

JRA-VAN Data Lab. SDK  
JV-Link  
インターフェース仕様書

2024年8月7日  
第4.9.0.1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
2024/08/07	4.9.0.1	2	修正	・JVRead/JVGets の中断・再開についての記述箇所を lastfiletimestamp の説明の外に記述するよう修正
2023/8/8	4.9.0	2	修正	・JVOpen の option パラメータとdataspec の指定不可に関する組合せ表を修正 ・JVOpen の読み出し終了ポイント時刻を指定することができないパラメータに関する記述を修正
2022/12/5	4.8.0.1	2	修正	・JVMVPlay/JVMVPlayWithTypeの【補足】のパトロール映像に関する記述を修正
2022/6/21	4.8.0	2	修正	・JVSetUIProperties のダイアログを変更
		2	修正	・JVOpen の option と dataspec の指定誤りを修正
		2	修正	・メソッド一覧の JVEvtPay を JVEvtPay に変更
2022/4/5	4.7.1	2	修正	・メソッド一覧の JVSavFlg を JVSavFlag に変更
		2	修正	・JVOpen メソッドの【パラメータ】に読み出し終了ポイント時刻指定に関する説明を追加
		2	修正	・JVFileDelete メソッドを JVFiledelete に変更
2022/2/22	4.7.0.1	2	修正	・スタートキット(CD/DVD-ROM)の提供終了に伴い、関連する記述を修正
		3	修正	・【戻り値】の「-501」にスタートキット(CD/DVD-ROM)の提供終了を追記
2021/10/19	4.7.0	3	修正	・JVOpen/JVRTOpen の戻り値「-305」を追加
2021/5/13	4.6.0	2	修正	・JVOpen メソッドの fromtime パラメータ修正に伴い、パラメータ設定方法の説明を修正
		2	修正	・動画再生ブラウザの説明を修正
		3	追加	・JVOpen/JVRTOpen の戻り値「-113」を追加
		3	修正	・JVOpen/JVRTOpen の戻り値「-112」、「-114」の説明を修正
2017/1/24	4.5.1	2	修正	・JV-Link マイナーバージョンアップのため版番号改訂
2016/7/26	4.5.0	2	修正	・JVMVCheckWithType、JVMVPlayWithType の【パラメータ】を追加 movietype“02”(マルチカメラ動画)、“03”(パトロール動画)を追加
		2	修正	・JVSetUIProperties の【解説】のダイアログを変更
2016/4/19	4.4.2	—	修正	・開発環境更新に伴うJV-Link マイナーバージョンアップのため版番号改訂
2015/10/20	4.4.1	—	修正	・開発環境更新に伴うJV-Link マイナーバージョンアップのため版番号改訂
2014/6/10	4.4.0	2	修正	・JVMVPlay/JVMVPlayWithTypeのMP4形式動画の対応に伴い、動画種別に関する記述を追記 ・JVMVPlay/JVMVPlayWithType の【解説】の動画再生方式に IE を追加 ・JVMVPlay/JVMVPlayWithTypeの【補足】に映像形式変更に関する記述を追記
		2	修正	・JVSetUIProperties の【解説】のダイアログを変更
2013/9/25	4.3.2	2	修正	・JVMVPlay/JVMVPlayWithTYpe、JVMVOpen、JVMVRead の【補足】に開発を行う際の説明を追記
2013/4/9	4.3.1	—	修正	・JV-Link マイナーバージョンアップのため版番号改訂
2012/12/4	4.3.0	2	追加	・JVMVChcekWithType メソッドを追加
		2	修正	・JVMVPlayWithType メソッドの【パラメータ】を追加 movietype“01”(パッドック動画)を追加
		2	修正	・メソッド追加に伴い、関連する他のメソッドの記述を修正
2012/2/21	4.2.0	1	修正	・JVSetUIProperties の【解説】の拡張ダイアログ表示に関する説明を追記
		2	修正	・JVSetUIProperties の拡張ダイアログ表示フラグのプロパティ名の誤りを修正

## JV-Linkインターフェース仕様書

2011/9/28	4.1.0	1	修正	・JVSetUIProperties のダイアログを変更 ダイアログ変更に伴い、「JRA-VANからのお知らせ」「払戻ダイアログ」の表示に関する説明を修正
2011/6/21	4.0.0	1	追加	・JVOpen Option“2”の指定できないデータ種別 ID に MING を追加
2010/10/12	3.5.0	1	追加	・m_payflag プロパティを追加
		2	追加	・JVWatchEvent、JVWatchEventClose メソッドを追加
		3	追加	・JVWatchEvent、JVWatchEventClose の戻り値を追加
		2	修正	・JVRTOpen 第二引数の指定方法についてイベント発生時の引数の指定も可能とする記述を追加 ・JVSetUIProperties のダイアログを変更 ・JVSetUIProperties の説明に払戻ダイアログに関するを追加
2009/12/8	3.4.0	3	修正	・JVCourseFile の【戻り値】に「-202」を追記
		3	修正	・JVCourseFile2 の【戻り値】に「-202」を追記
2009/9/8	3.3.0	2	修正	・JVSetUIProperties のダイアログを変更
2009/6/23	3.2.1	2	修正	・JVCourseFile の【パラメータ】の説明を修正 コース説明の最大バイト数を修正
2009/3/17	3.2.0	2	修正	・JVOpen の Option パラメータと dataspec パラメータの指定不可に関する組合せ表に追記(今週データは COMM が指定不可)
		2	追加	・JVCourseFile2 メソッドを追加
		3	追加	・JVCourseFile2 の戻り値を追加
2008/11/4	3.1.0.1	2	修正	・JVCourseFile の【パラメータ】の説明に追記 最新のコース図を取得する場合のパラメータについての説明を追記 コース説明の最大バイト数を追記
2008/9/24	3.1.0	2	追加	・JVCourseFile メソッドを追加
		3	追加	・JVCourseFile の戻り値を追加
2008/6/17	3.0.0	—	修正	Windows Vista 対応によりバージョン更新
2007/12/25	2.4.0	2	追加	・JVMVOpen、JVMVRead、JVMVPlayWithType メソッドを追加
		3	追加	・JVMVOpen、JVMVRead の戻り値を追加
		3	修正	・メソッド追加に伴い、関連する他のメソッドの記述を修正
2007/5/9	2.3.1	2	修正	・JVFukuFile の【パラメータ】の説明を修正 誤:作成した勝負服のビットマップファイルを出力するフォルダを指定 正:勝負服ファイルの出力ファイル名をフルパスで指定 ・JVFukuFile、JVFuku の【解説】の説明を修正
2007/4/25	2.3.0	2	追加	・JVFukuFile、JVFuku メソッドを追加
		3	追加	・JVFukuFile、JVFuku の戻り値を追加
		2	修正	・JVOpen の lastfiletimestamp の説明の誤りを修正
2006/11/7	2.2.0	2	追加	・JVMVCheck、JVMVPlay メソッドを追加
		2	修正	・JVSetSaveFlag の【戻り値】の説明を修正 誤:利用キーが正しくセットされた場合... 正:保存フラグが正しくセットされた場合...
		2	修正	・JVSetSavePath の【戻り値】の説明を修正 誤:利用キーが正しくセットされた場合... 正:保存パスが正しくセットされた場合...
		2	修正	・JVGets の解説を修正 VisualBasic.NET におけるメモリ解放に関する説明を削除
		3	追加	・JVMVPlay の戻り値を追加
2004/11/2	2.1.1	2	修正	・JVRead の説明を修正 Buff パラメータの解放に関する説明を【補足】へ移動し仕様とする
		2	修正	・JVGets の解説を修正 VisualBasic.NETに関する補足を追加 コーディング例の VisualBasic のバージョンを明確化(VB6)
2004/10/7	2.1.0	2	追加	・JVGets メソッドを追加
		2	修正	・JVOpen/JVRead の構文を修正

2004/8/10	2.0.0	1	追加	・m_TotalReadFilesize の説明を修正 1024 で割った結果が 0KB となる場合は、1KB を返すように変更
		1	追加	・ParentHWnd プロパティを追加
		3	追加	・JVRead の戻り値"-503"の説明を追加
2004/6/9	1.1.7	2	追加	・JVInit の解説に呼出タイミングに関する説明を追加
2004/4/2	1.1.5.1	2	追加	・JVOpen の dataspec の説明に既知の障害に関する記述を追加 (複数指定時に処理時間が長くなる)
		2	追加	・JVOpen の lastfiletimestamp の説明に処理中断・再開に関する記述を追加
		2	追加	・JVRead の buff の説明に既知の障害に関する記述を追加 (JVRead 内におけるデータ格納バッファの開放と確保)
2004/3/23	1.1.5	2	追加	・JVInit の説明に使用可能文字に関する記述を追加
		2	追加	・JVOpen の option パラメータにコード=4 を追加
		3	追加	・JVOpen/JVRTOpen の戻り値"-504"を追加
		3	追加	・JVOpen/JVRTOpen の戻り値"-301"の説明を修正
		2	追加	・Option パラメータと dataspec パラメータの指定不可に関する組合せ表を修正(セットアップ時に TOKU の指定が可能) ・セットアップデータ取得時における dataspec=TOKU に関する補足を追加
		1	修正	・m_TotalReadFilesize の説明を修正 戻り値である総データサイズが long の最大値を超えたため、単位を KB に変更
2004/3/2	1.1.4	2	修正	・JVSetUIProperties について説明を追加
		3	修正	・JVOpen/JVRTOpen の戻り値"-1"を追加
		3	修正	・JVStatus の戻り値"-502"の説明を修正
		3	追加	・JVRead の戻り値"-402", "-403", "-502"を追加
		3	追加	・JVSetServiceKey/JVSetSaveFlag/JVSetSavePath の戻り値"-101"を追加
		2	修正	・JVOpen の fromtime の説明の誤りを修正 ・JVRead のパラメータの説明を補足
2003/9/3	1.1.2 β	3	修正	・JVOpen/JVRTOpen の戻り値"-112"の説明文修正 ・JVOpen/JVRTOpen の戻り値"-113"の削除
		2	追加	・JVSkip メソッドの追加 ・m_CurrentFileTimestamp プロパティの追加
		2	修正	・JVRTOpen の第一引数の型の誤りを修正 誤: uint 型 → 正: string 型 ・JVRTOpen の第二引数の指定方法についてレース単位の場合に"YYYYMMDDJJRR"の指定も可能とする記述を追加
2003/8/8	1.1.1 β	1	追加	・m_JVLinkVersion, m_TotalReadFilesize, m_CurrentReadFilesize の3つのプロパティを追加
2003/6/3	1.0.5 β	2	追加	・JVRead の戻り値"-1"について説明を追加
2003/5/16	1.0.4 β	3	追加	・コード表に各戻り値について対処方法の記述を追加
		1	修正	・m_savepath プロパティのデフォルト値の誤りを修正し説明を追加 誤: 「%InstallPath%¥data」 → 正: 「%InstallPath%」 ・説明の中のメソッド名の誤りを修正 誤: 「JVUISetProperties」 → 正: 「JVSetUIProperties」
2003/5/2	1.0.3 β	2	修正	・JVOpen の引数の名前を実装された名前に合わせて修正 誤: 「lasttime」 → 「lastfiletimestamp」
		2	追加	・JVSetUIProperties 実行時の保存パス指定の処理について説明を追加 ・JVSetSavePath 実行時の保存パス指定内容について説明を追加
		3	追加	・JVOpen/JVRTOpen の戻り値として -1 (該当データ無し) を追加
2003/4/28	1.0.2 β	2	修正	・JVSetUIProperties, JVSetServiceKey, JVSetSaveFlag, JVSetSavePath のエラー時の戻りコードを修正 ・JVOpen の fromtime パラメータの省略を不可に修正

2003/4/22	1.0.1 <i>β</i>	2	追加	<ul style="list-style-type: none"> <li>・JVSetUIPropertiesのダイアログイメージ図を追加</li> <li>・JVOpenの <i>optin=3</i> での呼び出し時に表示されるダイアログイメージ図とその説明を追加</li> <li>・JVOpenの <i>option</i>パラメータと <i>dataspec</i>パラメータの不正な組み合わせについて記述を追加</li> <li>・JVOpenの <i>lasttime</i>パラメータについての説明文を追加</li> </ul>
		3	修正	「3. コード表」の戻り値と意味を修正
		3	追加	<ul style="list-style-type: none"> <li>・JVOpen/JVRTOpenの戻り値として-303(利用キーが設定されていない)を追加</li> <li>・JVOpenの戻り値として-115(<i>option</i>パラメータが不正)を追加</li> <li>・JVOpenの戻り値として-116(<i>dataspec</i>と<i>option</i>の組み合わせが不正)を追加</li> </ul>
		—	新規	初版作成
2003/4/1	1.0 <i>β</i>			

## 目次



JRA-VAN Data Lab. SDK .....	1
1. プロパティ.....	7
2. メソッド .....	9
2.1 メソッド一覧 .....	9
2.2 メソッド詳細 .....	10
JVInit.....	10
JVSetUIProperties .....	11
JVSetServiceKey .....	14
JVSetSaveFlag.....	15
JVSetSavePath .....	16
JVOpen.....	17
JVRTOpen.....	22
JVStatus .....	24
JVRead.....	25
JVGets.....	27
JVSkip.....	29
JVCancel.....	31
JVClose .....	32
JVFiledelete .....	33
JVFukuFile.....	34
JVFuku .....	35
JVMVCheck.....	36
JVMVCheckWithType.....	36
JVMVPlay .....	38
JVMVPlayWithType .....	38
JVMVOpen.....	41
JVMVRead .....	43
JVCourseFile .....	45
JVCourseFile2.....	47
JVWatchEvent.....	49
JVWatchEventClose .....	52
3. コード表 .....	53

