの参照をダイアログが持つ変数に渡します。

⑤ダイアログ内でダウンロードの完了を検査するために DounloadCount(ダウンロード予定ファイル数)を必要とするので、ダイアログが持つ変数に DownloadCount を渡します。

⑥データを読み込むためのバッファサイズは該当データのレコード長より大きい値を指定する必要があります。レコード長より小さい値を指定するとバッファに入りきれなかったぶんが切り捨てられます。

⑦JVOpenにより返された ReadCount が0である場合は読み込む対象のデータが無いことを示していますので読み込み処理をスキップします。

⑧JVReadの戻り値により処理を分岐しハンドリングする必要があります。戻り値が0より大きい場合は正常に読み込み

が行なわれています。サンプルプログラムではテキストボックスへの表示を行なっていますが、通常の競馬ソフトはデータベースへの取り込み処理を行います。読み込んだデータのデータ種別を判断するにはレコードの先頭にあるデータ種別フィールドを参照する必要があります。戻り値が－1の場合にはファイルの切れ目を示しています。バッファの中にはデータが格納されていませんので何も処理する必要がありません。あえて実施するなら読み込み済みのファイル数をこのタイミングでカウントし、ReadCount(読み込み予定ファイル数)との比較で読み込み処理の進捗状況を表示するすることが可能です。戻り値が0の場合は全てのファイルを読み終わったことを示しています。

⑨読み込んだレコードのヘッダ部分にあるレコード種別を判定し、処理を分岐する必要があります。サンプルプログラムではデータ種別「レース情報」(”RACE”)の中のレコード種別「レース詳細」(“RA”)だけを読むので、それ以外のレコードを読み飛ばしています。(データ種別およびレコード種別については「JV－Data仕様書」を参照して下さい)

※JV-Data レース詳細レコード構造体について

レース詳細レコード構造体にデータをセットするためには、SDK に含まれている JV-Data 構造体の「JVData_Structure.vb」をプロジェクトファイルに組み込む必要があります。

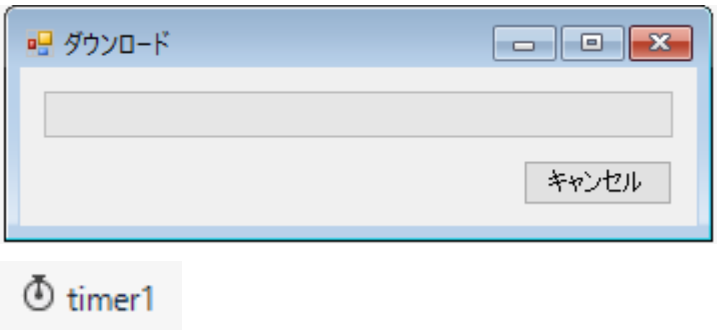
1. Visual Studio でプロジェクトを開いた状態で、メニューバーのファイル(F)→既存項目の追加(G)...を選択する。
2. 「JRA-VAN SDK」を解凍したディレクトリの下の「¥JV-Data 構造体¥VB.Net 版」にある「JVData_Structure.vb」を選択して、開く(O)をクリック。

上記の手順で、ソリューションエクスプローラーに「JVData_Structure.vb」が追加されます。

3. 4. 7 ダウンロード中プログレスバー表示ダイアログのコーディング

データを読み出すためにJVOpenインターフェースを呼び出すと必要なデータをサーバーからダウンロードする処理（別スレッド）がスタートします。ダウンロードスレッドがスタートすると呼出側に制御が戻りますがダウンロード処理が完了するまでJVReadインターフェースでのデータの読み出しは出来ません。したがって呼出側はダウンロードが完了するのを待つ必要があります。通常ダウンロード処理には時間がかかりますので、そのあいだ進行状況を表示するのが望ましいと思われます。サンプルソース3ではその進行状況をプログレスバーを使って表示します。

まずフォーム(Form2)を作成します。タイマー(Timer1)を貼り付けるのを忘れないでください。



コントロール	プロパティー	選択／設定
Form2	Text	“ダウンロード”
Button1	Text	“キャンセル“
Label1	Text	“”
ProgressBar1	なし	なし
Timer1	なし	なし

