



# SVF

***SVF Connect for Java API***  
***Ver.8.0***

**ユーザーズ・マニュアル**

**ウイングアーク テクノロジーズ 株式会社**  
**株式会社エフ・アイ・ティ**



## はじめに

この度は、SVF Connect for Java APIをお買い上げいただき、誠にありがとうございます。  
SVF Connect for Java APIは、JavaアプリケーションプログラムでSVF実行部（Java版、C版）を使用してPDFファイルの作成、プリンタ出力、Report Director EnterpriseやReport Directorへのスプールをおこなう場合に、Javaアプリケーションプログラムを使用するアプリケーションサーバとSVF実行サーバ（PDF作成、印刷実行、スプール管理）を別筐体で使用するために必要な製。

このマニュアルは、SVF Connect for Java APIを円滑に導入するためのガイドとして、そのセットアップ方法や使用方法が書かれています。

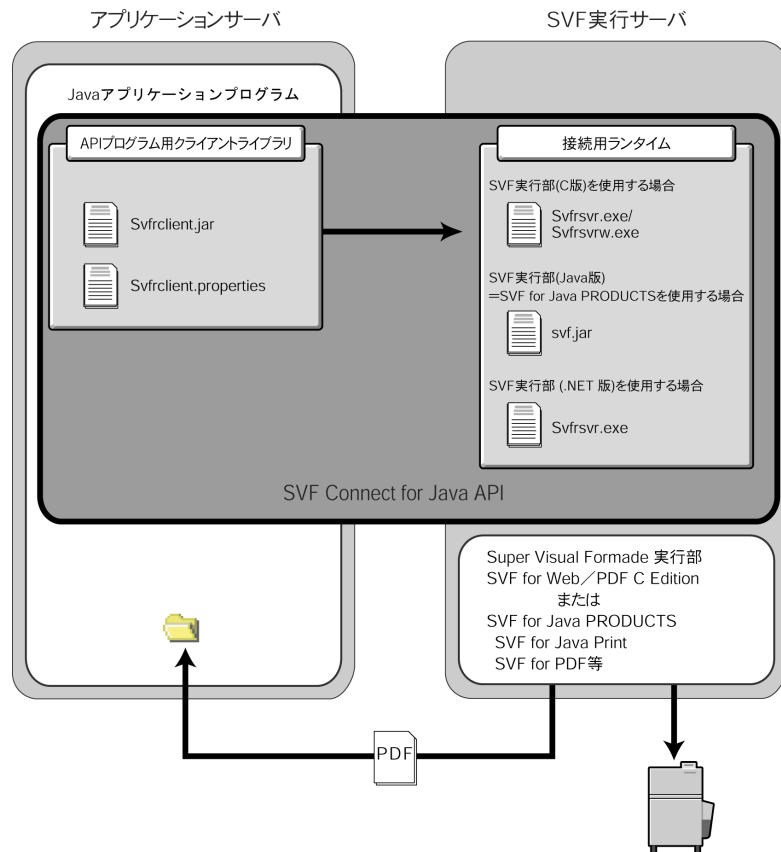
製品の動作環境をご確認の上、ご使用いただきますようお願い致します

<b>1 製品概要</b>	<b>1</b>
ファイル構成 .....	2
動作環境 .....	3
<b>2 セットアップと起動方法</b>	<b>4</b>
アプリケーションサーバでの設定 .....	4
Java プログラム用クラスライブラリのセットアップ .....	4
Java プログラム用クラスライブラリ側オプション .....	4
SVF 実行サーバでの設定 .....	5
SVF 実行部 (C 版) 接続用ランタイム .....	5
SVF 実行部 (C 版) 用ランタイム起動オプション .....	6
SVF 実行部 (Java 版) 接続用ランタイム .....	7
SVF 実行部 (Java 版) 用ランタイム起動オプション .....	7
SVF 実行部 (.NET 版) 接続用ランタイム .....	9
SVF 実行部 (.NET 版) 接続用ランタイム起動オプション .....	9
<b>3 サービスへの登録方法</b>	<b>11</b>
run ファイルの作成 .....	11
AppLauncher.exe の起動 .....	11
アプリケーションの設定 .....	12
ライブラリの設定 .....	13
サービスの設定 .....	13
設定の保存 .....	14
動作確認 .....	14
svf.jar の起動 .....	14
svf.jar の停止 .....	15
サービスへの登録 .....	15
サービス登録の解除 .....	16
<b>4 サンプルプログラム</b>	<b>17</b>
クエリー/レポートライタータイプ-「Sample.java」 .....	17
固定帳票タイプ-「CHAIN.java」 .....	22
<b>5 注意事項</b>	<b>27</b>
<b>索引</b>	<b>索引-1</b>

# 1 製品概要

SVF Connect for Java APIは、JavaアプリケーションプログラムでSVF実行部（Java版、C版）を使用してPDFファイルの作成、プリンタ出力、Report Director EnterpriseやReport Directorへのスプールをおこなう場合に、Javaアプリケーションプログラムを使用するアプリケーションサーバとSVF実行サーバ（PDF作成、印刷実行、スプール管理）を別筐体で使用するために必要な製品です。