## JV-Linkインターフェース仕様書

### 1. プロパティ

プロパティ	説明
Integer m_saveflag	<p>サーバーからダウンロードしたファイルを m_savepath が示すパスに保存するかどうかのフラグを保持します。</p> <p>0: 保存しない 1: 保存する</p>
String m_savepath	<p>JV-Data を保存するディレクトリへのパスを保持します。JVInit 呼び出し時にレジストリから値がセットされます。JV-Link のインストール直後にはデフォルトで次の値がセットされています。</p> <p>%InstallPath%</p> <p>※%InstallPath%は JRA-VAN Data Lab. SDK がインストールされたパス</p> <p>JV-Data はこのパスの下に作成される”cache”と”data”フォルダに保存されます。 値を変更する場合は JVSetSavePath または JVSetUIProperties を使用します。</p>
String m_servicekey	<p>JRA-VAN Data Lab. サービスを利用する権利を確認するための利用キー(17桁)を保持します。JVInit 呼び出し時にレジストリから値がセットされます。デフォルトでは値が設定されていないので認証出来ません。値を変更する場合は JVSetServiceKey または JVSetUIProperties を使用します。</p>
String m_JVLinkVersion	<p>JV-Link のバージョンを4桁数字(例:0100)で保持します。値は変更できません。</p> <p>JV-Link を呼び出すソフトが JV-Link の特定のバージョン以降でないと動作しないことがわかっている場合に、これをチェックすることで誤動作を防ぐために利用できます。</p>
Long m_TotalReadFileSize	<p>JVOpen 呼び出しから戻った時にこれから読み込む JV-Data の総データサイズを1024で割った値がセットされています。(ただし、結果が0の場合は、1がセットされます。)値は変更できません。</p> <p>これは JVRead/JVGets の戻り値に”0”が返るまでに読み取るデータの合計サイズなので、JVRead/JVGets を使用した読み出し処理全体のプログレスバーの表示のために利用できますが、単位が KB なので注意が必要です。</p>
Long m_CurrentReadFileSize	<p>JVRead/JVGets で読み込んでいる現在のファイルのサイズがセットされています。値は変更できません。</p> <p>JVOpen 後の最初の JVRead/JVGets 呼び出しでセットされ、JVRead/JVGets の戻り値に”-1”が返されるまで同じ値を維持します。戻り値”-1”の次の JVRead/JVGets 呼び出しで次のファイルのサイズに変更されます。</p>
String m_CurrentFileTimestamp	<p>JVRead/JVGets で読み込んでいる現在のファイルのタイムスタンプがセットされます。</p> <p>JVOpen 後の最初の JVRead/JVGets 呼び出しでセットされ、JVRead/JVGets の戻り値に”-1”が返されるまで同じ値を維持します。戻り値”-1”の次の JVRead/JVGets 呼び出しで次のファイルのタイムスタンプに変更されます。</p>

プロパティ	説明
Long ParentHwnd	JV-Link が表示するメッセージダイアログのオーナーウィンドウを設定します。 JVOpen/JVRTOpen の呼出前に設定を行って下さい。  <b>※:JV-Link Ver2.0.0 以降である必要がありますので、バージョンチェック後に設定を行って下さい。</b>
Integer m_payflag	払戻ダイアログを表示するかどうかのフラグを保持します。  0:表示する 1:表示しない

## 2. メソッド

### 2.1 メソッド一覧

メソッド	処理内容
<b>JVInit</b>	JV-Link の初期化をします。
<b>JVSetUIProperties</b>	JV-Link の設定変更(ダイアログ版)をします。
<b>JVSetServiceKey</b>	JV-Link の設定変更(利用キー)をします。
<b>JVSetSaveFlag</b>	JV-Link の設定変更(保存フラグ)をします。
<b>JVSetSavePath</b>	JV-Link の設定変更(保存パス)をします。
<b>JVOpen</b>	蓄積系データの取得要求をします。
<b>JVRTOpen</b>	リアルタイム系データの取得要求をします。
<b>JVStatus</b>	ダウンロード進捗情報の取得をします。
<b>JVRead</b>	JV-Data の読み込みをします。
<b>JVGets</b>	JV-Data の読み込みをします。
<b>JVSkip</b>	JV-Data の読みとばしをします。
<b>JVCancel</b>	ダウンロードスレッドの停止をします。
<b>JVClose</b>	JV-Data 読み込み処理の終了をします。
<b>JVFiledelete</b>	ダウンロードしたファイルの削除をします。
<b>JVFukuFile</b>	勝負服画像情報要求をします。
<b>JVFuku</b>	勝負服画像情報要求(バイナリ)をします。
<b>JVMVCheck</b>	JRA レーシングビューアー レース映像公開チェック要求をします。
<b>JVMVCheckWithType</b>	JRA レーシングビューアー映像公開チェック要求をします。
<b>JVMVPlay</b>	JRA レーシングビューアー レース映像再生要求をします。
<b>JVMVPlayWithType</b>	JRA レーシングビューアー映像再生要求をします。
<b>JVMVOpen</b>	動画リストの取得要求をします。
<b>JVMVRead</b>	動画リストの読み込みをします。
<b>JVCourseFile</b>	コース図の取得要求をします。
<b>JVCourseFile2</b>	コース図の取得要求をします。
<b>JVWatchEvent</b>	確定・変更情報の発生を通知するイベント通知を開始します。
<b>JVWatchEventClose</b>	イベント通知を終了します。

## 2.2 メソッド詳細

**JVInit**

## JV-Linkの初期化

## 【構文】

**Long JVInit ( String 型 sid );**

## 【パラメータ】

*sid*

呼出元のソフトを識別するソフトウェアIDを最大64バイトの文字列で指定します。ここにセットされる値は JRA-VAN サーバーにアクセスする際に User-Agent HTTP ヘッダーとして使用されます。ソフトウェア ID は JRA-VAN に登録済みのものである必要があります。デフォルトで“UNKNOWN”が使用可能です。また、使用可能な文字は、半角のみとし、アルファベット、数字、スペース、アンダースコア( )、ピリオド(. )、スラッシュ(/)です。

## 【戻り値】

初期化処理が正しく終了した場合は 0、エラーが発生した場合はエラーの理由コードとして負の値が返ります。(「3. コード表」参照)

## 【解説】

**JVInit** は JV-Link の初期化のために必ず最初に呼び出さなければなりません(※)。**JVInit** は JV-Link が使用する変数などのリソースを初期化します。サーバーとの通信は行なわれません。**JVInit** は自身のプロパティをレジストリの内容で初期化しますが、**JVSetUIProperties** を呼び出した後に変更されたレジストリの内容を読み込ませるという理由で再度**JVInit** を呼び出す必要はありません。**JVOpen** あるいは **JVRTOpen** により変更されたレジストリ内容は再度読み込まれます。

※アプリケーションの初期化時に呼び出しを行ってください。**JVOpen** あるいは **JVRTOpen** の都度呼び出す必要はありません。

**JVSetUIPropertie  
s**

## JV-Linkの設定変更(ダイアログ版)

## 【構文】

**Long JVSetUIProperties();**

## 【パラメータ】

なし

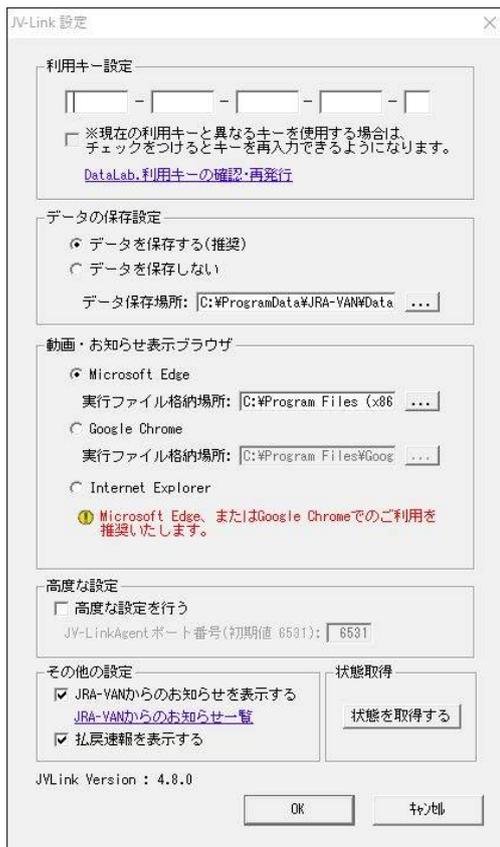
## 【戻り値】

プロパティが正しくセットされた場合あるいはキャンセルボタンが押された場合は0、エラー発生により終了した場合は-100が返り、値はレジストリにセットされません。

## 【解説】

JV-Link が持つプロパティのうちエンドユーザーによって変更可能な項目について値を変更するためのダイアログ表示と値のセットを行います。

表示されるダイアログは下図の通りです。



**JVSetUIProperties** で設定可能なプロパティの値は以下のとおりです。(利用キーが設定済の場合は、利用キーは変更不可となります。)

- m\_\_saveflag (保存フラグ)
- m\_\_savepath (保存パス)
- m\_\_servicekey (利用キー)
- m\_\_payflag (払戻ダイアログ表示フラグ)

**JVSetUIProperties** メソッドでプロパティを設定するとその値がレジストリに保存され、この値が JV-Link の動作に反映されるのは **JVInit**, **JVOpen**, **JVRTOpen** を呼び出したタイミングです。

JV-Link のインストール直後は利用キーに値が設定されていないので JRA-VAN サーバーの認証を受けられません。デフォルト値のまま **JVOpen** / **JVRTOpen** メソッドを呼び出すと認証エラーとなります。

m\_savepath(保存パス)が存在しないパスであった場合には新規に該当フォルダを作成するかどうかを確認するダイアログを表示し、「OK」ボタンを押すとフォルダを作成します。JVOpen / JVRTOpen 実行時に該当フォルダが存在しないと「-211」(レジストリ内容が不正)エラーとなります。

「JRA-VAN からのお知らせを表示する」を設定すると、JRA-VANからのお知らせが存在した場合、JVOpen / JVRTOpen 実行時にお知らせメッセージダイアログが表示されます。ダイアログ内にある「確認した」ボタンを押すと再度 JVOpen / JVRTOpen を実行した際には該当の お知らせが表示されなくなります。

「後で確認する」ボタンを押すと、JVOpen の場合は再度 JVOpen を実行した際にお知らせが表示されます。

JVRTOpen の場合は一度競馬ソフトを終了するまで、再度 JVRTOpen を実行してもお知らせは表示されません。

「払戻ダイアログを表示する」を設定すると、払戻ダイアログが表示されます。

<b>JVSetServiceKey</b>	JV-Linkの設定変更(利用キー)
------------------------	--------------------

## 【構文】

```
Long JVSetServiceKey( String 型 servicekey );
```

## 【パラメータ】

*servicekey*

JRA-VAN サーバーと通信する際に、認証に使用する利用キー(17桁の英数字)を文字列で指定します。

## 【戻り値】

利用キーが正しくセットされた場合は0、指定された値が不正である場合あるいはエラー発生により終了した場合は-100が返り、値はレジストリにセットされません。(利用キーが設定済の場合は、利用キーは変更不可となります。)

## 【解説】

JVSetServiceKey メソッドで利用キーを設定するとその値がレジストリに保存され、それ以降 JVInit, JVOpen, JVRTOpen を呼び出したタイミングでこのレジストリの値が使用されます。

**JVSetSaveFlag**

## JV-Linkの設定変更(保存フラグ)

## 【構文】

**Long JVSetSaveFlag( Long 型 saveflag );**

## 【パラメータ】

*saveflag*

ダウンロードしたファイルを保存パス(m\_savepath)に保存するかどうかを数値で指定します。

saveflag = 0	保存しない
saveflag = 1	保存する
上記以外	エラー

「保存する」に設定した場合、データはJRA-VANサーバーで該当データが提供されている間保存されます。

## 【戻り値】

保存フラグが正しくセットされた場合は0、指定された値が不正である場合あるいはエラー発生により終了した場合は-100が返り、値はレジストリにセットされません。

## 【解説】

JVSetSaveFlag メソッドで利用キーを設定するとその値がレジストリに保存され、それ以降JVInit,JVOpen,JVRTOpen を呼び出したタイミングでこのレジストリの値が使用されます。

**JVSetSavePath**

## JV-Linkの設定変更(保存パス)

## 【構文】

**Long JVSetSavePath( String 型 savepath );**

## 【パラメータ】

*savepath*

ダウンロードしたファイルを保存するパスを文字列で指定します。デフォルトでは JV-Link のインストールされたパスが設定されています。デフォルトの JV-Link インストールパスは、OSがインストールされたドライブの”Program Files¥JRA-VAN¥Data Lab”となります。引数には実際に存在するパスを指定する必要があります。

## 【戻り値】

保存パスが正しくセットされた場合は0、指定された値が不正である場合あるいはエラー発生により終了した場合は-100が返り、値はレジストリにセットされません。

## 【解説】

JVSetSavePath メソッドで保存パスを設定するとその値がレジストリに保存され、それ以降 JVInit,JVOpen,JVRTOpen を呼び出したタイミングでこのレジストリの値が使用されます。

実際に JV-Data が保存されるのはここで指定されたパスの下に自動的に作成される”cache”と”data”フォルダになります。”cache”と”data”フォルダが存在しない場合には JVOpen,JVRTOpen が自動的に作成しますが、保存パスそのものが存在しない場合には JVOpen,JVRTOpen は「-211」(レジストリ内容が不正)エラーとなります。したがってこの JVSetSavePath メソッドで設定する保存パスは存在するパスを指定する必要があります。