フォームの Load イベント、タイマーの Tick イベント、キャンセルボタンの Click イベントのハンドラに以下のコードを記述します。

サンプルソース3

```
Public JVLink As AxJVDTLabLib.AxJVLink ―――――――――――――― ①
Public DownloadCount As Integer

'---------------------------------------------------------------
' 初期処理
'---------------------------------------------------------------
Private Sub Form2_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

    Me.ProgressBar1.Maximum = DownloadCount    'プログレスバー最大値 ―――――――― ②
    Me.ProgressBar1.Minimum = 0                'プログレスバー最小値
    Me.ProgressBar1.Value = 0                  'プログレスバー初期値
```

```
        Me.Timer1.Interval = 10             'タイマー間隔
        Me.Timer1.Start()                   'タイマー始動

        Me.StartPosition = FormStartPosition.CenterParent

    End Sub
    '---------------------------------------------------------------------
    ' タイマー毎の処理
    '---------------------------------------------------------------------
    Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Timer1.Tick

        Dim Status As Integer           '進捗度合い
        Dim Count As Integer            'JVStatusの戻り値

        '******************
        ' JVStatus呼出
        '******************
        Count = Me.JVLink.JVStatus()  ―――――――――――――――――――― ③

        'エラー判定
        If (Count < 0) Then
            Timer1.Stop()
            MsgBox("ダウンロード失敗。RC=" & CStr(Count), MsgBoxStyle.OKOnly)
            Me.DialogResult = DialogResult.Abort
            '終了処理
            Me.Close()
        Else
            'プログレスバーを進める
            Me.ProgressBar1.Value = Count
            Me.Label1.Text = Count & " / " & DownloadCount
            '全てのデータをダウンロードしたら終了
            If (Count = DownloadCount) Then
                Me.DialogResult = DialogResult.OK
            End If
        End If

    End Sub

    '---------------------------------------------------------------------
    ' キャンセル処理
    '---------------------------------------------------------------------
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click

        '******************
        ' JVCancel
        '******************
        Me.JVLink.JVCancel()  ―――――――――――――――――――― ④
        Me.DialogResult = DialogResult.Cancel

        Timer1.Stop()
        Me.Close()
    End Sub
```

サンプルソース3解説

①呼び出し元フォームが持っているJVLinkへの参照および DownloadCount を受け取る変数。

②初期処理（フォームの Load イベント）ではプログレスバーを初期化し、タイマーを起動します。

③JVStatusインターフェースはダウンロード処理の進行状況をダウンロードしたファイルの数で返しますので、このインターフェースを使ったプログレスバー表示を行います。JVOpenから返されるダウンロード予定ファイル数とJVStatusが返すダウンロード済みファイル数を使って進捗状況のパーセンテージを求めます。JVStatusの戻り値が

DownloadCount と一致した場合ダウンロード処理の完了を意味し、戻り値が負の値の場合は何らかのエラーが発生したことを意味します。

④ダウンロード中にJVCancelインターフェースを呼び出すとダウンロード処理を中止することができます。

3．4．9 ビルド＆実行

メニューバーの**ビルド(B)→ソリューションのビルド(B)**を選択する。

以上で JVSample1 の開発は終了です。ビルドによって作成された.exe ファイルをダブルクリックするかデバッグコマンドにより実行が可能です。

APPENDIX

A. サンプルプログラム1

JV－Linkを利用したサンプルプログラムをご紹介します。
サンプルプログラム1はJV－Link が提供する基本機能を、より一層ご理解頂くためのプログラムです。

> **NOTE**
> サンプルプログラム1を実際のソフト製作にそのまま流用されることはお薦めしません。サンプルプログラム1はあくまでJV－Linkの各インターフェースの呼び出し方の学習、あるいは動作検証用途としてご利用下さい。
> 例えば、サンプルプログラム1では、JVOpen を呼び出す際のパラメータを、エンドユーザーが指定することになっていますが、Data Lab.では極力エンドユーザーの判断無しでデータ取り込み処理を行なえるようなソフトが理想的なソフトと考えています。

A. 1 サンプルプログラム1の構成

サンプルプログラム1は以下の開発環境でご利用できます。

- 『Microsoft Visual Basic 2019』
- 『Microsoft Visual C++ 2019』
- 『Borland Delphi 6.0 / 7.0 版共通』

サンプルプログラム1は「JRA－VAN Data Lab. SDK」に含まれます。「JRA－VAN Data Lab. SDK」を解凍した後、「サンプルプログラム」フォルダを参照して下さい。フォルダ内のファイル構成を下に表します。当ガイドには Microsoft Visual Basic .net 版のソースのみを記載しています。

また、サンプルプログラムをビルドする前にJV－Linkがインストールされていることを確認して下さい。
※ JV－Linkの使用については当ガイドと共にJRA－VAN Data Lab. SDK に収録されている README.TXT も同時に参照下さい。