「svfrclient.jar」は、Javaのマルチスレッド環境での実行をサポートし、Servlet等のWeb環境に適しています。また、このファイルをインポートすることにより簡単にインターフェースを取る事ができます。








- PDFを作成する場合、SVF Connect for Java APIでは作成したPDFを呼び出し元であるアプリケーションサーバに戻します。そのため、PDFファイルの保存先フォルダは、アプリケーションサーバからのパスを指定する必要があります。様式ファイルやクエリーファイルの指定は、SVF実行部からのパスを指定します。

## ファイル構成

提供されるソフトウェアは次のファイル構成となっています。

### SVF Connect for Java API




Javaプログラム用クラスライブラリ

-  svfrclient.jar
-  svfrclient.properties
-  sample/sample.frm
-  sample/Sample.java
-  sample/sample.vrq

Java用サンプル                      Java言語用メソッドリスト

クエリー/レポートライタータイプ

(サブフォーム・レコードが設計されている様式ファイル使用)

-  Sample.java      Java言語用コーディングサンプル
-  Sample.frm      サンプル様式ファイル
-  Sample.vrq      サンプルクエリーファイル



- サーバ起動説明ファイルはVer7以降には含まれていません。SVF Connect Service 起動用runファイル、およびスクリプトファイルはSVF Java Print、およびSVF PDFに同梱されています。
- 各SVF実行部接続用ランタイムは、各SVF実行部に含まれています。本製品には含まれません。

## 動作環境

OS	Microsoft® Windows Server™2003 Standard Edition、Enterprise Edition(x86互換 32bit、EM64T 64bit) Microsoft® Windows Server™2003 R2 Standard Edition、Enterprise Edition(x86互換 32bit、EM64T 64bit) SUN®Solaris®、9 (SPARC)、Solaris® 10 (SPARC) HP®HP/UX11i V2 (IPF)、 IBM®AIX5L Ver5.3、 Red Hat Enterprise Linux AS4.0(x86互換 32bit、EM64T 64bit) Red Hat Enterprise Linux ES4.0(x86互換 32bit) Red Hat Enterprise Linux 5.0 Advanced Platform (32bit) Red Hat Enterprise Linux 5.0 (Server 32bit)
動作可能なJDK	Sun Microsystems J2EEおよびJ2SE準拠JDK1.4/1.5 IBM J2EEおよびJ2SE準拠 JDK1.4/1.5 HP J2EEおよびJ2SE準拠 JDK1.4/1.5
SVF実行部(C版) 接続用ランタイム	SVF for Web/PDF C Edition Ver.6.3 SVF実行部(C版) Ver.6.3
SVF実行部(Java版) 接続用ランタイム	SVF Java PRODUCTS
SVF for .NET Framework 接続用ランタイム	SVF for .NET Framework Ver.1.1、2.01、2.1



- このプログラムを使用する場合には、TCP/IP接続が可能である必要があります。
- 上記以外の環境につきましては、あらかじめ問い合わせください。

# 2

## セットアップと起動方法

ここでは、SVF Connect for Java APIのセットアップについてアプリケーションサーバ上で設定、SVF実行サーバ上での設定に分けて説明します。

### アプリケーションサーバでの設定

Javaアプリケーションが稼動するアプリケーションサーバでは、SVF Connect for Java APIのJavaプログラム用クラスライブラリのセットアップをおこないます。

#### ● Java プログラム用クラスライブラリのセットアップ .....

Javaプログラム用クラスライブラリのセットアップは、つぎの手順でおこないます。

#### 1

「svfrclient.jar」および「svfrclient.properties」を同一ディレクトリへ配置します。

#### 2

ファイルの配置先を CLASSPATH に追加します。

Java プログラムから使用する事が可能となります。



● Web Application Serverに組み込んで使用する際、warファイルなどにこれらのモジュールを含めることは動作保証していません。かならず、モジュールをサーバに配置し、Web Application ServerのCLASSPATHに追加して、弊社で提供しているAPIを使用するようにしてください。

#### ● Java プログラム用クラスライブラリ側オプション .....

「svfrclient.properties」ファイル内の設定について記述します。

**Support.Encoding=SJIS,MS932,Cp942,Cp942C,Cp943,Cp943C,UTF8**

サポートする Encoding を記述します。

**Default.Encoding=**

SVF Connect for Java API が受け取るデータの文字コードを指定します。ここで指定できるのは、Support.Encoding に記述されているもののみとなります。何も指定されていない場合は、MS932 が使用されます。

**DefaultSpoolFileName=default.pdf**

スプールファイル名を省略した際のデフォルトファイル名を設定します。

**Remote.retryCount=5**

TCP 接続のリトライ回数を設定します。

**Remote.retryWait=3000**

TCP 接続のリトライ待ち時間(ミリ秒)を設定します。



**Remote.Hosts=localhost**

svfrClient の接続先ホストを示しています。

**Remote.Port=4141**

svfrClient の使用ポートを示しています。

**Remote.RecvTimeout=120**

タイムアウトの間隔(分)を設定します。

**Remote.SendTimeout=15**

SVF Connect for Java API では使用できません。

**Debug.Mode=false**

デバッグの ON/OFF を設定します。デバッグを ON に設定すると、SVF Connect for Java API 側のデバッグログを取得できるようになります。

true ON  
false OFF

**Debug.LogDir=** デバッグログの出力先ディレクトリを指定します。

例 C:/SVFJP

## SVF 実行サーバでの設定

ここでは、SVF 実行サーバで SVF 実行部 (C 版) を使用している場合、SVF 実行部 (Java 版) の場合、SVF 実行部 (.NET 版) の場合の各接続用ランタイムのセットアップ、起動方法、起動オプションについて説明します。