## JVOpen

## 蓄積系データの取得要求

## 【構文】

**Long JVOpen( String 型 dataspec , String 型 fromtime , Long 型 option , Long 型 readcount , Long 型 downloadcount , String 型 lastfiletimestamp );**

## 【パラメータ】

*dataspec*

読み出したいデータを識別するデータ種別IDを文字列として連結したものを指定します。1つのデータ種別IDは4桁固定ですので *dataspec* に指定する文字列は4の倍数桁でなければいけません。option パラメータとの組み合わせで指定できないデータ種別 ID があります。

**既知の障害について**

**・dataspec を複数個指定した場合、個別に指定した場合と比較すると「対象ファイル数が多い場合に JVRead の処理時間が遅くなる」という障害が報告されています。処理時間が遅い場合には、dataspec を個別に指定、または Fromtime に読み出し開始ポイント時刻・終了ポイント時刻を指定して回避してください。また、セットアップデータ取得時は option=4(ダイアログ無しセットアップ)を指定することで、セットアップ時のダイアログ表示を回避 可能です。**

指定可能なデータ種別IDについては「JV-Data仕様書」を参照して下さい。

*fromtime*

指定方法は以下の 2 通り。

- ・読み出し開始ポイント時刻のみを指定する場合

*dataspec* に指定したデータの読み出し開始ポイントを時刻(YYYYMMDDhhmmss の形式)で指定します。ここで指定した時刻より大きくかつ現在時刻までに提供されたデータが読み出しの対象となります。

例) *fromtime*

YYYYMMDDhhmmss

- ・読み出し開始ポイント時刻、および読み出し終了ポイント時刻を指定する場合  
dataspec に指定したデータの読み出し開始ポイント(YYYYMMDDhhmmss の形式)と読み出し終了ポイント時刻(YYYYMMDDhhmmss の形式)を「-(半角ハイフン)」で結合し指定します。ここで指定した開始時刻より大きくかつ指定した終了時刻までに提供されたデータが読み出しの対象となります。

例) fromtime

YYYYMMDDhhmmss-YYYYMMDDhhmmss

なお、dataspec にて以下のデータ種別 ID を指定する場合、全データを取得するため、読み出し終了ポイント時刻を指定することができません。指定した場合、「戻り値:-1(該当データなし)」が出力されます。

- ・TOKU(特別登録馬情報)
- ・DIFF、DIFN(蓄積系ソフト用 蓄積情報)
- ・HOSE、HOSN(競走馬市場取引価格情報)
- ・HOYU(馬名の意味由来情報)
- ・COMM(各種解説情報)

#### option

option = 1	通常データ
option = 2	今週データ
option = 3	セットアップデータ
option = 4	ダイアログ無しセットアップデータ (初回のみダイアログを表示します。)
上記以外	エラー

蓄積系ソフトがデータをメンテナンスする際の差分データの読み出しの場合は1を指定します。非蓄積系ソフトが今週ぶんのデータのみを読み出したい場合には2を指定します。2を指定すると直近の未来のレースに関するデータ(出走予定馬の過去走情報を含む)と直前のレースの成績関連のデータに該当するデータだけを読み出します。1を指定した場合は全てのデータの中から dataspec,fromtime に該当するデータが読み出されます。また、蓄積系ソフトがセットアップを行なう場合は3または4を指定します。3または4を指定した場合はセットアップ専用のデータを JRA-VAN サーバー から取得し読み出します。セットアップを分割して行いたい場合(データ種別毎にセットアップを行う場合等)には、4を指定します。4を指定した場合は、セットアップデータの取得元を指定するダイアログが初回のみ表示され2回目以降は初回に指定した取得元にてセットアップが行われます。

*readcount*

*dataspec,fromtime,option* で指定した条件に該当する全ファイル数が返されます。呼び出し時に値をセットする必要はありません。

*downloadcount*

*readcount* で返された数のうちダウンロードが必要なファイルの数が返されます。

*lastfiletimestamp*

*dataspec,fromtime,option* で指定した条件に該当する全ファイルのうち最も新しいファイルのタイムスタンプが YYYYMMDDhhmmss の形式で返されます。このタイムスタンプは次に *JVOpen* を呼び出す場合に *fromtime* として指定するために必要ですのでレジストリやファイルまたはデータベースなどに保存しておく必要があります。

***JVRead/JVGets の中断・再開*****・通常データの中断・再開**

最後に読んだファイルの *m\_CurrentFileTimestamp* を保持し、再開時の *JVOpen* の *fromtime* に、保持している *m\_CurrentFileTimestamp* を設定することで *JVRead/JVGets* を再開することができます。( *dataspec* を個別に指定した場合は *dataspec* 毎に保持を行う。)

**・セットアップデータの中断・再開**

最後に読んだファイル名を保持し、再開時には前回と同じパラメータにて *JVOpen* を行い、保持しているファイル名まで *JVSkip* を行う事で *JVRead/JVGets* を再開することができます。( *dataspec* を個別に指定した場合は *dataspec* 毎に保持を行う。)

## 【戻り値】

オープン処理が正しく終了した場合、0 が返されます。エラーが発生した場合にはエラーの理由コードとして負の数が返されます。(「3. コード表」参照)

## 【解説】

データ識別文字列の組み合わせと開始ポイント時刻(fromtime)で指定したデータ(ファイル群)を読み込むための準備をします。具体的には以下の処理を行います。

- ・dataspec,fromtime,option が正しいか検査を行います。
- ・dataspec,fromtime,option に該当するファイルのリストをサーバに問い合わせます。
- ・該当するファイルがローカルディスクに存在するかどうか検査します。
- ・ローカルディスクに無いデータをサーバからダウンロードするスレッドを開始します。
- ・ダウンロードスレッドが開始したら処理をアプリケーションに戻します。

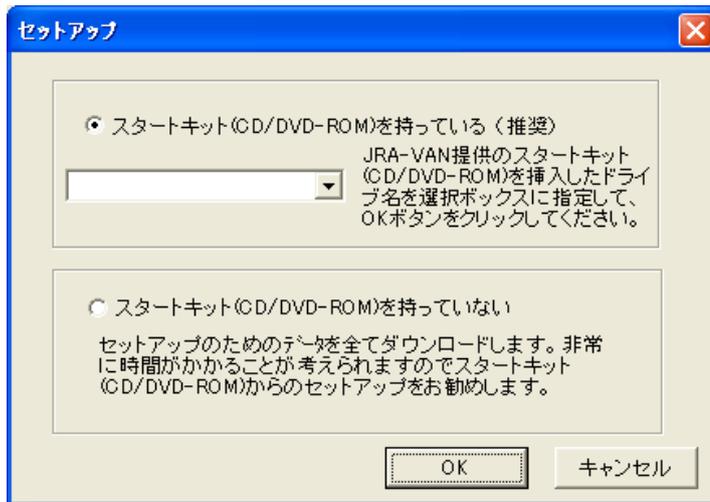
option パラメータの指定により JV-Link は以下のデータを返します。

Option	結果
1	サーバが提供している全てのデータの中から <i>dataspec</i> が合致し、 <i>fromtime</i> より大きくかつ現在時刻、または <i>fromtime</i> にて「-」(半角ハイフン)以降で指定した終了時刻までに該当するデータを取得します。
2	先週のレース結果情報と次週のレース関連情報を含んだ約1週間ぶんのデータの中から <i>dataspec</i> が合致し、 <i>fromtime</i> より大きい時刻に該当するデータ、または <i>fromtime</i> より大きくかつ <i>fromtime</i> にて「-」(半角ハイフン)以降で指定した終了時刻に該当するデータを取得します。
3, 4	セットアップ用データの中から <i>dataspec</i> が合致する <i>fromtime</i> より大きい時刻を持つ全てのデータ(前月までのデータ)と、今月の通常データの中で <i>dataspec</i> が合致する現在時刻、または <i>fromtime</i> にて「-」(半角ハイフン)以降で指定した終了時刻までに提供しているデータを取得します。(※1)

以下の option パラメータと *dataspec* の組み合わせにて指定が可能です。

Option	<i>dataspec</i> に指定可能なデータ種別ID
1	TOKU, RACE, DIFF, BLOD, SNAP, SLOP, WOOD, YSCH, HOSE, HOYU, DIFN, BLDN, SNPN, HOSN
2	TOKU, RACE, TCOV, RCOV, SNAP, TCVN, RCVN, SNPN
3, 4	TOKU, RACE, DIFF, BLOD, SNAP, SLOP, WOOD, YSCH, HOSE, HOYU, COMM, MING DIFN, BLDN, SNPN, HOSN

option に3または4を指定した場合、スタートキット(CD/DVD-ROM)からセットアップするか全てのデータをサーバからダウンロードするかを選択する次のようなダイアログが表示されます。(optionに4を指定した場合は、初回のみ表示されます。)



JRA-VAN 提供のスタートキット(CD/DVD-ROM)を持っている場合はこのダイアログで指定します。

このダイアログで「スタートキット(CD/DVD-ROM)を持っている」を選択すると指定されたドライブから必要なデータをローカルディスクにコピーし、足りないデータをサーバーからダウンロードします。古いスタートキット(CD/DVD-ROM)を使用した場合でも最大限スタートキット(CD/DVD-ROM)に収容されたデータを利用するように動作し、足りないぶんをサーバーからダウンロードします。

「スタートキット(CD/DVD-ROM)を持っていない」を選択するとセットアップのためのデータを全てサーバーからダウンロードします。スタートキット(CD/DVD-ROM)の提供は 2022 年3月をもちまして終了いたしましたので、こちらを選択してセットアップをお願いいたします。

※1: dataspec=TOKU(特別登録馬)に関する注意事項 セットアップデータ取得時には最新分(当週+翌週)の特別登録馬の取得が可能です。過去分の取得を行うには、通常データ取得にて fromtime を過去日付に設定し取得を行う必要があります。

また、新しいお知らせや新しいバージョンの JV-Link がリリースされていれば、告知ダイアログを表示します(新しいお知らせは、JRA-VAN からのお知らせを表示するになっている場合のみ表示されます)。

## JVRTOpen

## リアルタイム系データの取得要求

## 【構文】

**Long JVRTOpen( String 型 dataspec , String 型 key);**

## 【パラメータ】

*dataspec*

読み出したいデータを識別するデータ種別IDを文字列として指定します。1つのデータ種別IDしか指定できませんので4桁固定となります。

指定可能なデータ種別IDについては「JV-Data仕様書」を参照して下さい。

*key*

該当データを取得するための要求キーを指定します。要求するデータの提供単位に応じて以下のように指定します。

提供単位	指定するキー ( <i>key</i> )	説明
レース毎	“YYYYMMDDJJKKHRR” または “YYYYMMDDJJRR”	Y Y Y Y : 開催年 M M : 開催月 D D : 開催日 J J : 場コード K K : 回次 H H : 日次 R R : レース番号
開催日単位	“YYYYMMDD”	Y Y Y Y : 開催年 M M : 開催月 D D : 開催日
変更情報単位	各イベントから返されるパラメータ ※1	各イベントから返されるパラメータを指定 いただくことにより変更情報単位の取得が 可能※1

イベントに関しては「JVWatchEvent」を参照して下さい。各データの提供単位については「JV-Data仕様書」を参照して下さい。

## 【戻り値】

オープン処理が正しく終了した場合、0が返ります。エラーが発生した場合にはエラーの理由コードとして負の数が返されます。(「3. コード表」参照)

## 【解説】

データ識別文字列で指定したデータを読み込むための準備をします。具体的には以下の処理を行います。

- ・dataspec の検査を行います。
- ・dataspec および key に対応するデータをサーバにリクエストします。
- ・データの受信を完了した時点で処理をアプリケーションに戻します。

※1:各イベントから返されるパラメータを key に JVRTOpen を使用する場合 イベント通知を受信するメソッドから返されるパラメータを key として JVRTOpen を使用する場合は、Dataspec を以下のように指定してください。

種類	Dataspec
払戻確定	0B12
騎手変更	0B16
天候馬場状態変更	0B16
コース変更	0B16
出走取消・競走除外	0B16
発走時刻変更	0B16
馬体重発表	0B11

イベントから返されるパラメーターを key に JVRTOpen を使用した場合の提供単位は以下ようになります。

種類	提供単位
払戻確定	レース単位
騎手変更	変更情報発表単位(複数件の騎手変更情報を提供する場合もあります)
天候馬場状態変更	場単位
コース変更	レース単位
出走取消・競走除外	変更情報発表単位 (複数件の出走取消・競走除外情報を提供する場合もあります)
発走時刻変更	変更情報発表単位(複数件の発走時刻変更情報を提供する場合もあります)
馬体重発表	レース単位

**JVStatus****ダウンロード進捗情報の取得****【構文】****Long JVStatus();****【パラメータ】**

なし

**【戻り値】**

ダウンロード完了ファイル数が Long 型で返されます。ダウンロード処理にエラーが発生した場合は負のエラーコードが返されます。(「3. コード表」参照)

**【解説】**

JVOpen をコールした際に起動されたダウンロードスレッドがダウンロードを完了したファイル数を返します。ファイルのダウンロードが完了すると JVStatus の返す値は JVOpen をコールした際の downloadcount と一致します。この一致をもってダウンロードの終了を判断してください。以降 JVClose がコールされるまでの間、JVStatus はこの値を返します。ダウンロード処理の完了を待たず JVRead/JVGets を呼び出すと 予期しないエラーが発生する場合があります。

JVOpen が呼び出されていない、または dataspec に合致するデータが存在せずダウンロードが行われていない時に呼び出された場合や、エラーが発生した場合は負のエラーコードが返ります。

JVStatus の戻り値をプログレスバー表示に利用する場合は、ダウンロードすべきファイルが存在しないときに、0による除算が発生する場合がありますことに注意してください。