① 『Microsoft Visual Basic 2019 版』

```
/sample1_VB2019/---|- JVLinkSample1.sln          （ソリューションファイル）
                   |- JVLinkSample1.vbproj       （プロジェクトファイル）
                   |- AssemblyInfo.vb            （アセンブリに関する一般情報）
                   |- Form1.vb          メイン画面（ソース）
                   |- Form1.resx        メイン画面（テンプレート）
                   |- Form2.vb          データ取込み画面（ソース）
                   |- Form2.resx        データ取込み画面（テンプレート）
                   |- app.config        （アプリケーション構成ファイル）
```

② 『Microsoft Visual C++ 2019 版』

```
/sample1_VC2019/--|- sample1.sln          （ソリューションファイル）
                  |- sample1.vcxproj       （プロジェクトファイル）
                  |- sample1.vcxproj.filters       （フィルターファイル）
                  |- jvlink.h        （ActiveX コントロールのラッパークラス）
                  |- jvlink.cpp      （ActiveX コントロールのラッパークラス）
                  |- ReadMe.txt      （各ファイルの概要説明）
                  |- resource.h      （リソースファイル）
                  |- sample1.h       （インクルードファイル）
                  |- sample1.cpp     （ソースファイル）
                  |- sample1.dsp     （ビルド用プロジェクトファイル）
                  |- sample1.dsw     （ワークスペースファイル）
                  |- sample1.rc      （Windows リソースファイル）
                  |- sample1.ncb     （パーサー情報ファイル）
                  |- sample1Del.h    ファイル削除画面（インクルードファイル）
                  |- sample1Del.cpp     ファイル削除画面s（ソースファイル）
                  |- sample1Dlg1.h     メイン画面（インクルードファイル）
                  |- sample1Dlg1.cpp    メイン画面（ソースファイル）
                  |- sample1Dlg2.h     データ取込み画面（インクルードファイル）
                  |- sample1Dlg2.cpp     データ取込み画面（ソースファイル）
```

③『Borland Delphi 6.0 / 7.0 版共通』

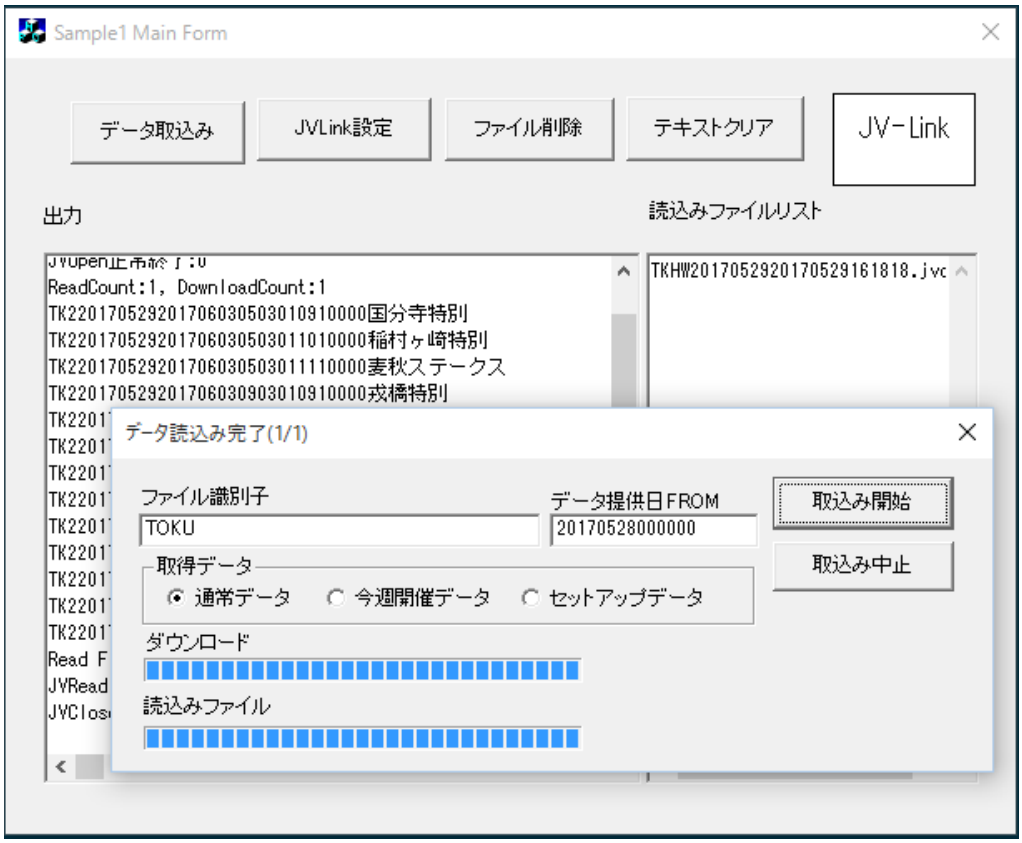
```
/sample1_DELPHI7/- |- JVLinkSample1.dpr          （プロジェクトファイル）
                  |- JVLinkSample1.Res          （リソースファイル）
                  |- Unit1.frm        （メイン画面フォーム）
                  |- Unit1.pas        （メイン画面ソース）
                  |- Unit2.frm        （データ取込み画面フォーム）
                  |- Unit2.pas        （データ取込み画面ソース）
```