### ● SVF 実行部 (C 版) 接続用ランタイム .....

SVF 実行サーバで、SVF 実行部 (C 版) を使用している場合は、SVF 実行部 (C 版) 接続用ランタイムをセットアップします。セットアップは、つぎの手順でおこないます。

#### 1

SVF 実行部 (C 版) がセットアップされているコンピュータで、svfrsvr.exe を起動します。

```
C:¥WinNT¥svfrsvr.exe (options)
```

以下の環境では、サービス起動にも対応しています。

- ・Windows Server 2003 Standard Edition, Enterprise Edition (x86 互換 32bit)、
  - ・Windows Server 2003 R2 Standard Edition, Enterprise Edition (x86 互換 32bit)
- 詳細については次項「SVF 実行部 (C 版) 用ランタイム起動オプション」を参照してください。

## ● SVF 実行部 (C 版) 用ランタイム起動オプション .....

### Usage: svfrsvr [options]

- P[n] リスポートアドレス[n]を指定します。  
指定されていない場合は「4141」となります。
- B[a] バインドアドレス[a]を指定します。  
指定されていない場合にはデフォルトとなります。
- W[n] クライアント側からのリクエスト受信待ち時間のタイムアウト間隔  
[n](min)を指定します。  
指定されない場合はデフォルトとなります。
- T[n] クライアントへの結果送信開始から送信完了までの最大待ち  
時間[n](min)を指定します。
- L[f] ログファイル名[f]を指定します。  
指定されていない場合、ログは出力されません。
- R サービスに登録します。
- U サービスを解除します
- H ヘルプを表示します。  
この起動オプションの説明が表示されます



#### ● 起動オプションでのログ出力設定例

設定例

```
svfrsvr -L"C:¥log¥ server.log"
```

出力例

```
Visual Formade for Report – Native Report Server
Copyright(C) 2000, Version 6.3.0.1
host address is [10.17.100.212]
bind address is [4141]

Start Listen: [10.17.100.212:4141] Wed Sep 27 16:48:54 2006
Connect Socket: [127.0.0.1:2700] Wed Sep 27 16:55:15 2006
Close Socket: [127.0.0.1:2700] Wed Sep 27 16:55:23 2006
```

svfrsvr.exe (接続用ランタイム) のバージョンによって、出力内容が異なることがあります。接続用ランタイムが起動されているかどうかの確認は、コマンドプロンプトから「netstat -an」コマンドを入力しておこなうことも可能です。ポート「4141」がLISTENING状態になっていれば起動しています。

- サービスへ登録して使用する場合は、「svfrsvrw.exe」を使用してください。

## ● SVF 実行部 (Java 版) 接続用ランタイム .....

SVF 実行サーバで、SVF 実行部 (Java 版) を使用している場合は、SVF 実行部 (Java 版) 接続用ランタイムをセットアップします。セットアップは、つぎの手順でおこないます。

### 1

SVF 実行部 (Java 版) がセットアップされているコンピュータで、「svf.jar」に CLASSPATH を通します。

### 2

「svf.jar」を起動します。

```
java jp.co.fit.vfreport.SvfrServer (options)
```

オプションの詳細については次項「SVF 実行部 (Java 版) 用ランタイム起動オプション」を参照してください。



- Ver7 では「svf.jar」、「svfpdfparser.jar」、「svfgrph.jar」、「xercesImpl.jar」、「xml-apis.jar」、「jai\_codec.jar」、「jai\_core.jar」を CLASSPATH に通してください。  
Ver8 では「svf.jar」、「svfpdfparser.jar」、「svfgrph.jar」、「jai\_codec.jar」、「jai\_core.jar」、「forme.jar」を CLASSPATH に通してください。  
詳細については、弊社サポート窓口までお問い合わせください。

## ● SVF 実行部 (Java 版) 用ランタイム起動オプション .....

Usage: java jp.co.fit.vfreport.SvfrServer [options]

- P[n] リスポートアドレス[n]を指定します。  
指定されていない場合は「4141」となります。
- B[a] バインドアドレス[a]を指定します。  
指定されていない場合はデフォルトとなります。
- W[n] クライアント側からのリクエスト受信待ち時間のタイムアウト間隔 [n](min)を指定します。  
指定されていない場合はデフォルトとなります。
- L[f] ログファイル名[f]を指定します。  
指定されていない場合はログは出力されません。
- G ガベージコレクションを実行しません
- G[n] ガベージコレクションを実行する間隔[n](ミリ秒)を指定します。
- X 致命的なエラー(Error 派生クラス)が発生した場合、ログに "server stopped" を書き込み、SvfrServer を正常終了します。
- H ヘルプを表示します。  
この起動オプションの説明が表示されます。