**JVRead****JV-Dataの読み込み****【構文】****Long JVRead( String 型 buff , Long 型 size , String 型 filename );****【パラメータ】***buff*

呼出側で用意したデータ格納バッファを指定します。1行単位で読み出しますので改行コード(0x0d,0x0a)を含めたレコードデータが収容可能なサイズ+1(ストリング終端文字 NULL)を用意します。

*size*

呼出側で用意したデータ格納バッファにコピーするデータの長さを指定します。この値がレコード長よりも小さい場合には残りのデータは切り捨てられ、データ格納バッファの最後の1バイトがストリング終端文字 NULLとなります。

*filename*

現在読み込み中のファイル名が JVReadから返されます。

**【戻り値】**

正常にレコードを読み込んだ場合はバッファにセットされたデータのバイト数が返ります。読み込んでいるファイルが次のファイルに切り替わる際には-1が返ります。全てのファイルを読み終わると0が返ります。エラーが発生した場合にはエラーの理由コードとして負の数が返されます。(「3. コード表」参照)

-1の戻り値は実際には物理ファイルの終わりに返されるので、全てのファイルの終わりには一度-1が返った後、次の呼び出しで0が返されます。

**【解説】**

JVOpen / JVROpen で準備した JV-Data を現在のファイルポインタから1行分読み出します。JVOpen / JVROpen を行わずに JVRead メソッドを呼び出すとエラーが返ります。

JVOpen ではデータ種別IDが複数指定できるため、JVRead メソッドは物理的には複数のファイルで

あっても1つのファイルであるかのように連続してデータを読み出します。ただし、ファイル間をまたぐごとに戻り値としてファイル切り替わり(-1)が返り、全てのファイルを読み終わった際に戻り値として EOF(0)が返ります。

例えば「2002年11月10日以降現在までの RACE データ」を指定して読み出した場合に、レース詳細データが36件存在したとすると1回目から36回目の呼出しまではレース詳細のレコードが1行ずつバッファにセットされ、37回目の呼び出しでファイル切り替わり(-1)が返されます。38回目以降の呼出しには馬毎レース情報がセットされます。全てのレコードが無くなった時点で EOF(0)が返されます。

**【補足】**

JVRead では、渡されたデータ格納バッファは JV-Link 内にて解放し、新たに確保したバッファを buff に割り当てられます。

## JVGets

## JV-Dataの読み込み

## 【構文】

**Long JVGets( Byte Array 型 buff , Long 型 size , String 型 filename );**

## 【パラメータ】

*buff*

データが格納されたBYTE型配列がセットされるポインタを指定します。

*size*

コピーするデータの長さを指定します。この値がレコード長よりも小さい場合には残りのデータは切り捨てられ、データ格納バッファの最後の1バイトがSTRING終端文字 NULLとなります。

*filename*

現在読み込み中のファイル名が JVGets から返されます。

## 【戻り値】

正常にレコードを読み込んだ場合はバッファにセットされたデータのバイト数が返ります。読み込んでいるファイルが次のファイルに切り替わる際には-1が返ります。全てのファイルを読み終わると0が返ります。エラーが発生した場合にはエラーの理由コードとして負の数が返されます。(「3. コード表」参照)

-1の戻り値は実際には物理ファイルの終わりに返されるので、全てのファイルの終わりには一度-1が返った後、次の呼び出しで0が返されます。

## 【解説】

JVOpen / JVROpen で準備した JV-Data を現在のファイルポインタから1行分読み出します。

JVOpen / JVROpen を行わずに JVGets メソッドを呼び出すとエラーが返ります。

JVOpen ではデータ種別IDが複数指定できるため、JVGets メソッドは物理的には複数のファイルであっても1つのファイルであるかのように連続してデータを読み出します。ただし、ファイル間をまたぐごとに戻り値としてファイル切り替わり(-1)が返り、全てのファイルを読み終わった際に戻り値として EOF(0)が返ります。

例えば「2002年11月10日以降現在までのRACEデータ」を指定して読み出した場合に、レース詳細データが36件存在したとすると1回目から36回目の呼び出しまではレース詳細のレコードが1行ずつバッファにセットされ、37回目の呼び出しでファイル切り替わり(-1)が返されます。38回目以降の呼び出しには馬毎レース情報がセットされます。全てのレコードが無くなった時点でEOF(0)が返されます。

#### JVGetsについて

JVGetsは従来のJVReadと互換性のあるメソッドとして、Ver2.1.0で新たに追加された公開メソッドです。

従来のJVReadは、内部で渡されたメモリを解放し、SJISで開いたファイルをUNICODE変換して新たに確保したメモリアreaに渡す処理をしていることから、パフォーマンス低下の原因となっていました。

JVGetsでは、メモリ受け渡しをバイト配列型のポインタで行い、そのポインタに対してメモリアreaを確保して渡す方法になります。その際、SJISはSJISのまま渡すことにより、JV-Link内部での変換およびアプリケーション側でのUNICODE→SJIS変換が不要になりコード変換におけるオーバーヘッドがなくなりました。

VB以外の言語では、VARIANT\*によってバイト型SafeArrayのポインタを受け取り、Returnする際にJV-Link内部で確保したByte SafeArrayをセットして返します。

途中の動作はすべてJVReadと共通しているため、JVGets → JVRead → JVGetsというように交互に呼ばれたとしても、矛盾なくレコードが取得できるように構成されています。

また、JVGetsではメモリの解放を行わないので、アプリケーション側で読み出しの度に解放する必要があります。

#### 【 VisualBasic6でのJVGets読み出し方法】

Dim BuffSize As Long	‘バッファサイズ
Dim BuffName As String	‘バッファ名
Dim ReturnCode As Long	‘JV-Link 返値
Dim bytData() As Byte	‘JVGets用Byte型配列ポインタ

ReturnCode = frmMain.JVLink1.JVGets(bytData, BuffSize, BuffName)

Debug.Print bytData                    ‘データを表示します。

Erase bytData                            ‘読み込みで確保されたメモリを明示的に削除します

**JVSkip****JV-Dataの読みとばし****【構文】**

```
void JVSkip( );
```

**【パラメータ】**

なし

**【戻り値】**

なし

**【解説】**

JVOpen で準備した JV-Data を読み込み中に不要なレコード種別を読み飛ばすために使用します。JVSkip メソッドを呼び出すと現在読み込み中のファイルにレコードが残っていても次のファイルの先頭までファイルポインタを進めます。

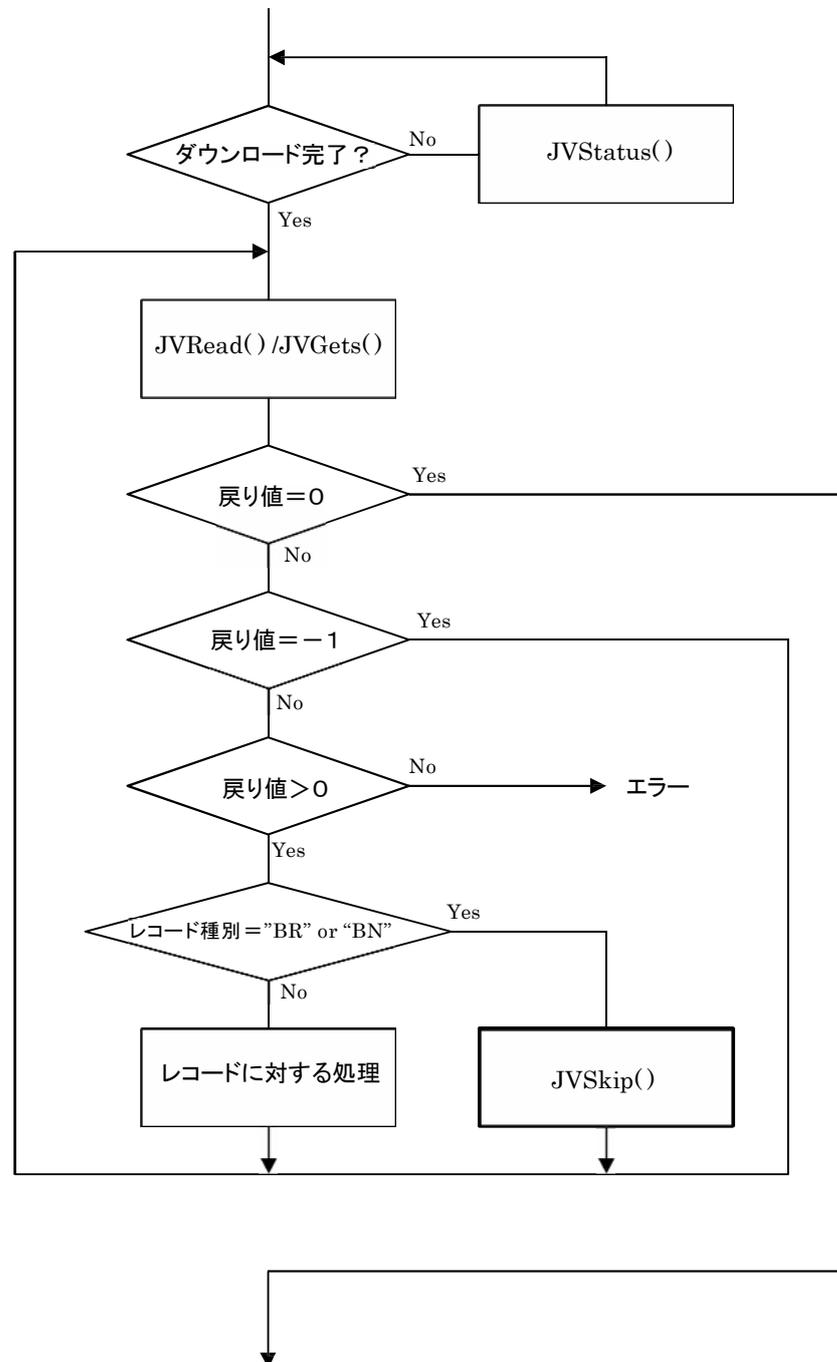
JVRead/JVGets メソッドは物理的には複数のファイルであっても1つのファイルであるかのように連続してデータを読み出します。蓄積系データは1つのファイルにレコード種別は1種類しか收容されていません。したがって JVRead/JVGets メソッドで読み出されたレコードが処理不要のレコード種別であった場合に JVSkip メソッドを呼び出し、そのファイルに收容されている残りの処理不要レコードを読み飛ばし処理時間を短縮することができます。速報系データは1回の JVRTOpen に対して1ファイルしか返されないため JVSkip には意味がありません。

例えばデータ種別 **"DIFN"** を dataspec に指定して JVRead/JVGets を行なうと

- |              |                |
|--------------|----------------|
| ・レース詳細("RA") | ・馬毎レース情報("SE") |
| ・競走馬("UM")   | ・騎手("KS")      |
| ・調教師("CH")   | ・生産者("BR")     |
| ・馬主("BN")    | ・レコード("RC")    |

の8種類のデータを読み出す可能性があります、このうち "BR" と "BN" が不要なデータである場合には次のロジックにより処理時間が短縮されます。

**【不要データ読み飛ばし】**



## 【注意点】

- ・JVSkip を複数回連続して呼び出しても1回呼び出した場合と同じ動作をします。
- ・JVOpen 直後に JVSkip を呼び出すと次の JVRead/JVGets では2ファイル目の先頭レコードを読みます。
- ・JVRead/JVGets で-1 が返った直後に JVSkip を呼び出すと次のファイルを読み飛ばします。
- ・JVSkip で読み飛ばしたファイルが最後のファイルである場合は次の JVRead/JVGets で0が返ります。

<b>JVCancel</b>	<b>ダウンロードスレッドの停止</b>
-----------------	----------------------

## 【構文】

```
void JVCancel( );
```

## 【パラメータ】

なし

## 【戻り値】

なし

## 【解説】

JVOpen により起動されたファイル準備処理(ダウンロード)を中止します。

JVOpen を呼び出すとローカルディスクに無いデータは JRA-VAN サーバーからダウンロードを開始します。ダウンロード中はその進捗状況を JVStatus で取得することが可能ですが、このときアプリケーション側から JVCancel を呼び出すことで、これらのファイル準備処理を中断することができます。

JVCancel によって中断した状態で JVRead/JVGets を呼び出すとエラーとなります。

JVCancel の代わりに JVClose を呼び出した場合もファイル準備処理を中断します。

**JVClose**

JV-Data読み込み処理の終了

## 【構文】

**Long JVClose( );**

## 【パラメータ】

なし

## 【戻り値】

サービス終了処理が正しく終了した場合は、0が返ります。

## 【解説】

開いているファイルを全てクローズし、実行中のダウンロードスレッドがあれば中止します。保存パスが示すフォルダから不要なファイルを削除します。

**JVFiledelete****ダウンロードしたファイルの削除****【構文】****Long JVFiledelete( String 型 filename);****【パラメータ】***filename*

削除対象のファイル名を指定します。

**【戻り値】**

処理が正しく終了した場合は0、エラーが発生した場合は-1が返ります。

**【解説】**

保存パスが示すフォルダから指定されたファイルを削除します。

保存パスに保存されているファイルの問題により JVRead/JVGets 中にエラーが発生した場合、JVFiledelete メソッドでファイルを削除してください。削除が成功した後、直前に行なった JVOpen からの処理をやり直してください。