A. 2 サンプルプログラム1の解説

A. 2. 1 プログラミングのゴール

サンプルプログラム1は JV-Data を取得するためのプログラムです。取得できるデータは、通常データ、今週データ、セットアップデータの 3 種類のデータで速報系データは取得できません。読み込んだデータおよび読み込んだファイルのファイル名をウィンドウに表示します。

このサンプルではJV－Linkに渡すパラメータをダイアログで指定できますのでパラメータの指定と取得されるデータの関係が良くわかるサンプルになっています。しかしながら実際のソフト開発ではパラメータの指定をエンドユーザーに行なわせるやり方は避けるべきです。



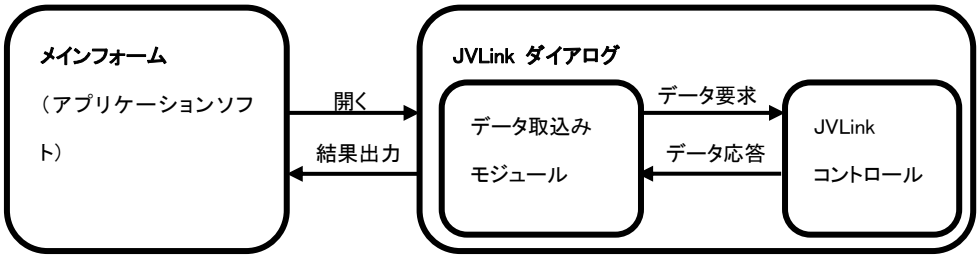
A. 2. 2 画面構成

サンプルフォームは「メイン画面」と「データ取込み画面」の2画面で構成されています。

「メイン画面」は、「データ取込み画面」の呼出しを行います。JV－Linkが取込んだデータは、この「メイン画面」に出力表示されます。

「データ取込み画面」はJV－Linkを実装し、JV－Data情報取込みまでの 一連動作を行います。

【画面構成概要図】



A. 2. 3 JV-Data 情報を取込むまでの操作手順

JV-Data 情報を取込むまでの各操作を説明します。

① JV－Linkの環境設定を行います。

サンプルプログラム1を起動すると、「メイン画面」が表示されます。はじめに、JV－Linkを設定して下さい。「メイン画面」より、「JV－Link環境設定」ボタンをクリックし、サービスキーの設定、データの保存設定フォルダの設定を行います。

② 「データ取込み画面」を開きます。

「メイン画面」より、「データ取込み」ボタンをクリックし、「データ取込み画面」を開きます。

③ パラメータを入力し、データを取込みます。

「データ取込み画面」より、JVOpenに指定するパラメータを設定します。パラメータはテキストボックス「ファイル識別子」、「データ提供日 FROM」、ラジオボタン「取得データ」に入力し指定します。

パラメータを指定したら「データ取込み開始」ボタンを押して下さい。JV－LinkがJV－Data情報を取得します。

④ 出力表示

JV－Linkの実行結果（JVLink戻り値、JV－Data情報 等）は、「メイン画面」のテキストボックス「出力」に出力表示されます。読込んだファイル名は、「読込みファイルリスト」に表示されます。

JV－Data情報のダウンロードが始まると、プログレスバー「ダウンロード」が表示されます。ダウンロードが終了すると、読込み処理が始まります。読込み処理が始まるとプログレスバー「読込み」が表示されます。途中で処理を停止したい場合は、「取込み中止」ボタンを押して下さい。