- 起動オプションは、ご使用のsvf.jarのBuild番号によって異なります。使用できるオプションは「-H」オプションでご確認ください。
- 起動オプションでのログ出力設定例

設定例 `jp.co.fit.vfreport.SvfrServer  
-L"C:¥temp¥SVF Connect for ASP/COM API¥server.log"`

設定時には、ファイル名まで指定する必要があります。

出力例 `Super Visual Formade for Report, Java Report Server  
Copyright(c) 2000, Version 1.14  
host address is [host名 / 10.17.100.212]  
bind address is [0.0.0.0:4141]  
free/total memory is [746,136/2,031,616] bytes  
[host名][Wed Apr 23 13:08:45 JST 2003] server start  
[127.0.0.1:1081][Wed Apr 23 13:08:52 JST 2003] connect  
[127.0.0.1:1081][Wed Apr 23 13:08:54 JST 2003] disconnect`

svf.jarのBuild番号「200504122140」以降を使用している場合、内部的な仕様変更により「出力例」にあるような文字列は表示されません。

起動されているかどうかの確認は、コマンドプロンプトから「netstat -an」コマンドを入力しておこないます。ポート「4141」がLISTENING状態になっていれば起動しています。

Build番号は、次のjavaコマンドを実行することで確認できます。

【Windows】

```
java -jar "%FIT_PRODUCTS_BASE%¥svfjpd¥lib¥svf.jar"
```

【Linux】

```
java -jar "$FIT_PRODUCTS_BASE/svfjpd/lib/svf.jar"
```

【出力例】

Product Information's

Product : SVF for Java Print

Copyright (C) 2000-2005 Flexible Integration Technology Corp.

Version : 7.0 (build 200512021748)

:

- SvfrServer起動時のjava.awt.headlessオプションの指定  
UNIX/Linuxコンピュータ上で画像ファイル（バーコード、QRコードを含む）を扱う際に（JDK1.4.2の使用が前提）、java.awt.headlessオプションを指定することで画像ファイルを出力することが可能となります。java.awt.headlessオプションの設定が必要となるのは、UNIX、またはLinux上でSvfrServerを起動している場合のみとなります。

【指定方法】

SVF JavaPRODUCTSがインストールされているサーバ上で起動しているSvfrServer（待受けモジュール）の起動オプションを、次のように変更します。

```
java -Xrs -Xms512m -Xmx512m jp.co.fit.vfreport.SvfrServer
```

↓

```
java -Xrs -Xms512m -Xmx512m -Djava.awt.headless=true jp.co.fit.vfreport.SvfrServer
```

## ● SVF 実行部 (.NET 版) 接続用ランタイム .....

SVF実行サーバで、SVF実行部 (.NET 版) を使用している場合は、SVF実行部 (.NET 版) 接続用ランタイムをセットアップします。セットアップは、つぎの手順でおこないます。

### 1

SVF 実行部 (.NET 版) がセットアップされているコンピュータで、SvfrServer.exe を起動します。

<ドライブ>:¥SVFdNP¥lib¥SvfrServer.exe (options)  
 \*SVF for .NET Framework のインストールディレクトリをデフォルトから変更している場合は、上記ディレクトリとは異なります。

サービス起動にも対応しています。詳細については次項「SVF実行部 (.NET 版) 接続用ランタイム起動オプション」を参照してください。



- SVF実行部 (SVF for .NET Framework) Ver1.0.1.0を使用している場合は、デフォルトで「SvfrServer.exe」はインストールされません。SVF SDK for .NET Framework (設計部 Ver6.4) のインストールで作成される「SvfrServer.exe」および「SvfrServer.properties」(デフォルトでは、<ドライブ>:¥SVFdNP¥lib¥に作成されます) を実行部側のlibフォルダ (デフォルトでは、<ドライブ>:¥SVFdNP¥lib¥) にコピーして起動してください。または、弊社サポート宛にご連絡頂ければ、モジュールを送付いたします。

## ● SVF 実行部 (.NET 版) 接続用ランタイム起動オプション .....

Usage: SvfrServer [options]

- P[n] リスポートアドレス[n]を指定します。  
指定されていない場合は「4141」となります。
- B[a] バインドアドレス[a]を指定します。  
指定されていない場合にはデフォルトとなります。
- W[n] クライアント側からのリクエスト受信待ち時間のタイムアウトの間隔[n] (min) を指定します。  
指定されていない場合にはデフォルトとなります。
- T[n] 使用できません。
- L[f] ログファイル名、コンソールへの出力、イベントログへの出力など [f] を指定します。  
指定されていない場合はログは出力されません。
- R サービスに登録します。
- U サービスを解除します。
- X 致命的なエラー (OutOfMemoryException) が発生した場合に例外内容をログ出力後に正常終了します。
- H ヘルプを表示します。  
この起動オプションの説明が表示されます



- 起動オプションでのログ出力設定例

設定例

[ファイルへの出力の場合]  
SvfrServer -L"C:¥log¥ server.log"

[実行したコマンドプロンプト画面への出力の場合]  
SvfrServer -L(STDOUT)

[イベントログへの出力の場合]  
SvfrServer -L(EVENTLOG)