## JVFile

## 勝負服画像情報要求

## 【構文】

**Long JVFile(String 型 pattern, String 型 filepath);**

## 【パラメータ】

*pattern*

勝負服の色・模様を示す服色標示を最大全角 30 文字で指定します。服色標示の文字列は、レーシングプログラムに記載されているものとなります。

(パラメータ例) 「水色, 赤山形一本輪, 水色袖」

## 注意事項:

・勝負服画像が作成できない場合は、「No Image」画像が出力されます。

*filepath*

勝負服ファイルの出力ファイル名をフルパスで指定します。実際に存在するフォルダを指定する必要があります。

## 【戻り値】

処理が正しく終了した場合はコード 0 を返します。

エラーが発生した場合にはエラーの理由コードとしての負の数が返されます。(「3. コード表」参照)

## 【解説】

服色標示パラメータより、勝負服画像のファイルを作成します。

作成される画像データは、サイズ 50pix × 50pix のビットマップ形式(24ビット)となります。形式・サイズについては、必要に応じて競馬ソフト側で変換してご使用ください。

## JVFuku

## 勝負服画像情報要求(バイナリ)

## 【構文】

**Long JVFuku ( String 型 pattern, Byte Array 型 buff );**

## 【パラメータ】

*pattern*

勝負服の色・模様を示す服色標示を最大全角 30 文字で指定します。服色標示の文字列は、レーシングプログラムに記載されているものとなります。

(パラメータ例) 「水色, 赤山形一本輪, 水色袖」

## 注意事項:

・勝負服画像が作成できない場合は、「No Image」画像が出力されます。

*buff*

画像データが格納されたバイト配列がセットされるポインタを設定します。

## 【戻り値】

処理が正しく終了した場合はコード 0 を返します。

エラーが発生した場合にはエラーの理由コードとしての負の数が返されます。

(「3. コード表」参照)

## 【解説】

服色標示パラメータより、勝負服画像の画像データを返します。

作成される画像データは、サイズ 50pix × 50pix のビットマップ形式(24ビット)となります。形式・サイズについては、必要に応じて競馬ソフト側で変換してご使用ください。

**JVMVCheck**  
**JVMVCheckWithType**
**JRA レーシングビューアー 映像公開チェック要求**

## 【構文】

**Long JVMVCheck( String 型 key );**
**Long JVMVCheckWithType( String 型 movietype , String 型 key );**

## 【パラメータ】

*movietype*

再生を行う映像の種類を指定します。

JVMVCheck は JVMVCheckWithType の *movietype* に“00”を指定したものと同等です。

*key*

レース映像の公開状況をチェックするレースを指定します。

パラメータは、以下のように指定します。

種類	<i>movietype</i>	指定するキー ( <i>searchkey</i> )	説明
レース映像	“00”	“YYYYMMDDJJRR” または “YYYYMMDDJJKKHRR”	Y Y Y Y : 開催年 M M : 開催月 D D : 開催日 J J : 場コード K K : 回次 H H : 日次 R R : レース番号
パドック映像	“01”	“YYYYMMDDJJRR” または “YYYYMMDDJJKKHRR”	Y Y Y Y : 開催年 M M : 開催月 D D : 開催日 J J : 場コード K K : 回次 H H : 日次 R R : レース番号
マルチカメラ映像	“02”	“YYYYMMDDJJRR” または “YYYYMMDDJJKKHRR”	Y Y Y Y : 開催年 M M : 開催月 D D : 開催日 J J : 場コード K K : 回次 H H : 日次 R R : レース番号
パトロール映像	“03”	“YYYYMMDDJJRR” または “YYYYMMDDJJKKHRR”	Y Y Y Y : 開催年 M M : 開催月 D D : 開催日 J J : 場コード K K : 回次 H H : 日次 R R : レース番号

【戻り値】

動画公開チェック要求が正しく終了した場合、公開状況を0または1で返ります。公開あ

り:1

公開なし:0

※:MovieType、:Key で指定したレースが存在しない場合は-1が返ります。

エラーが発生した場合にはエラーの理由コードとしての負の数が返されます。(「3. コード表」参照)

※JVOpen/JVRTOpen/JVMVOpen 中は本メソッドを使用できません。オープン中の場合、先にJVClose を呼び出してから使用してください。

※1度公開されたパドック動画のレースが中止になった場合、該当レースについて本メソッドを使用すると公開なし:0が返ります。

JVMVPlay

JVMVPlayWithType

JRAレーシングビューアー映像再生要求

## 【構文】

**Long JVMVPlay( String 型 key);****Long JVMVPlayWithType( String 型 movietype , String 型 key );**

## 【パラメータ】

*movietype*

再生を行う映像の種類を指定します。

JVMVPlay は JVMVPlayWithType の *movietype* に“00”を指定したものと同等です。*key*

再生するレース映像を指定します。

JVMVPlayWithType では、*movietype* の指定によりパラメータの設定内容が異なります。

種類	<i>movietype</i>	指定するキー ( <i>key</i> )	説明
レース映像	“00”	“YYYYMMDDJJKKHRRRTT” または “YYYYMMDDJJRRRTT” または “YYYYMMDDJJKKHRRR” または “YYYYMMDDJJRRR”	Y Y Y Y : 開催年 M M : 開催月 D D : 開催日 J J : 場コード K K : 回次 H H : 日次 R R : レース番号 T T : 動画種別
パドック映像	“01”	“YYYYMMDDJJKKHRRR” または “YYYYMMDDJJRRR”	Y Y Y Y : 開催年 M M : 開催月 D D : 開催日 J J : 場コード K K : 回次 H H : 日次 R R : レース番号
マルチカメラ映像	“02”	“YYYYMMDDJJKKHRRR” または “YYYYMMDDJJRRR”	Y Y Y Y : 開催年 M M : 開催月 D D : 開催日 J J : 場コード K K : 回次 H H : 日次 R R : レース番号
パトロール映像	“03”	“YYYYMMDDJJKKHRRR” または “YYYYMMDDJJRRR”	Y Y Y Y : 開催年 M M : 開催月 D D : 開催日 J J : 場コード K K : 回次 H H : 日次 R R : レース番号

調教映像	“11” または “12” または “13”	“YYYYMMDDNNNNNNNNNN”	YYYY : 調教実施年 MM : 調教実施月 DD : 調教実施日 NNNNNNNNNN : 血統登録番号
------	------------------------------------	----------------------	---

WMV 形式の動画の場合のみ、TT には以下のオプションが指定可能です。  
MP4 形式の動画の場合に TT を指定しても再生は可能ですが、画質は常に一定となります。  
(MP4 形式の場合、TT 以降は JV-Link では認識されません)

TT=01	高解像度版を優先して再生
TT=02	通常版を優先して再生
指定なし	高解像度版を優先して再生
上記以外	エラー

#### 【戻り値】

映像再生要求が正しく終了した場合、0が返ります。エラーが発生した場合にはエラーの理由コードとしての負の数が返されます。(「3. コード表」参照)

#### 【解説】

key で指定したレース映像の再生を行います。具体的には以下の処理を行います。

- ・JRA レーシングビューアー連携機能が利用可能なソフトウェアであるかの認証を行います。
- ・指定した映像が公開されている場合、映像の再生を行います。

映像の再生は HTML5 Player (MP4 形式の場合)を利用して行います。

再生するブラウザは JV-Link 設定画面の動画再生ブラウザにて設定した「Microsoft Edge」「Google Chrome」「IE」のいずれかになります。

レース映像については、JVMVCheck メソッドもしくは JVMVCheckWithType メソッドを利用する事で、レース映像の公開状況を判別する事が可能となります。JVMVCheck もしくは JVMVCheckWithType と組み合わせてご利用下さい。

パドック映像、マルチカメラ映像、パトロール映像については、JVMVCheckWithType メソッドを利用する事で、パドック動画、マルチカメラ動画、パトロール動画の公開状況を判別する事が可能となります。JVMVCheckWithType と組み合わせてご利用下さい。

※1度公開されたパドック動画のレースが中止になった場合、該当のレースについて本メソッドを使用すると該当データ無し: -1が返ります。

調教映像については、JVMVOpen で対象映像を指定し、JVMVRead で公開中の調教映像の key が取得可能です。取得した key を JVMVPlayWithType に設定して下さい。また、movietype については”11”、”12”、”13”どれを指定しても再生される映像は同じです。

※JVOpen/JVRTOpen/JVMVOpen 中は本メソッドを使用できません。オープン中の場合、先に JVClose を呼び出してから使用してください。

**【補足】**

当メソッドを利用するためには、JRAレーシングビューア連携機能利用申請が必要になります。  
(未申請の場合、戻り値に-304 が返されます。)

詳細については JRA-VANホームページのソフト作者サポートページを参照ください。

※当メソッドを使用した開発を行う際には、JVInit メソッドにてソフトウェア ID に  
”SA000000/SD000004”をセットしていただくことで当メソッドを利用可能となります。

パトロール映像については、通常、各レース終了からおおよそ 40 分後に更新されます。

マルチカメラ映像については、通常、レース終了後の翌月曜日の午後に更新されます。

## JVMVOpen

## 動画リストの取得要求

## 【構文】

**Long JVMVOpen(String 型 movietype, String 型 searchkey);**

## 【パラメータ】

*movietype*

取得する動画リストの種類を指定します。種類によって、searchkey の内容が異なります。

*searchkey*

該当データを取得するための要求キーを指定します。

種類	<i>movietype</i>	指定するキー ( <i>searchkey</i> )	説明
調教映像 指定週全馬	"11"	"YYYYMMDD"	YYYY : 開催年 MM : 開催月 DD : 開催日 指定開催年月日の週に行われた調教の映像が対象になります。指定の開催年月日に出走する競走馬の調教映像ではなく、すべての競走馬が対象になります。データの順序は保障されません。
調教映像 指定週指定馬	"12"	"YYYYMMDDNNNNNNNNNN"	YYYY : 開催年 MM : 開催月 DD : 開催日 NNNNNNNNNN N : 血統登録番号 血統登録番号で指定された競走馬の、指定開催年月日の週の調教映像が対象になります。
調教映像 指定馬全調教	"13"	"NNNNNNNNNN"	NNNNNNNNNN : 血統登録番号 血統登録番号で指定された競走馬の、公開中の調教映像が対象になります。データの順序は調教日の降順です。

## 【戻り値】

処理が正しく終了した場合はコード 0 を返します。

エラーが発生した場合にはエラーの理由コードとしての負の数が返されます。

(「3. コード表」参照)

**【解説】**

データ識別文字列で指定したデータを読み込むための準備をします。具体的には以下の処理を行います。

- movietype、searchkey の検査を行います。
- movietype および searchkey に対応するデータをサーバにリクエストします。
- データの受信を完了した時点で処理をアプリケーションに戻します。

**【補足】**

当メソッドを利用するためには、JRAレーシングビューア連携機能利用申請が必要になります。  
(未申請の場合、戻り値に-304 が返されます。)

詳細については JRA-VAN ホームページのソフト作者サポートページを参照ください。

※当メソッドを使用した開発を行う際には、JVInit メソッドにてソフトウェア ID に  
"SA000000/SD000004" をセットしていただくことで当メソッドを利用可能となります。

## JVMVRead

## 動画リストの読み込み

## 【構文】

**Long JVMVRead( String 型 buff , Long 型 size);**

## 【パラメータ】

*buff*

呼出側で用意したデータ格納バッファを指定します。1行単位で読み出しますので改行コード(0x0d,0x0a)を含めたレコードデータが収容可能なサイズ+1(STRING終端文字 NULL)を用意します。

*size*

呼出側で用意したデータ格納バッファにコピーするデータの長さを指定します。この値がレコード長よりも小さい場合には残りのデータは切り捨てられ、データ格納バッファの最後の1バイトがSTRING終端文字 NULLとなります。

読み出されるレコードデータは以下のようになります。

種類	レコードデータ内容	最大長	説明
調教映像	“YYYYMMDDNNNNNNNNNN<改行>”	21	Y Y Y Y : 調教実施年 M M : 調教実施月 D D : 調教実施日 N N N N N N N N N N : 血統登録番号

※改行コードが不要な場合、*size*に最大長よりも2少ない値を設定してください。

## 【戻り値】

正常にレコードを読み込んだ場合はバッファにセットされたデータのバイト数が返ります。全てのデータを読み終わると0が返ります。エラーが発生した場合にはエラーの理由コードとして負の数が返されます。(「3. コード表」参照)

JVReadと異なり、データの読み終わりの際に-1を返さないことに注意してください。

## 【解説】

JVMVOpen で準備した動画リストをカレント行から1行分読み出します。

JVMVOpen を行わずに JVMVRead メソッドを呼び出すとエラーが返ります。

JVOpen / JVROpen 中は使用できません。(エラーが返ります。)

【補足】

JVMVRead では、渡されたデータ格納バッファは JV-Link 内にて解放し、新たに確保したバッファを buff に割り当てられます。

処理後は、JVClose を呼ぶ必要があります。

当メソッドを利用するためには、JRAレーシングビューア連携機能利用申請が必要になります。詳細については JRA-VAN ホームページのソフト作者サポートページを参照ください。

※当メソッドを使用した開発を行う際には、JVInit メソッドにてソフトウェア ID に "SA000000/SD000004" をセットしていただくことで当メソッドを利用可能となります。

## JVCourseFile

## コース図要求

この機能は都度コース図要求をすることで、最新のコース図画像ファイル、コース説明を表示できるように作成されたものです。そのためコース図画像ファイル、コース説明をローカルに蓄積するには適していません。コース情報をローカルに蓄積する場合は、コース説明を JVOpen で蓄積系データとして取得していただき、コース図画像ファイルを JVCourseFile2 で取得していただくようお願いいたします。

## 【構文】

**Long JVCourseFile(String 型 key, String 型 filepath, String 型 explanation);**

## 【パラメータ】

*key*

コース図を取得するための要求キーを指定します。

指定するキー ( <i>key</i> )	説明
“YYYYMMDDJJKKKT”	Y Y Y Y : 開催年 M M : 開催月 D D : 開催日 J J : 場コード K K K K : 距離 T T : トラックコード