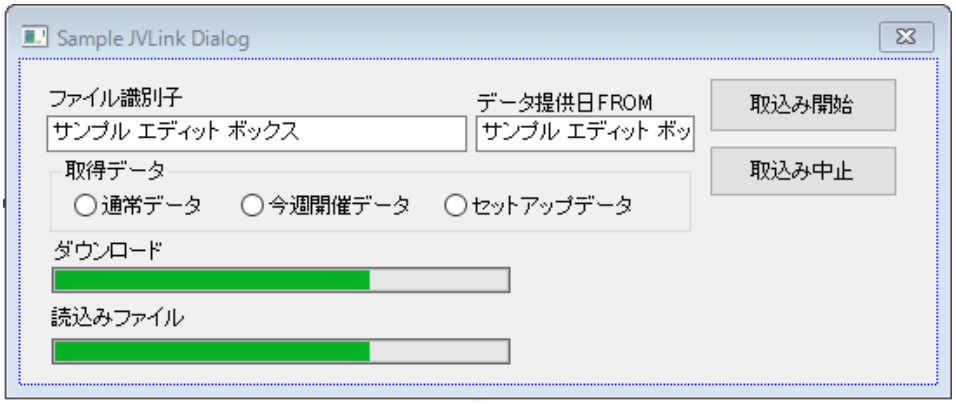
A. 2. 4 コントロールの構成

各コントロールの構成（配置）とプロパティを下に示します。各コントロールに設定されているプロパティは、下表を参照してください。

① メイン画面(Form1.frm)

コントロール	プロパティー	選択／設定
フォーム	Name Text StartPosition FormBorderStyle	**frmMain** “Sample1 Main Form” CenterScreen FixedToolWindow
コマンドボタン	Name Text	**cmdJVLinkDialog** “データ取込み”
コマンドボタン	Name Text	**cmdJVSetUIProperties** “JVLink 設定”
コマンドボタン	Name Text	**cmdDelete** “ファイル削除”
リッチテキストボックス	Name Text AutoSize Multiline ScrollBars WordWrap	**txtStatus** 空白 False True Both False
リッチテキストボックス	Name Text AutoSize Multiline ScrollBars WordWrap	**txtFileList** 空白 False True Both False
ActiveXCOM コントロール	Name	**AxJVLink1**

② JV－Linkコントロールパネル(Form2.frm)



コントロール	プロパティー	選択／設定
フォーム	Name Text StartPosition FormBorderStyle	**frmJVLinkDialog** “Sample1 JVLink Dialog” CenterScreen FixedToolWindow
コマンドボタン	Name Text	**cmdStart** “取込み開始”
コマンドボタン	Name Text	**cmdCancel** “取込み中止”
プログレスバー	Name	**ProgressBar1**
プログレスバー	Name	**ProgressBar2**

A. 2. 5 注意事項

・各コントロールの役割を明確にするために、ラベル、コマンドボタン等のコントロール名、キャプション名を デフォルトで設定される値から変更しています。

・「出力」「読込みファイルリスト」のテキスト出力に、リッチテキストコントロールを使用しています。リッチテキストボックスは、テキストボックスコントロールに比べて、最大文字数の制限に余裕があり、柔軟性に優れています。