出力例

```
SVF .NET Framework API Connect Server
Copyright(C) 2005, Version 7.0.0.0
host address is [host名/10.17.100.212]
bind address is [0.0.0.0:4141]
used memory is [456800] bytes

[host名][Wed Sep 27 15:43:36 JST 2006] server start
[127.0.0.1:1081][Wed Sep 27 15:57:52 JST 2006] connect
[127.0.0.1:1081][Wed Sep 27 15:57:55 JST 2006] disconnect
```

SvfrServer.exe(接続用ランタイム)のバージョンによって、出力内容が異なることがあります。

接続用ランタイムが起動されているかどうかの確認は、コマンドプロンプトから「netstat -an」コマンドを入力しておこなうことも可能です。ポート「4141」がLISTENING状態になっていれば起動しています。

- 接続用ランタイムがサービスとして起動されている場合、SVF実行部(.NET版)のプレビュー機能は使用できません。

# 3 サービスへの登録方法

SVF Connect for Java APIには、SVF JavaPRODUCTSやReport Director Enterpriseとの接続をおこなうためのプロセス「SvfrServer」をサービス起動するための機能が含まれています。

「SvfrServer」をサービス起動するためには、サービス登録をおこなう必要があります。サービス登録をするために、まず AppLancher.exe を使用して run ファイルを作成し、SvcLauncher.exe を使用してサービス登録をおこないます。



- Ver7以降では「SvfJavaConnectServer」となります。

## run ファイルの作成

まず AppLancher.exe を使用して、run ファイルを作成します。

AppLancher.exe と svcLauncher.exe は任意のフォルダに配置してください。作成した run ファイルも、同じフォルダに入れてください。

### ● AppLancher.exe の起動

AppLancher.exe をダブルクリックして起動させます。AppLancher.exe が起動すると、[Application Launcher] 画面が表示されます。



< [Application Launcher] 画面 >

[Application Launcher] 画面は、設定内容によって 4 つのタブに分かれています。

#### [アプリケーション] タブ

JDK への PATH の設定、svf.jar を起動するための設定をおこないます。

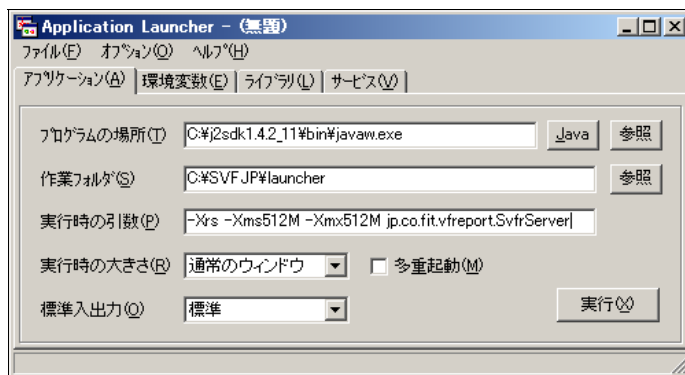
[ライブラリ] タブ CLASSPATH の設定をおこないます。

[サービス] タブ Windows に登録するサービス名を設定します。

[環境変数] タブ ここでは使用しません。

## ● アプリケーションの設定 .....

[アプリケーション] タブでは、JDK への PATH の設定、svf.jar を起動するための設定をおこないます。



<[アプリケーション] タブ>

### プログラムの場所

Windows のシステムに登録されている JDK への PATH(javaw.exe のインストールフォルダ)を設定します。

[Java] をクリックすると、使用している Windows に登録されている javaw.exe のインストールフォルダが一覧表示されます。表示された一覧から javaw.exe のインストールディレクトリを指定します。または、[参照] ボタンをクリックしてフォルダを選択します。

**作業フォルダ** svf.jar のインストールフォルダを指定します。[参照] ボタンをクリックしてフォルダを選択します。

**実行時の引数** svf.jar 起動時のコマンドを指定します。

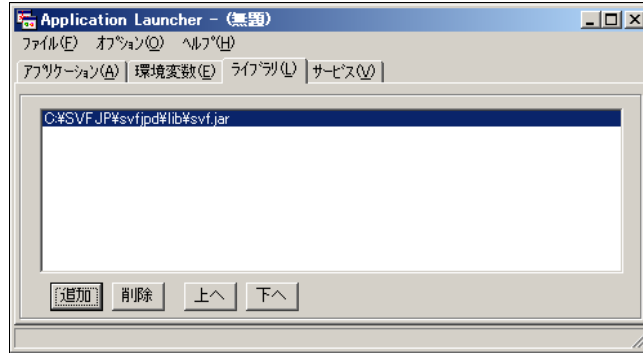
その他の項目 ([実行時の大きさ]、[多重起動]、[標準入出力])については、デフォルト値のままで問題ありません。



## ● ライブラリの設定 .....

[ライブラリ]タブでは、Windowsのシステムに登録されているsvf.jarのCLASSPATHの設定をおこないます。

[追加]ボタンをクリックして、svf.jarを選択します。



<[ライブラリ]タブ>



- Ver7 では「svf.jar」、「svfpdfparser.jar」、「svfgrph.jar」、「xercesImpl.jar」、「xml-apis.jar」、「jai\_codec.jar」、「jai\_core.jar」をCLASSPATHに通してください。  
Ver8 では「svf.jar」、「svfpdfparser.jar」、「svfgrph.jar」、「jai\_codec.jar」、「jai\_core.jar」、「forme.jar」をCLASSPATHに通してください。  
詳細については、弊社サポート窓口までお問い合わせください。