※最新のコース図を取得したい場合は、開催年月日に「99999999」を指定してください。東京芝 2400m の最新のコース図を取得したい場合に指定する要求キーは、「9999999905240011」となります。

*filepath*

画像のファイルパス(ドライブ名を含むフルパス)が返されます。対象コースがない場合は「No Image」画像のファイルパスが返されます。

*explanation*

コースの説明が返されます。コースの説明は最大で 6800 バイトとなります。対象コースがない場合は空文字列が返されます。

## 【戻り値】

処理が正しく終了した場合はコード 0 を返します。

エラーが発生した場合にはエラーの理由コードとしての負の数が返されます。(「3. コード表」参照)

**【解説】**

key で指定した該当レースのコース図が取得できます。コンテンツサーバから取得したコース図は m\_savepath 以下の pictures フォルダに一時的に保存 されます。

取得したコース図は、サイズ 256pix × 200pix の GIF 形式となります。形式・サイズについては、必要に応じて競馬ソフト側で変換してご使用ください。

## JVCourseFile2

## コース図要求

## 【構文】

**Long JVCourseFile2(String 型 key, String 型 filepath);**

## 【パラメータ】

*key*

コース図を取得するための要求キーを指定します。

指定するキー ( <i>key</i> )	説明
“YYYYMMDDJJKKKKTT”	Y Y Y Y : 開催年 M M : 開催月 D D : 開催日 J J : 場コード K K K K : 距離 T T : トラックコード

※最新のコース図を取得したい場合は、開催年月日に「99999999」を指定してください。東京芝 2400m の最新のコース図を取得したい場合に指定する要求キーは、「9999999905240011」となります。注意事項:該当するコース図が存在しない場合は、

「No Image」画像が出力されます。

*filepath*

コース図ファイルの出力ファイル名をフルパスで指定します。実際に存在するフォルダを指定する必要があります。

## 【戻り値】

処理が正しく終了した場合はコード 0 を返します。エラーが発生した場合にはエラーの理由コー

ドとしての負の数が返されます。(「3. コード表」参照)

## 【解説】

*key* で指定した該当レースのコース図を取得します。取得したファイルは *filepath* で指定したパスに保存されます。

取得したコース図は、サイズ 256pix × 200pix の GIF 形式となります。形式・サイズについては、必要に応じて競馬ソフト側で変換してご使用ください。蓄積系データとして取得したコース情報からコース図を取得する際は、コース改修年月日を開催年、開催月、開催日として要求キーに指定してください。データ区分「2:更新」のデータは、コース解説やコース図画像ファイルが変更された際に提供するデ

ータです。従いまして、データ区分が「1:新規登録」、または「2:更新」のレコードが存在する場合は、このメソッドを使用してコース図を取得してください。

データ仕様の詳細につきましては、JV-Data 仕様書をご参照ください。

## JVWatchEvent

## イベント通知開始

イベント通知開始をおこなうことで、払戻確定、騎手変更、天候馬場状態変更、コース変更、出走取消・競走除外、発走時刻変更、馬体重発表が発表された際、イベントを受理することが可能になります。イベント通知開始およびイベントを受理するには特殊な宣言およびコード記述が必要となります。下記【イベント使用方法】に使用例を記述していますので参考にして下さい。

## 【構文】

```
Long JVWatchEvent();
```

## 【パラメータ】

なし

## 【戻り値】

処理が正しく終了した場合はコード 0 を返します。エラーが発生した場合にはエラーの理由コードとし

ての負の数が返されます。(「3. コード表」参照)

## 【解説】

確定・変更情報が発生した際、イベントを通知するスレッドを開始します。

JVInit を行わずに JVWatchEventメソッドを呼び出すとエラーが返ります。

## 【イベント】

イベントを受理するためのメソッドは下記の通りになります。

## 【イベント構文】

**Void** 各イベントメソッド名(**String** 型 **bstr**);

受信可能な確定・変更イベントの種類は以下のようになります。

種類	イベントメソッド名	説明
払戻確定	JVEvtPay	払戻確定が発表された際イベントを受理します。
騎手変更	JVEvtJockeyChange	騎手変更が発表された際イベントを受理します。
天候馬場状態変更	JVEvtWeather	天候馬場状態変更が発表された際イベントを受理します。
コース変更	JVEvtCourseChange	コース変更が発表された際イベントを受理します。
出走取消・競走除外	JVEvtAvoid	出走取消・競走除外が発表された際イベントを受理します。
発走時刻変更	JVEvtTimeChange	発走時刻変更が発表された際イベントを受理します。
馬体重発表	JVEvtWeight	馬体重が発表された際イベントを受理します。

## 【パラメータ】

*bstr*

JVRTOpen に渡すためのパラメータが返されます。確定・変更イベントから返されるパラメータは以下のようになります。

イベントメソッド名	パラメータ	説明
JVEvtPay	"YYYYMMDDJJRR"	Y Y Y Y : 開催年
JVEvtWeight		MM : 開催月 D D : 開催日 J J : 場コード R R : レース番号
JVEvtJockeyChange	"TTYYYMMDDJJRRNNNNNNNNNNNNNNNN"	T T : レコード種別 ID
JVEvtWeather		Y Y Y Y : 開催年
JVEvtCourseChange		MM : 開催月
JVEvtAvoid		D D : 開催日
JVEvtTimeChange		J J : 場コード
		R R : レース番号 NNNNNNNNNNNNNNNN : 送信年月日

## 【イベント使用方法】

※以下のコードはVisualBasic6 での使用例となります。

**‘WithEvents 付きでインターフェイスを宣言**

```
Friend WithEvents InterfaceJVLink As JVDTLabLib.JVLink
Dim ReturnCode As Long                                ‘JV-Link 返値
InterfaceJVLink = New JVDTLabLib.JVLink              ‘オブジェクトインスタンスを作成
ReturnCode = InterfaceJVLink.JVWatchEvent()          ‘イベント通知スレッド開始
```

‘上記の手順を踏み JVWatchEvent メソッドを行うことにより、  
‘下記のようなイベント通知を受信するメソッドを使用することが出来るようになります。

‘払戻確定イベントが発生した際の例を下記に記述します。

```
Private Sub InterfaceJVLink _JVEvtPay(ByVal bstr As String)
                                Handles InterfaceJVLink JVEvtPay

‘払戻確定イベントが発生した際に行いたい処理をここに記述
ReturnCode = frmMain.JVLink1.JVRTOpen("0B12", bstr)
End Sub
```

‘イベントが発生した際の処理を上記メソッドの中に記述してください。  
‘第一引数の bstr を key として JVRTOpen※1 に渡すことで対象リアルタイム系データを  
‘取得することが出来ます。

※1: 各イベントから返されるパラメータを key に JVRTOpen を使用する場合 イベント通知を受信するメソッドから返されるパラメータを key として JVRTOpen を使用する場合は、Dataspecを以下のように指定してください。

種類	Dataspec
払戻確定	0B12
騎手変更	0B16
天候馬場状態変更	0B16
コース変更	0B16
出走取消・競走除外	0B16
発走時刻変更	0B16
馬体重発表	0B11

JVWatchEventClose	イベント通知終了
-------------------	----------

## 【構文】

```
Long JVWatchEventClose();
```

## 【パラメータ】

なし

## 【戻り値】

サービス終了処理が正しく終了した場合は0、エラーが発生した場合は-1が返ります。

## 【解説】

イベント通知スレッドを終了します。イベント受信処理を終了したい場合ご使用ください。

### 3. コード表

#### JVInit

戻り値	意味	原因と対処
0	正常	
-101	sid が設定されていない	sid パラメータの渡し方に問題があると思われます。サンプルプログラム等を参照し、正しくパラメータが JV-Link に渡っているか確認してください。
-102	sid が 64byte を超えている	sid パラメータの渡し方に問題があるか、渡した内容に問題があると思われます。64byte 以内の正しい sid を設定してください。
-103	sid が不正 (sid の1桁目がスペース)	sid パラメータの内容に問題があると思われます。sid の1桁目は必ずスペース以外である必要があります。

#### JVSetUIProperties

戻り値	意味	原因と対処
0	正常(キャンセルボタンが押された場合を含む)	
-100	パラメータが不正あるいはレジストリへの保存に失敗	JVSetUIProperties の内部的なエラーが発生したと思われます。ユーザーがダイアログ内で指定した内容に問題がある場合はダイアログは閉じません。正しく動作した場合かキャンセルボタンをおされた場合だけダイアログが終了します。また、レジストリへのアクセス権限の問題で設定内容のレジストリへの反映に失敗した場合もこのエラーとなります。レジストリのアクセス権限に問題が無い場合は JRA-VAN へご連絡ください。

#### JVSetServiceKey/JVSetSaveFlag/JVSetSavePath

戻り値	意味	原因と対処
0	正常	
-100	パラメータが不正あるいはレジストリへの保存に失敗	内部的なエラーが発生したと思われます。また、レジストリへのアクセス権限の問題で設定内容のレジストリへの反映に失敗した場合もこのエラーとなります。レジストリのアクセス権限に問題が無い場合は JRA-VAN へご連絡ください。
-101	既に利用キーが登録されている	既に利用キーが登録されています。利用キーの変更を行うには、JV-Link のアンインストールを行い、再度インストールを行う必要があります。

## JVOpen/JVRTOpen

戻り値	意味	原因と対処
0	正常	
-1	該当データ無し	指定されたパラメータに合致する新しいデータがサーバーに存在しない。又は、最新バージョンが公開され、ユーザーが最新バージョンのダウンロードを選択しました。JVClose を呼び出して取り込み処理を終了してください。
-2	セットアップダイアログでキャンセルが押された	セットアップ用データの取り込み時にユーザーがダイアログでキャンセルを押しました。JVClose を呼び出して取り込み処理を終了してください。
-111	<i>dataspec</i> パラメータが不正	パラメータの渡し方かパラメータの内容に問題があると思われます。サンプルプログラム等を参照し、正しくパラメータが JV-Link に渡っているか確認してください。
-112	<i>fromtime</i> パラメータが不正 (読み出し開始ポイント時刻不正)	パラメータ(読み出し開始ポイント時刻)の渡し方かパラメータ(読み出し開始ポイント時刻)の内容に問題があると思われます。サンプルプログラム等を参照し、正しくパラメータが JV-Link に渡っているか確認してください。
-113	<i>fromtime</i> パラメータが不正 (読み出し終了ポイント時刻不正)	パラメータ(読み出し終了ポイント時刻)の渡し方かパラメータ(読み出し終了ポイント時刻)の内容に問題があると思われます。サンプルプログラム等を参照し、正しくパラメータが JV-Link に渡っているか確認してください。
-114	<i>key</i> パラメータが不正	パラメータの渡し方かパラメータの内容に問題があると思われます。サンプルプログラム等を参照し、正しくパラメータが JV-Link に渡っているか確認してください。
-115	<i>option</i> パラメータが不正	同上
-116	<i>dataspec</i> と <i>option</i> の組み合わせが不正	同上
-201	JVInitが行なわれていない	JVOpen/JVRTOpen に先立って JVInit が呼ばれていないと思われます。必ず JVInit を先に呼び出してください。
-202	前回の JVOpen/JVRTOpen/JVMVOpen に対して JVClose が呼ばれていない(オープン中)	前回呼び出した JVOpen/JVRTOpen/JVMVOpen が JVClose によってクローズされていないと思われます。JVOpen/JVRTOpen/JVMVOpen を呼び出した後は次に呼び出すまでの間に JVClose を必ず呼び出してください。
-211	レジストリ内容が不正(レジストリ内容が不正に変更された)	JV-Link はレジストリに値をセットする際に値のチェックを行います(例えば利用キーの桁数など)が、レジストリから値を読み出して使用する際に問題が発生するとこのエラーが発生します。レジストリが直接書き換えられたなどの状況が考えられない場合には JRA-VAN へご連絡ください。

JV-Linkインターフェース仕様書

—301	認証エラー	利用キーが正しくない。あるいは複数のマシンで同一利用キーを使用した場合に発生します。複数のマシンで同じ利用キーをしようした場合には、このエラーが発生したマシンのJV-Linkアンインストールし、再インストール後、利用キーの再発行が必要となります。
—302	利用キーの有効期限切れ	Data Lab.サービスの有効期限が切れています。サービス権の自動延長が停止していると思われます。解消するにはサービス権の再購入が必要です。現在ソフト作者様に配布している利用キーではこのエラーは発生しません。
—303	利用キーが設定されていない(利用キーが空値)	利用キーを設定していないと思われます。JV-Link インストール直後は利用キーが空なので必ず設定する必要があります。
—305	利用規約に同意していない	利用規約に同意していないため、JVOpen 処理で蓄積系データを取得することができません。利用規約同意画面にて利用規約を一読し同意してください。
—401	JV-Link 内部エラー	JV-Link 内部でエラーが発生したと思われます。JRA-VAN へご連絡ください。
—411	サーバーエラー (HTTP ステータス 404 NotFound)	レジストリが直接変更されたか、Data Lab.用サーバーに問題が発生したと思われます。JRA-VAN のメンテナンス中でない場合で、このエラーが続く場合は JRA-VAN へご連絡ください。
—412	サーバーエラー (HTTP ステータス 403 Forbidden)	Data Lab.用サーバーに問題が発生したと思われます。このエラーが続く場合は JRA-VAN へご連絡ください。
—413	サーバーエラー(HTTP ステータス 200,403,404 以上)	同上
—421	サーバーエラー(サーバーの応答が不正)	同上
—431	サーバーエラー(サーバーアプリケーション内部エラー)	同上
—501	セットアップ処理においてスタートキット(CD/DVD-ROM)が無効	JRA-VAN が提供した正しいスタートキット(CD/DVD-ROM)をセットしていないと思われます。正しいスタートキット(CD/DVD-ROM)をセットしてください。  スタートキット (CD/DVD-ROM)の提供は 2022 年3月をもちまして終了いたしました。
—504	サーバーメンテナンス中	サーバーがメンテナンス中です。