A. 3 サンプルソース

A. 3. 1 サンプルソース『MicrosoftVisualBasic.2019 版』

サンプルプログラム１のソースコードを以下に示します。

【メイン画面： frmSample(Form1.frm)】

※Windows フォームデザイナで生成されたコード「#Region」は省略しています。

```
'==========================================================================
'  ＪＲＡ－ＶＡＮ　Ｄａｔａ　Ｌａｂ.　サンプルプログラム１(Form1.vb)
'
'
'    作成: JRA-VAN ソフトウェア工房　2003年4月22日
'
'==========================================================================
'    (C) Copyright Turf Media System Co.,Ltd. 2003 All rights reserved
'==========================================================================

Public Class frmMain
    Inherits System.Windows.Forms.Form

    '---------------------------------------------------------------------------------------------
    '     初期化
    '---------------------------------------------------------------------------------------------
    Private Sub frmMain_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) _
                                                                                Handles MyBase.Load
        Try
            Dim sid As String                       "引数 JVInit:ソフトウェアID
            Dim ReturnCode As String                "戻値

            '引数設定
            sid = "UNKNOWN"
            '***************
            'JVLink初期化
            '***************
            '※※※ JVInitは JVLinkメソッド使用前(但し、JVSetUIProPertiesを除く)に呼出す
            ReturnCode = AxJVLink1.JVInit(sid)

            'エラー判定
            If ReturnCode <> 0 Then       "エラー
                Call PrintOut("JVInitエラー:" & ReturnCode & ControlChars.CrLf)
                '終了
                Exit Sub
            Else                                "正常
                Call PrintOut("JVInit正常終了:" & ReturnCode & ControlChars.CrLf)
            End If
        Catch
            PrintOut(Err.Description)
        End Try
    End Sub

    '---------------------------------------------------------------------------------------------
    '     データ取込みボタンクリック時の処理
    '---------------------------------------------------------------------------------------------
    Private Sub cmdJVLinkDialog_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) _
                                                                    Handles cmdJVLinkDialog.Click
```

```
        Try
            'Form2：JVLinkコントロールパネルを開く
            Dim frmDialog As frmJVLinkDialog
            frmDialog = New frmJVLinkDialog()
            frmDialog.ShowDialog(Me)                "オーナーウィンドウに自画面を指定

            frmDialog.Dispose()
            frmDialog = Nothing

            '終了
            Exit Sub
        Catch
            PrintOut(Err.Description)
        End Try
    End Sub

    '-------------------------------------------------------------------------------
    '    「出力」に処理結果を表示
    '-------------------------------------------------------------------------------
    Public Sub PrintOut(ByVal Message As String)
        Try
            'txtOutに文字列を表示し、改行する
            txtOut.AppendText(Message)

        Catch
            Debug.WriteLine(Err.Description)
        End Try
    End Sub


    '-------------------------------------------------------------------------------
    '    「ファイルリスト」にダウンロードしたファイルリストを表示
    '-------------------------------------------------------------------------------
    Public Sub PrintFilelist(ByVal strMessage As String)
        Try
            'txtFileListに文字列を表示し、改行する
            txtFileList.AppendText(strMessage)
        Catch
            Debug.WriteLine(Err.Description)
        End Try
    End Sub

    '-------------------------------------------------------------------------------
    '    JVLink設定ウィンドウ表示
    '-------------------------------------------------------------------------------
    Private Sub cmdJVSetUIProperties_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) _
                                                                    Handles cmdJVSetUIProperties.Click
        Try
            Dim ReturnCode As String                "戻値

            '**********************
            'JVLink設定画面表示
            '**********************
            ReturnCode = AxJVLink1.JVSetUIProperties()

            'エラー判定
            If ReturnCode <> 0 Then             "エラー
                PrintOut("JVSetUIPropertiesエラー:" & ReturnCode & ControlChars.CrLf)
            Else                                "正常
                PrintOut("JVSetUIProperties正常終了:" & ReturnCode & ControlChars.CrLf)
            End If

            '終了
            Exit Sub
        Catch
            PrintOut(Err.Description)
```

```
        End Try
    End Sub

    '------------------------------------------------------------------------------
    '     指定したファイルを削除
    '------------------------------------------------------------------------------
    Private Sub cmdDelete_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) _
                                                                        Handles cmdDelete.Click
        Try
            Dim Message, Title, DefaultValue As String
            Dim MyValue As String
            Dim ReturnCode As String                "戻値

            '初期値設定
            Message = "ファイル名を入力して下さい"                    "メッセージ
            Title = "ファイル削除"                                   "タイトル名
            DefaultValue = ""                                       "初期値

            'ファイル名入力
            MyValue = InputBox(Message, Title, DefaultValue)

            '**********************
            'JVFileDelete
            '**********************
            ReturnCode = AxJVLink1.JVFiledelete(MyValue)

            'エラー判定
            If ReturnCode <> 0 Then             "エラー
                PrintOut("JVFiledeleteエラー:" & ReturnCode & ControlChars.CrLf)
            Else                                "正常
                PrintOut("JVFiledelete正常終了:" & ReturnCode & ControlChars.CrLf)
            End If

            '終了
            Exit Sub
        Catch
            PrintOut(Err.Description)
        End Try
    End Sub

    '------------------------------------------------------------------------------
    '     表示をクリア
    '------------------------------------------------------------------------------
    Private Sub cmdClear_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) _
                                                                        Handles cmdClear.Click
        txtOut.Text = ""
        txtFileList.Text = ""
    End Sub

End Class
```