## ● サービスの設定 .....

[サービス]タブでは、Windowsに登録するサービス名を設定します。



<[サービス]タブ>

**表示名** Windows に登録するサービス名を記述します。

**再起動秒時間** 何かしらの原因でサービスが落ちた場合に、サービスを自動的に再起動するまでの時間を設定します。

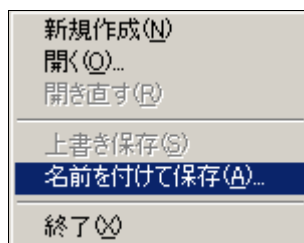
「0」を設定すると自動再起動はおこないません。

## ● 設定の保存 .....

各種設定が終了したら、設定内容をファイルとして保存します。

### 1

[ファイル]メニューから[名前を付けて保存]を選択して保存操作をおこないます。



設定内容がファイルとして保存されます。ファイル名の拡張子は、「run」となります。ここでは、SvfrServer.runというファイル名で保存します。



- Ver.7以降ではサービス名は「Svf.JavaConnectServer」となります。

## 動作確認

runファイルを作成後、設定した内容でsvf.jarが正常に起動することを確認します。

## ● svf.jar の起動 .....

svf.jarの起動は、次の手順でおこないます。

### 1

コマンドプロンプト、または、[スタート]-[ファイル名を指定して実行]で次の内容を入力します。

AppLauncher.exeのフルパス -START runファイル名のフルパス



起動に成功すると、WindowsのタスクマネージャにAppLauncher.exeが表示されます。

## ● svf.jar の停止 .....

svf.jarの停止は、次の手順でおこないます。

### 1

コマンドプロンプト、または、[スタート]–[ファイル名を指定して実行]で次の内容を入力します。

```
AppLauncher.exeのフルパス -STOP runファイル名のフルパス
```



## サービスへの登録

作成したrunファイルを使用して、モジュール「svf.jar」をサービス登録する方法について説明します。

### 1

コマンドプロンプト、または[スタート]–[ファイル名を指定して実行]で次の内容を入力します。

```
svcLauncher.exeのフルパス -INSTALL runファイル名のフルパス
```



正常にサービス登録がおこなわれると、Windowsのサービス一覧に、runファイルで指定しているサービス名が表示されます。

## サービス登録の解除

サービス登録を解除したい場合は、次の手順で操作をおこないます。

### 1

コマンドプロンプト、または[スタート]–[ファイル名を指定して実行]で次の内容を入力します。

```
svcLauncher.exeのフルパス -UNSTALL runファイル名のフルパス
```



サービス登録が解除されます。

# 4 サンプルプログラム

ここでは、Javaアプリケーションプログラムを使用するサンプルプログラムを記載します。

## クエリー/レポートライタータイプ「Sample.java」

```

/*
 * Sample.java
 *
 * 本プログラムコードはサンプルとして提供されるものです。
 * あくまでサンプルとしての理解しやすさを主眼としているため、エラーハンドリ
 * ング、動作に関する効率、メンテナンス性、セキュリティ、同時実行性、負荷への
 * 耐久性等について、何ら保証はいたしかねます。ご使用になられるお客様の責任
 * においてご使用ください。
 * また、このコードをお客様が使用されたことにより損害が発生した場合にも、弊
 * 社は当該損害に関する一切の責任を負いかねます。お客様の方で十分内容を吟
 * 味いただき、上記の点をご承諾のうえご使用ください。
 *
 * Copyright (C) 2006 WingArc Technologies Co., Ltd.
 */
// SvfrClient クラスを使うために必要な宣言
import import.jp.co.fit.vfreport.SvfrClient;

//
// 以降は、SvfrClient クラスの使い方を示すためのサンプルプログラムです。
//

public class Sample
{
    private static final String[][] sampleDats = {
        { "EMPNO", "ENAME", "JOB", "MGR", "HIREDATE", "SAL", "COMM",
        "DEPTNO", "DNAME", "LOC" },
        { "7369", "SMITH", "CLERK", "7902", "80-12-17", "800", "", "20",
        "RESEARCH", "DALLAS" },
        { "7499", "ALLEN", "SALESMAN", "7698", "81-02-20", "1600", "300",
        "30", "SALES", "CHICAGO" },
        { "7521", "WARD", "SALESMAN", "7698", "81-02-22", "1250", "500",
        "30", "SALES", "CHICAGO" },
        { "7566", "JONES", "MANAGER", "7839", "81-04-02", "2975", "",
        "20", "RESEARCH", "DALLAS" },
        { "7654", "MARTIN", "SALESMAN", "7698", "81-09-28", "1250", "1400",
        "30", "SALES", "CHICAGO" },
        { "7698", "BLAKE", "MANAGER", "7839", "81-05-01", "2850", "",
        "30", "SALES", "CHICAGO" },
    };
}