JVStatus

戻り値	意味	原因と対処
0以上	正常(ダウンロード済みファイル数)	
—201	JVInitが行なわれていない	JVStatus に先立って JVInit が呼ばれていないと思われます。必ず JVInit を先に呼び出してください。
—203	JVOpenが行なわれていない	JVStatus に先立って JVOpen が呼ばれていないと思われます。必ず JVOpen を先に呼び出してください。

-502	ダウンロード失敗(通信エラーやディスクエラーなど)	ダウンロード処理に失敗しました。エラーの原因を除去しないかぎり解決しないと思われます。原因を除去できたら JVClose を呼び出し、JVOpen からの処理をやりなおしてください。サーバーが混雑している場合のタイムアウトでもこの戻り値が返されることがあります。
------	---------------------------	---

## JVRead/JVGets

戻り値	意味	原因と対処
0以上 0	正常(バッファにセットしたデータのサイズ) 全ファイル読み込み終了(EOF)	JVOpen で取得した全てのデータの終わりを示しています。読み込み処理を終了してください。
-1	ファイル切り替わり	エラーではありません。物理ファイルの終わりを示しています。バッファにはデータが返されませんのでそのまま読み込み処理を続行してください。
-3	ファイルダウンロード中	読み出そうとするファイルがダウンロードの最中です。少し待ってから読み込みを再開してください。
-201	JVInitが行なわれていない	JVRead/JVGets に先立って JVInit/JVOpen が呼ばれていないと思われます。必ず JVInit/JVOpen を先に呼び出してください。
-202	前回の JVOpen/JVRTOpen/JVMVOpen に対して JVClose が呼ばれていない(オープン中)	前回呼び出した JVOpen/JVRTOpen/JVMVOpen が JVClose によってクローズされていないと思われます。JVOpen/JVRTOpen/JVMVOpen を呼び出した後、別の Open を呼び出す場合、先に JVClose を呼び出してください。
-203	JVOpenが行なわれていない	JVRead/JVGets に先立って JVOpen が呼ばれていないと思われます。必ず JVOpen を先に呼び出してください。
-402	ダウンロードしたファイルが異常(ファイルサイズ=0)	ダウンロード中に何らかの問題が発生しファイルが異常な状態になったと思われます。JVFiledelete で該当ファイル(JVRead/JVGets から戻されたファイル名)を削除し、再度 JVOpen からの処理をやりなおしてください。
-403	ダウンロードしたファイルが異常(データ内容)	同上
-502	ダウンロード失敗(通信エラーやディスクエラーなど)	ダウンロード処理に失敗しました。エラーの原因を除去しないかぎり解決しないと思われます。原因を除去できたら JVClose を呼び出し、JVOpen からの処理をやりなおしてください。サーバーが混雑している場合のタイムアウトでもこの戻り値が返されることがあります。
-503	ファイルが見つからない	JVOpen から JVRead/JVGets までの間に読み出すべきファイルが削除された、または該当ファイルが使用中と思われます。JVOpen からやりなおせば解消しますが、削除された原因を除去する必要があります。

## JVClose

戻り値	意味	原因と対処
0	正常	

## JVFiledelete

戻り値	意味	原因と対処
0	正常	
-201	JVInitが行なわれていない	JVFiledelete に先立って JVInit が呼ばれていないと思われる。必ず JVInit を先に呼び出してください。
-503	ファイルが見つからない	指定されたファイルが見つかりません。他のソフトから削除された可能性があります。ファイルの削除が目的ですから、そのまま処理を続行してください。

## JVFukuFile

戻り値	意味	原因と対処
0	正常	
-1	該当データ無し	pattern パラメータから、勝負服画像を作成することができませんでした。(No Image 画像を出力しました。)
-118	filepath パラメータが不正	filepath パラメータの渡し方かパラメータの内容に問題があるか、または、filepath パラメータで設定されたフォルダが存在していません。正しくパラメータが JV-Link に渡っているか確認してください。
-201	JVInitが行なわれていない	JVFukuFile に先立って JVInit が呼ばれていないと思われる。必ず JVInit を先に呼び出してください。
-211	レジストリ内容が不正(レジストリ内容が不正に変更された)	JV-Link はレジストリに値をセットする際に値のチェックを行います(例えば利用キーの桁数など)が、レジストリから値を読み出して使用する際に問題が発生するとこのエラーが発生します。レジストリが直接書き換えられたなどの状況が考えられない場合には JRA-VAN へご連絡ください。
-301	認証エラー	利用キーが正しくない。あるいは複数のマシンで同一利用キーを使用した場合に発生します。複数のマシンで同じ利用キーをしようした場合には、このエラーが発生したマシンの JV-Link をアンインストールし、再インストール後、利用キーの再発行が必要となります。
-302	利用キーの有効期限切れ	Data Lab.サービスの有効期限が切れています。サービス権の自動延長が停止していると思われる。解消するにはサービス権の再購入が必要です。現在ソフト作者様に配布している利用キーではこのエラーは発生しません。
-303	利用キーが設定されていない(利用キーが空値)	利用キーを設定していないと思われる。JV-Link インストール直後は利用キーが空なので必ず設定する必要があります。
-401	JV-Link 内部エラー	JV-Link 内部でエラーが発生したと思われる。JRA-VAN へご連絡ください。

-411	サーバーエラー (HTTP ステータス 404 NotFound)	レジストリが直接変更されたか、Data Lab.用サーバーに問題が発生したと思われます。JRA-VAN のメンテナンス中でない場合で、このエラーが続く場合は JRA-VAN へご連絡ください。
-412	サーバーエラー (HTTP ステータス 403 Forbidden)	Data Lab.用サーバーに問題が発生したと思われま す。このエラーが続く場合は JRA-VAN へご連絡くださ い。
-413	サーバーエラー (HTTP ステータス 200,403,404 以外)	同上
-421	サーバーエラー (サーバーの応答が不正)	同上
-431	サーバーエラー (サーバーアプリケーション内部 エラー)	同上
-504	サーバーメンテナンス中	サーバーがメンテナンス中です。

## JVFuku

戻り値	意味	原因と対処
0	正常(公開動画なし)	
-1	該当データ無し	pattern パラメータから、勝負服画像を作成することが できませんでした。(No Image 画像を出力しました。)
-201	JVInitが行なわれていない	JVFuku に先立って JVInit が呼ばれていないと思われ ます。必ず JVInit を先に呼び出してください。
-211	レジストリ内容が不正 (レジストリ内容が不正に 変更された)	JV-Link はレジストリに値をセットする際に値のチェッ クを行います (例えば利用キーの桁数など) が、レジ ストリから値を読み出して使用する際に問題が発生 するとこのエラーが発生します。レジストリが直接書き 換えられたなどの状況が考えられない場合には JRA-VAN へご連絡ください。
-301	認証エラー	利用キーが正しくない。あるいは複数のマシンで同一 利用キーを使用した場合に発生します。複数のマシ ンで同じ利用キーをしようした場合には、このエラー が発生したマシンの JV-Link をアンインストールし、再 インストール後、利用キーの再発行が必要となります ます。
-302	利用キーの有効期限切れ	Data Lab.サービスの有効期限が切れています。サー ビス権の自動延長が停止していると思われま す。解消するにはサービス権の再購入が必要です。現在ソ フト作者様に配布している利用キーではこのエラーは 発生しません。
-303	利用キーが設定されていない (利用キーが空 値)	利用キーを設定していないと思われま す。JV-Link イ ンストール直後は利用キーが空なので必ず設定する 必要があります。
-401	JV-Link 内部エラー	JV-Link 内部でエラー が発生したと思われま す。 JRA-VAN へご連絡ください。
-411	サーバーエラー (HTTP ステータス 404 NotFound)	レジストリが直接変更されたか、Data Lab.用サーバ ーに問題が発生したと思われま す。JRA-VAN のメン テナンス中でない場合で、このエラーが続く場合は JRA-VAN へご連絡ください。
-412	サーバーエラー (HTTP ステータス 403 Forbidden)	Data Lab.用サーバーに問題が発生したと思われま す。このエラーが続く場合は JRA-VAN へご連絡くだ さい。
-413	サーバーエラー (HTTP ステータス 200,403,404 以外)	同上
-421	サーバーエラー (サーバーの応答が不正)	同上

-431	サーバーエラー(サーバーアプリケーション内部エラー)	同上
-504	サーバーメンテナンス中	サーバーがメンテナンス中です。

## JVMVCheck/ JVMVCheckWithType

戻り値	意味	原因と対処
0	正常(公開動画なし)	
1	正常(公開動画あり)	
-1	該当データ無し	指定されたパラメータに合致するデータがサーバーに存在しない。
-111	<i>movietype</i> パラメータが不正	パラメータの渡し方かパラメータの内容に問題があると思われます。サンプルプログラム等を参照し、正しくパラメータが JV-Link に渡っているか確認してください。
-114	<i>key</i> パラメータが不正	パラメータの渡し方かパラメータの内容に問題があると思われます。サンプルプログラム等を参照し、正しくパラメータが JV-Link に渡っているか確認してください。
-201	JVInitが行なわれていない	JVMVCheckに先立ってJVInitが呼ばれていないと思われます。必ずJVInitを先に呼び出してください。
-202	前回のJVOpen/JVRTOpen/JVMVOpenに対してJVCloseが呼ばれていない(オープン中)	前回呼び出したJVOpen/JVRTOpen/JVMVOpenがJVCloseによってクローズされていないと思われます。JVOpen/JVRTOpen/JVMVOpenを呼び出した後は次に呼び出すまでの間にJVCloseを必ず呼び出してください。
-211	レジストリ内容が不正(レジストリ内容が不正に変更された)	JV-Linkはレジストリに値をセットする際に値のチェックを行います(例えば利用キーの桁数など)が、レジストリから値を読み出して使用する際に問題が発生するとこのエラーが発生します。レジストリが直接書き換えられたなどの状況が考えられない場合にはJRA-VANへご連絡ください。
-301	認証エラー	利用キーが正しくない。あるいは複数のマシンで同一利用キーを使用した場合に発生します。複数のマシンで同じ利用キーをしようした場合には、このエラーが発生したマシンのJV-Linkをアンインストールし、再インストール後、利用キーの再発行が必要となります。
-302	利用キーの有効期限切れ	Data Lab.サービスの有効期限が切れています。サービス権の自動延長が停止していると思われます。解消するにはサービス権の再購入が必要です。現在ソフト作者様に配布している利用キーではこのエラーは発生しません。
-303	利用キーが設定されていない(利用キーが空値)	利用キーを設定していないと思われます。JV-Linkインストール直後は利用キーが空なので必ず設定する必要があります。
-401	JV-Link 内部エラー	JV-Link 内部でエラーが発生したと思われます。JRA-VANへご連絡ください。
-411	サーバーエラー (HTTPステータス 404 NotFound)	レジストリが直接変更されたか、Data Lab.用サーバーに問題が発生したと思われます。JRA-VANのメンテナンス中でない場合で、このエラーが続く場合はJRA-VANへご連絡ください。

-412	サーバーエラー (HTTP ステータス 403 Forbidden)	Data Lab.用サーバーに問題が発生したと思われ ません。このエラーが続く場合は JRA-VAN へご 連絡ください。
-413	サーバーエラー (HTTP ステータス 200,403,404 以	同上
-421	サーバーエラー (サーバーの応答が不正)	同上
-431	サーバーエラー (サーバーアプリケーション内部 エラー)	同上
-504	サーバーメンテナンス中	サーバーがメンテナンス中です。

## JVMVPlay / JVMVPlayWithType

戻り値	意味	原因と対処
0	正常	
-1	該当データ無し	指定されたパラメータに合致するレース映像が公開 されていない。JVMVCheck を利用しレース映像の公開 状況を確認してください。
-100	パラメータが不正あるいはレジストリへの保 存に失敗	内部的なエラーが発生したと思われ ます。また、レジ ストリへのアクセス権限の問題で設定内容のレジス トリへの反映に失敗した場合もこのエラーとなり ます。レジストリのアクセス権限に問題が無い場 合は JRA-VAN へご連絡ください。
-111	<i>movietype</i> パラメータが不正	パラメータの渡し方かパラメータの内容に問題があ ると思われ ます。サンプルプログラム等を参照し、正しく パラメータが JV-Link に渡っているか確認してく ださい。
-114	<i>key</i> パラメータが不正	パラメータの渡し方かパラメータの内容に問題があ ると思われ ます。サンプルプログラム等を参照し、正しく パラメータが JV-Link に渡っているか確認してく ださい。
-201	JVInitが行なわれていない	JVMVPlay/JVMVPlayWithType に先立って JVInit が呼 ばれていないと思われ ます。必ず JVInit を先に呼び出 してください。
-202	前回の JVOpen/JVRTOpen/JVMVOpen に対 して JVClose が呼ばれていない (オープン中)	前回呼び出した JVOpen/JVRTOpen/JVMVOpen が JVClose によってクローズされていないと思われ ます。JVOpen/JVRTOpen/JVMVOpen を呼び出した後は 次に呼び出すまでの間に JVClose を必ず呼び出してく ださい。
-211	レジストリ内容が不正 (レジストリ内容が不正に 変更された)	JV-Link はレジストリに値をセットする際に値のチェ ックを行います (例えば利用キーの桁数など) が、レジ ストリから値を読み出して使用する際に問題が発生す るとこのエラーが発生します。レジストリが直接書き換 えられたなどの状況が考えられない場合には JRA- VAN へご連絡ください。
-301	認証エラー	利用キーが正しくない。あるいは複数のマシンで同一 利用キーを使用した場合に発生します。複数のマシ ンで同じ利用キーをしようとした場合には、このエ ラーが発生したマシンの JV-Link をアンインストールし、再 インストール後、利用キーの再発行が必要となります。
-302	利用キーの有効期限切れ	Data Lab.サービスの有効期限が切れています。サー ビス権の自動延長が停止していると思われ ます。解消するにはサービス権の再購入が必要 です。現在ソフト作者様に配布している利用キー ではこのエラーは発生しません。