【データ取り込み画面 ： frmJVLinkDialog(Form2.frm)】

※Windows フォームデザイナで生成されたコード #Region は省略しています。

```
'==============================================================================
'  ＪＲＡ－ＶＡＮ　Ｄａｔａ　Ｌａｂ.　サンプルプログラム１(Form2.vb)
'
'
'    作成: JRA-VAN ソフトウェア工房　2003年4月22日
'
'==============================================================================
'    (C) Copyright Turf Media System Co.,Ltd. 2003 All rights reserved
'==============================================================================
```

```
Public Class frmJVLinkDialog
    Inherits System.Windows.Forms.Form

    Private frmOwner As frmMain                    "メインフォーム
    Private CancelFlag As Boolean                  "キャンセルフラグ
    Private ReadCount As Integer                   "JVOpen:総読込みファイル数
    Private DownloadCount As Integer               "JVOpen:総ダウンロードファイル数
    Private LastFileTimeStamp As String            "JVOpen:最後にダウンロードしたファイルのタイムスタンプ


    '---------------------------------------------------------------------------------------------
    '     データ取得実行ボタンクリック時の処理
    '---------------------------------------------------------------------------------------------
    Private Sub cmdStart_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) _
                                                                                    Handles cmdStart.Click
        Try

            Dim DataSpec As String                 "引数 JVOpen:ファイル識別子
            Dim FromDate As String                 "引数 JVOpen:データ提供日付FROM
            Dim DataOption As Integer              "引数 JVOpen:オプション
            Dim ReturnCode As Integer              "JVLink戻値

            '初期値設定
            tmrJVStatus.Enabled = False            "タイマー停止
            frmOwner = Owner                          "親フォームを指定
            CancelFlag = False                     "キャンセルフラグ初期化
            progressBar1.Value = 0                 "プログレスバー初期化
            progressBar2.Value = 0

            '引数設定
            DataSpec = txtDataSpec.Text
            FromDate = txtFromDate.Text

            If rbtNormal.Checked = True Then
                DataOption = 1
            ElseIf rbtIsthisweek.Checked = True Then
                DataOption = 2
            ElseIf rbtSetup.Checked = True Then
                DataOption = 3
            End If

            Cursor = Cursors.AppStarting()

            '**********************
            'JVLinkダウンロード処理
            '**********************
            ReturnCode = frmOwner.AxJVLink1.JVOpen(DataSpec, _
                                                   FromDate, _
                                                   DataOption, _
                                                   ReadCount, _
                                                   DownloadCount, _
                                                   LastFileTimeStamp)

            'エラー判定
            If ReturnCode <> 0 Then       "エラー
                Call frmOwner.PrintOut("JVOpenエラー:" & ReturnCode & ControlChars.CrLf)
                '終了処理
                Call JVClosing()
            Else                                   "正常
                Call frmOwner.PrintOut("JVOpen正常終了:" & ReturnCode & ControlChars.CrLf)
                Call frmOwner.PrintOut("ReadCount:" & _
                                       ReadCount & _
                                       " , DownloadCount:" & _
                                       DownloadCount & _
                                       ControlChars.CrLf)
```

```
            '総ダウンロード数をチェック
            If DownloadCount = 0 Then                           "総ダウンロード数＝０
                'プログレスバー100%表示
                progressBar1.Maximum = 100                      "MAXを100に設定
                progressBar1.Value = progressBar1.Maximum       "プログレスバー100%表示
                Text = "ダウンロード完了"
                '読込み処理
                Call JVReading()
                '終了処理
                Call JVClosing()
            Else                                                "総ダウンロード数が０以上
                '初期値設定
                Text = "ダウンロード中・・・"
                progressBar1.Maximum = DownloadCount            "プログレスバーのＭＡＸ値設定
                'タイマー始動：ダウンロード進捗率をプログレスバー表示
                tmrJVStatus.Enabled = True                      "ダウンロードステータスを監視する
            End If
        End If

        '終了
        Exit Sub

    Catch
        frmOwner.PrintOut(Err.Description)
    End Try
End Sub


'-------------------------------------------------------------------------------------------
'     タイマー：ダウンロード状況をプログレスバー表示
'-------------------------------------------------------------------------------------------
Private Sub tmrJVStatus_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) _
                                                                        Handles tmrJVStatus.Tick
    Try
        Dim ReturnCode As Integer               "JVLink戻値

        '**********************
        'JVLinkダウンロード進捗率
        '**********************
        ReturnCode = frmOwner.AxJVLink1.JVStatus                    "ダウンロード済のファイル数を返す

        'エラー判定
        If ReturnCode < 0 Then                                      "エラー
            Call frmOwner.PrintOut("JVStatusエラー:" & ReturnCode)
            'タイマー停止
            tmrJVStatus.Enabled = False