```

```
        { "7782", "CLARK", "MANAGER" , "7839", "81-06-09", "2450", "" ,
"10" , "ACCOUNTING", "NEW YORK" },
        { "7788", "SCOTT", "ANALYST" , "7566", "87-04-19", "3000", "" , "20" ,
"RESEARCH" , "DALLAS" },
        { "7839", "KING" , "PRESIDENT", "" , "81-11-17", "5000", "" , "10" ,
"ACCOUNTING", "NEW YORK" },
        { "7844", "TURNER", "SALESMAN" , "7698", "81-09-08", "1500", "0" ,
"30" , "SALES" , "CHICAGO" },
        { "7876", "ADAMS", "CLERK" , "7788", "87-05-23", "1100", "" , "20" ,
"RESEARCH" , "DALLAS" },
        { "7900", "JAMES", "CLERK" , "7698", "81-12-03", "950", "" , "30" ,
"SALES" , "CHICAGO" },

        { "7902", "FORD" , "ANALYST" , "7566", "81-12-03", "3000", "" , "20" ,
"RESEARCH" , "DALLAS" },
        { "7934", "MILLER", "CLERK" , "7782", "82-01-23", "1300", "" , "10" ,
"ACCOUNTING", "NEW YORK" }
    };

    //
    // 以降は、SvfrClient クラスの使い方を示すためのサンプルプログラムです。
    //

    public static void main(String[] args)
    {
        String dbServer = args.length > 0 ? args[0] : "";
        String hostName = args.length > 1 ? args[1] : "localhost";
        new Sample(dbServer, hostName);
    }

    //
    // PDF 作成サンプルプログラム
    //

    public Sample(String dbServer, String hostName)
    {
        String dbq = dbServer.length() > 0 ? "dbq=" + dbServer + ";" : "";
        String form = "." + java.io.File.separator + "sample.frm";
        String pdf = "." + java.io.File.separator + "sample.pdf";
        String query = "." + java.io.File.separator + "sample.vrq";
        String conn = dbq + "uid=scott;pwd=tiger";
        int mode, code;
        boolean ok = false;

        try {

            //
            // SvfrClient と SVF Connect for Java API サーバとの接続開始
            // hostName 部分に PDF 作成サーバの IP アドレスを記述
            //
```

```
SvfrClient svf = new SvfrClient(hostName);
try {

    //
    // PDF 作成開始
    //

    code = svf.VrInit();
    if (code < 0)
        System.out.println("error: VrInit() = " + code);
    else {
        try {
            //
            // PDF 出力を明示的に指示
            //

            code = svf.VrSetPrinter("", "PDF");
            if (code < 0)
                System.out.println("error: VrSetPrinter() = " + code);

            //
            // 作成する PDF 名の指定
            //

            code = svf.VrSetSpoolFileName2(pdf);
            if (code < 0)
                System.out.println("error: VrSetSpoolFileName2(¥" + pdf + "¥) =
" + code);

            //
            // フォームファイルの指定
            //

            mode = 5;
            code = svf.VrSetForm(form, mode);
            if (code < 0)
                System.out.println("error: VrSetForm(¥" + form + "¥", " + mode +
") = " + code);
            else {

                //
                // レコードデータの出力
                //

                if (dbServer.equals("")) {

                    code = -1;
```

```
recordLoop:

for (int row = 1; row < sampleDatas.length; ++row) {
    for (int col = 0; col < sampleDatas[0].length; ++col)
        svf.VrsOut(sampleDatas[0][col], sampleDatas[row][col]);

    //
    // 1レコード分のデータの出力終了命令
    //

    code = svf.VrEndRecord();
    if (code < 0) {
        System.out.println("error: VrEndRecord() = " + code);
        break recordLoop;

    }
}
if (code == 0)
    ok = true;
}
else {

    //
    // クエリーファイル使用時のファイル名指定
    //

    mode = 0;
    code = svf.VrSetQuery(conn, query, mode);
    if (code < 0)
        System.out.println("error: VrSetQuery(¥" + conn + "¥", ¥" +
query + "¥", " + mode + ") = " + code);
    else {

        //
        // クエリー実行開始
        //

        code = svf.VrExecQuery(); // API 関数マニュアルに記載されている
VrReport と同じ関数ですが、使用する名前が違います。
        if (code < 0)
            System.out.println("error: VrExecQuery() = " + code);
        else
            ok = true;

    }
}
}
}
catch (Exception e) {
    // io error
```



```
        e.printStackTrace();
    }

    finally {

        //
        // ※ VrInit() 成功後は、必ず呼び出してください。
        //

        code = svf.VrQuit();
        if (code < 0)
            System.out.println("error: VrQuit() = " + code);
        else if (ok)
            System.out.println("success: output to ¥" + pdf + "¥");
    }
}
}
catch (Exception e) {
    // VrInit() error ?
    e.printStackTrace();
}

finally {

    //
    // ※ new SvfrClient() 成功後は、必ず呼び出してください。
    //
    svf.close();
}
}
catch (Exception e) {
    // connection failed
    e.printStackTrace();
}
}
}
```