-303	利用キーが設定されていない(利用キーが空値)	利用キーを設定していないと思われます。JV-Link インストール直後は利用キーが空なので必ず設定する必要があります。
-304	JRA レーシングビューアー連携機能認証エラー	JRA レーシングビューアー連携機能利用申請が行われていないと思われます。 JVMVPlay/JVMVPlayWithTypeを使用するには、利用申請が必要になります。詳細については JRA-VAN ホームページを参照下さい。
-401	JV-Link 内部エラー	JV-Link 内部でエラーが発生したと思われます。JRA-VAN へご連絡ください。
-411	サーバーエラー (HTTP ステータス 404 NotFound)	レジストリが直接変更されたか、Data Lab.用サーバーに問題が発生したと思われます。JRA-VAN のメンテナンス中でない場合で、このエラーが続く場合は JRA-VAN へご連絡ください。
-412	サーバーエラー (HTTP ステータス 403 Forbidden)	Data Lab.用サーバーに問題が発生したと思われます。このエラーが続く場合は JRA-VAN へご連絡ください。
-413	サーバーエラー (HTTP ステータス 200,403,404 以外)	同上
-421	サーバーエラー (サーバーの応答が不正)	同上
-431	サーバーエラー (サーバーアプリケーション内部エラー)	同上
-504	サーバーメンテナンス中	サーバーがメンテナンス中です。

## JVMVOpen

戻り値	意味	原因と対処
0	正常	
-1	該当データ無し	指定されたパラメータに合致する新しいデータがサーバーに存在しない。又は、最新バージョンが公開され、ユーザーが最新バージョンのダウンロードを選択しました。JVClose を呼び出して取り込み処理を終了してください。
-111	<i>movietype</i> パラメータが不正	パラメータの渡し方かパラメータの内容に問題があると思われます。サンプルプログラム等を参照し、正しくパラメータが JV-Link に渡っているか確認してください。
-114	<i>searchkey</i> パラメータが不正	同上
-201	JVInitが行なわれていない	JVMVOpen に先立って JVInit が呼ばれていないと思われます。必ず JVInit を先に呼び出してください。
-202	前回の JVOpen/JVRTOpen/JVMVOpen に対して JVClose が呼ばれていない(オープン中)	前回呼び出した JVOpen/JVRTOpen/JVMVOpen が JVClose によってクローズされていないと思われます。JVOpen/JVRTOpen/JVMVOpen を呼び出した後は次に呼び出すまでの間に JVClose を必ず呼び出してください。
-211	レジストリ内容が不正(レジストリ内容が不正に変更された)	JV-Link はレジストリに値をセットする際に値のチェックを行います(例えば利用キーの桁数など)が、レジストリから値を読み出して使用する際に問題が発生するとこのエラーが発生します。レジストリが直接書き換えられたなどの状況が考えられない場合には JRA-VAN へご連絡ください。

-301	認証エラー	利用キーが正しくない。あるいは複数のマシンで同一利用キーを使用した場合に発生します。複数のマシンで同じ利用キーをしようした場合には、このエラーが発生したマシンのJV-Linkをアンインストールし、再インストール後、利用キーの再発行が必要となります。
-302	利用キーの有効期限切れ	Data Lab.サービスの有効期限が切れています。サービス権の自動延長が停止していると思われます。解消するにはサービス権の再購入が必要です。現在ソフト作者様に配布している利用キーではこのエラーは発生しません。
-303	利用キーが設定されていない(利用キーが空値)	利用キーを設定していないと思われます。JV-Link インストール直後は利用キーが空なので必ず設定する必要があります。
-304	JRA レーシングビューア連携機能認証エラー	JRA レーシングビューア連携機能利用申請が行われていないと思われます。 JVMVPlay/JVMVPlayWithTypeを使用するには、利用申請が必要になります。詳細についてはJRA-VAN ホームページを参照下さい。
-401	JV-Link 内部エラー	JV-Link 内部でエラーが発生したと思われます。JRA-VAN へご連絡ください。
-411	サーバーエラー (HTTP ステータス 404 NotFound)	レジストリが直接変更されたか、Data Lab.用サーバーに問題が発生したと思われます。JRA-VAN のメンテナンス中でない場合で、このエラーが続く場合はJRA-VAN へご連絡ください。
-412	サーバーエラー (HTTP ステータス 403 Forbidden)	Data Lab.用サーバーに問題が発生したと思われます。このエラーが続く場合は JRA-VAN へご連絡ください。
-413	サーバーエラー (HTTP ステータス 200,403,404 以外)	同上
-421	サーバーエラー (サーバーの応答が不正)	同上
-431	サーバーエラー (サーバーアプリケーション内部エラー)	同上
-504	サーバーメンテナンス中	サーバーがメンテナンス中です。

## JVMVRead

戻り値	意味	原因と対処
0以上 0	正常(バッファにセットしたデータのサイズ) 全ファイル読み込み終了 (EOF)	JVMVOpen で取得した全てのデータの終わりを示しています。読み込み処理を終了してください。
-3	データダウンロード中	読み出そうとするデータがダウンロードの最中です。少し待ってから読み込みを再開してください。
-201	JVInit が行なわれていない	JVMVRead に先立って JVInit/JVMVOpen が呼ばれていないと思われます。必ず JVInit/JVMVOpen を先に呼び出してください。
-203	JVMVOpen が行なわれていない	JVMVRead に先立って JVMVOpen が呼ばれていないと思われます。必ず JVMVOpen を先に呼び出してください。
-402	ダウンロードしたデータが異常(サイズ=0)	ダウンロード中に何らかの問題が発生しデータが異常な状態になったと思われます。再度 JVMVOpen からの処理をやりなおしてください。

-403	ダウンロードしたデータが異常(データ内容)	同上
-502	ダウンロード失敗(通信エラーやディスクエラーなど)	ダウンロード処理に失敗しました。エラーの原因を除去しないかぎり解決しないと思われます。原因を除去できたら JVClose を呼び出し、JVMVOpen からの処理をやりなおしてください。サーバーが混雑している場合のタイムアウトでもこの戻り値が返されることがあります。
-503	一時ファイルが見つからない	JVMVOpen から JVMVRead までの間に、処理で使用する一時ファイルが削除された、または該当ファイルが使用中と思われます。JVMVOpen からやりなおせば解消しますが、削除された原因を除去する必要があります。

## JVCourseFile

戻り値	意味	原因と対処
0	正常	
-1	該当データ無し	各パラメータで設定されたレースが存在しません。(No Image 画像を出力しました。)
-114	key パラメータが不正	パラメータの渡し方かパラメータの内容に問題があります。正しくパラメータがJV-Linkに渡っているか確認してください。
-201	JVInit が行なわれていない	JVCourseFile に先立って JVInit が呼ばれていないと思われます。必ず JVInit を先に呼び出してください。
-202	前回の JVOpen/JVRTOpen/JVMVOpen に対して JVClose が呼ばれていない(オープン中)	前回呼び出した JVOpen/JVRTOpen/JVMVOpen が JVClose によってクローズされていないと思われます。JVOpen/JVRTOpen/JVMVOpen を呼び出した後は次に呼び出すまでの間に JVClose を必ず呼び出して下さい。
-211	レジストリ内容が不正(レジストリ内容が不正に変更された)	JV-Link はレジストリに値をセットする際に値のチェックを行います(例えば利用キーの桁数など)が、レジストリから値を読み出して使用する際に問題が発生するとこのエラーが発生します。レジストリが直接書き換えられたなどの状況が考えられない場合には JRA-VAN へご連絡ください。
-301	認証エラー	利用キーが正しくない。あるいは複数のマシンで同一利用キーを使用した場合に発生します。複数のマシンで同じ利用キーをしようした場合には、このエラーが発生したマシンの JV-Link をアンインストールし、再インストール後、利用キーの再発行が必要となります。
-302	利用キーの有効期限切れ	Data Lab.サービスの有効期限が切れています。サービス権の自動延長が停止していると思われます。解消するにはサービス権の再購入が必要です。現在ソフト作者様に配布している利用キーではこのエラーは発生しません。
-303	利用キーが設定されていない(利用キーが空値)	利用キーを設定していないと思われます。JV-Link インストール直後は利用キーが空なので必ず設定する必要があります。
-401	JV-Link 内部エラー	JV-Link 内部でエラーが発生したと思われます。JRA-VAN へご連絡ください。

JV-Linkインターフェース仕様書

-411	サーバーエラー (HTTP ステータス 404 NotFound)	レジストリが直接変更されたか、Data Lab.用サーバーに問題が発生したと思われます。JRA-VAN のメンテナンス中でない場合で、このエラーが続く場合は JRA-VAN へご連絡ください。
-412	サーバーエラー (HTTP ステータス 403 Forbidden)	Data Lab.用サーバーに問題が発生したと思われま す。このエラーが続く場合は JRA-VAN へご連絡くださ い。
-413	サーバーエラー (HTTP ステータス 200,403,404 以外)	同上
-421	サーバーエラー (サーバーの応答が不正)	同上
-431	サーバーエラー (サーバーアプリケーション内部 エラー)	同上
-502	ダウンロード失敗 (通信エラーやディスクエラー など)	ダウンロード処理に失敗しました。エラーの原因を除 去しないかぎり解決しないと思われま す。エラーの原因として、画像ファイルの書き込みに失敗した事など があげられます。また、サーバーが混雑している場 合のタイムアウトでもこの戻り値が返されることがあ ります。
-504	サーバーメンテナンス中	サーバーがメンテナンス中です。

JVCourseFile2

戻り値	意味	原因と対処
0	正常	
-1	該当データ無し	各パラメータで設定されたレースが存在しません。 (No Image 画像を出力しました。)
-114	key パラメータが不正	パラメータの渡し方かパラメータの内容に問題があり ます。正しくパラメータが JV-Link に渡っているか確認 してください。
-118	filepath パラメータが不正	filepath パラメータの渡し方かパラメータの内容に 問題があるか、または、filepath パラメータで設定さ れたフォルダが存在していません。正しくパラメー タが JV-Link に渡っているか確認してください。
-201	JVInitが行なわれていない	JVCourseFile に先立って JVInit が呼ばれていないと 思われます。必ず JVInit を先に呼び出してください。
-202	前回の JVOpen/JVRTOpen/JVMVOpen に対し て JVClose が呼ばれていない (オープン中)	前回呼び出した JVOpen/JVRTOpen/JVMVOpen が JVClose によってクローズされていないと思わ れます。JVOpen/JVRTOpen/JVMVOpen を呼び出し た後 は次に呼び出すまでの間に JVClose を必ず呼 び出してください。
-211	レジストリ内容が不正 (レジストリ内容が不正に 変更された)	JV-Link はレジストリに値をセットする際に値のチェッ クを行います (例えば利用キーの桁数など) が、レジ ストリから値を読み出して使用する際に問題が発生 するとこのエラーが発生します。レジストリが直接書き 換えられたなどの状況が考えられない場合には JRA-VAN へご連絡ください。
-301	認証エラー	利用キーが正しくない。あるいは複数のマシンで同一 利用キーを使用した場合に発生します。複数のマシ ンで同じ利用キーをしようした場合には、このエラー が発生したマシンの JV-Link をアンインストールし、再 インストール後、利用キーの再発行が必要となります。

## JV-Linkインターフェース仕様書

-302	利用キーの有効期限切れ	Data Lab.サービスの有効期限が切れています。サービス権の自動延長が停止していると思われます。解消するにはサービス権の再購入が必要です。現在ソフト作者様に配布している利用キーではこのエラーは発生しません。
-303	利用キーが設定されていない(利用キーが空値)	利用キーを設定していないと思われます。JV-Link インストール直後は利用キーが空なので必ず設定する必要があります。
-401	JV-Link 内部エラー	JV-Link 内部でエラーが発生したと思われます。JRA-VAN へご連絡ください。
-411	サーバーエラー (HTTP ステータス 404 NotFound)	レジストリが直接変更されたか、Data Lab.用サーバーに問題が発生したと思われます。JRA-VAN のメンテナンス中でない場合で、このエラーが続く場合は JRA-VAN へご連絡ください。
-412	サーバーエラー (HTTP ステータス 403 Forbidden)	Data Lab.用サーバーに問題が発生したと思われます。このエラーが続く場合は JRA-VAN へご連絡ください。
-413	サーバーエラー(HTTP ステータス 200,403,404 以外)	同上
-421	サーバーエラー(サーバーの応答が不正)	同上
-431	サーバーエラー(サーバーアプリケーション内部エラー)	同上
-502	ダウンロード失敗(通信エラーやディスクエラーなど)	ダウンロード処理に失敗しました。エラーの原因を除去しないかぎり解決しないと思われます。サーバーが混雑している場合のタイムアウトでもこの戻り値が返されることがあります。
-504	サーバーメンテナンス中	サーバーがメンテナンス中です。

## JVWatchEvent

戻り値	意味	原因と対処
0	正常	
-201	JVInitが行なわれていない	JVWatchEvent に先立って JVInit が呼ばれていないと思われます。必ず JVInit を先に呼び出してください。

## JVWatchEventClose

戻り値	意味	原因と対処
0	正常	