            '終了処理
            Call JVClosing()
            '終了
            Exit Sub
        ElseIf ReturnCode < DownloadCount Then                      "ステータス
            'プログレス表示
            Text = "ダウンロード中...  (" & ReturnCode & "/" & DownloadCount & ")"
            progressBar1.Value = ReturnCode
        ElseIf ReturnCode = DownloadCount Then                      "ステータス100%
            'タイマー停止
            tmrJVStatus.Enabled = False
            'プログレス表示
            Text = "ダウンロード完了(" & ReturnCode & "/" & DownloadCount & ")"
            progressBar1.Value = ReturnCode
            '読込み処理
            Call JVReading()
            '終了処理
            Call JVClosing()
```

```
                '終了
                Exit Sub
            End If

        Catch
            frmOwner.PrintOut(Err.Description)
        End Try
    End Sub

    '-------------------------------------------------------------------------------
    '    読込処理
    '-------------------------------------------------------------------------------
    Public Sub JVReading()
        Try
            Dim Buff As String                          "バッファ
            Dim BuffSize As Integer                     "バッファサイズ
            Dim BuffName As String                       "バッファ名
            Dim JVReadingCount As Integer                "読込みファイル数
            Dim ReturnCode As Integer                    "JVLink戻値

            '初期値設定
            progressBar2.Maximum = ReadCount
            JVReadingCount = 0
            progressBar2.Value = 0
            Text = "データ読込み中...  (0/" & ReadCount & ")"

            'バッファ領域確保
            BuffSize = 60000
            Buff = New String(vbNullChar, BuffSize)

            Do
                'バックグラウンドでの処理
                System.Windows.Forms.Application.DoEvents()

                'キャンセルが押されたら処理を抜ける
                If CancelFlag = True Then Exit Sub

                '**********************
                'JVLink読込み処理
                '**********************
                ReturnCode = frmOwner.AxJVLink1.JVRead(Buff, BuffSize, BuffName)

                'エラー判定
                Select Case ReturnCode
                    Case Is > 0         "正常
                        Call frmOwner.PrintOut(Buff)
                    Case -1             "ファイルの切れ目
                        'ファイル名表示
                        Call frmOwner.PrintFilelist(BuffName & ControlChars.CrLf)
                        Call frmOwner.PrintOut("JVRead File :" & ReturnCode & ControlChars.CrLf)
                        'プログレスバー表示
                        JVReadingCount = JVReadingCount + 1 "カウントアップ
                        progressBar2.Value = JVReadingCount
                        Text = "データ読込み中...  (" & JVReadingCount & "/" & ReadCount & ")"
                    Case 0              "全レコード読込み終了(EOF)
                        Call frmOwner.PrintOut("JVRead EndOfFile :" & ReturnCode & ControlChars.CrLf)
                        Text = "データ読込み完了(" & JVReadingCount & "/" & ReadCount & ")"
                        '終了
                        Exit Sub
                    Case Is < -3        "エラー
                        Call frmOwner.PrintOut("JVReadエラー:" & ReturnCode & ControlChars.CrLf)
                        '終了
                        Exit Sub
                End Select

            Loop
```

```
        Catch
            frmOwner.PrintOut(Err.Description)
        End Try
    End Sub

    '-------------------------------------------------------------------------------------------
    '     キャンセルボタンクリック時の処理
    '-------------------------------------------------------------------------------------------
    Private Sub cmdCancel_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) _
                                                                                Handles cmdCancel.Click
        Try
            'タイマー停止
            tmrJVStatus.Enabled = False

            '***************
            'JVLink中止処理
            '***************
            frmOwner.AxJVLink1.JVCancel()

            'キャンセルフラグをたてる
            CancelFlag = True

            Call frmOwner.PrintOut("JVCancel:キャンセルされました" & ControlChars.CrLf)
            Text = "JVCancel:キャンセルされました"

            '終了
            Exit Sub
        Catch
            frmOwner.PrintOut(Err.Description)
        End Try
    End Sub

    '-------------------------------------------------------------------------------------------
    '     終了処理
    '-------------------------------------------------------------------------------------------
    Private Sub JVClosing()
        Try
            Dim ReturnCode As Integer               "JVLink戻値

            '***************
            'JVLink終了処理
            '***************
            ReturnCode = frmOwner.AxJVLink1.JVClose()

            Cursor = Cursors.Default()

            If ReturnCode <> 0 Then             "エラー
                Call frmOwner.PrintOut("JVCloseエラー:" & CStr(ReturnCode) & ControlChars.CrLf)
            Else                                "正常
                Call frmOwner.PrintOut("JVClose正常終了:" & CStr(ReturnCode) & ControlChars.CrLf)
            End If

            '終了
            Exit Sub
        Catch
            frmOwner.PrintOut(Err.Description)
        End Try
    End Sub

End Class
```