## 固定帳票タイプ「CHAIN.java」

```
/*
 * CHAIN.java
 *
 * 本プログラムコードはサンプルとして提供されるものです。
 * あくまでサンプルとしての理解しやすさを主眼としているため、エラーハンドリ
 * ング、動作に関する効率、メンテナンス性、セキュリティ、同時実行性、負荷への
 * 耐久性等について、何ら保証はいたしかねます。ご使用になられるお客様の責任
 * においてご使用ください。
 * また、このコードをお客様が使用されたことにより損害が発生した場合にも、弊
 * 社は当該損害に関する一切の責任を負いかねます。お客様の方で十分内容を吟
 * 味いただき、上記の点をご承諾のうえご使用ください。
 *
 * Copyright (C) 2006 WingArc Technologies Co., Ltd.
 */
//SvfrClient クラスを使うために必要な宣言
import.jp.co.fit.vfreport.SvfrClient;

//
// 以降は、SvfrClient クラスの使い方を示すためのサンプルプログラムです。
//

public class CHAIN {

    public static void main(String[] args) {
        String hostName = args.length > 1 ? args[1] : "localhost";
        String dbServer = args.length > 0 ? args[0] : "";
        new CHAIN(dbServer, hostName);
    }

//
//PDF 作成サンプルプログラム
//

    public CHAIN(String dbServer, String hostName){

        int code;
        boolean ok = false;

        try {

            //
            //SvfrClient と SVF Connect for Java API サーバとの接続開始
            //"localhost"部分に PDF 作成サーバの IP アドレスを記述
            //

            SvfrClient svf = new SvfrClient("localhost");

            try {

                //
                //PDF 作成開始
                //
```

```
code = svf.VrInit();
if (code < 0)
System.out.println("error: VrInit() = " + code);
else {

try {

    //
    //PDF 出力を明示的に指示
    //

code = svf.VrSetPrinter("", "PDF");
if (code < 0)
    System.out.println("error: VrSetPrinter() = " + code);

    //
    //フォームファイルの指定
    //

code = svf.VrSetForm("CHAIN.frm", 1);
if (code < 0)
    System.out.println("error: VrSetForm() = " + code);
else {

    //
    //作成する PDF ファイル名の指定
    //

code = svf.VrSetSpoolFileName2("chain.pdf");
if (code < 0)
    System.out.println("error: VrSetSpoolFileName2() = " + code);

    //
    //以降データを出力
    //

code = svf.VrsOut("SHAMEI", "ウイングアーク");
if (code < 0)
    System.out.println("error: VrsOut(SHAMEI) = " + code);

code = svf.VrsOut("TENMEI", "ツール営業部");
if (code < 0)
    System.out.println("error: VrsOut(TENMEI) = " + code);

code = svf.VrsOut("SHATEN_CODE", "010203040506");
if (code < 0)
    System.out.println("error: VrsOut(SHATEN_CODE) = " + code);

code = svf.VrsOut("BUNRUI", "0001");
if (code < 0)
    System.out.println("error: VrsOut(BUNRUI) = " + code);

code = svf.VrsOut("DENPYO_KUBUN", "11");
if (code < 0)
    System.out.println("error: VrsOut(DENPYO_KUBUN) = " + code);
```

```
code = svf.VrsOut("NO", "0000001");
if (code < 0)
    System.out.println("error: VrsOut(NO) = " + code);

code = svf.VrsOut("TORIHIKISAKI_CODE", "110001");
if (code < 0)
    System.out.println("error: VrsOut(TORIHIKISAKI_CODE) = " +
code);

code = svf.VrsOut("TORIHIKISAKI", "FIT");
if (code < 0)
    System.out.println("error: VrsOut(TORIHIKISAKI) = " + code);

code = svf.VrsOut("TORIHIKISAKI_TEL", "03-5766-2833");
if (code < 0)
    System.out.println("error: VrsOut(TORIHIKISAKI_TEL) = " + code);

code = svf.VrsOutn("KIKAKU", 1, "SVF 設計部");
if (code < 0)
    System.out.println("error: VrsOut(KIKAKU,1) = " + code);

//
//繰り返し項目へのデータ出力
//

code = svf.VrsOutn("SHOHIN_CODE", 1, "0000000000001");
if (code < 0)
    System.out.println("error: VrsOut(SHOHIN_CODE,1) = " + code);

code = svf.VrsOutn("SURYO", 1, "5");
if (code < 0)
    System.out.println("error: VrsOut(SURYO,1) = " + code);

code = svf.VrsOutn("TANKA", 1, "700000");
if (code < 0)
    System.out.println("error: VrsOut(TANKA,1) = " + code);

code = svf.VrsOutn("KIKAKU", 2, "SVF/Web 実行部");
if (code < 0)
    System.out.println("error: VrsOut(KIKAKU,2) = " + code);

code = svf.VrsOutn("SHOHIN_CODE", 2, "0000000000002");
if (code < 0)
    System.out.println("error: VrsOut(SHOHIN_CODE,2) = " + code);

code = svf.VrsOutn("SURYO", 2, "3");
if (code < 0)
    System.out.println("error: VrsOut(SURYO,1) = " + code);

code = svf.VrsOutn("TANKA", 2, "800000");
if (code < 0)
    System.out.println("error: VrsOut(TANKA,1) = " + code);

code = svf.VrsOutn("KIKAKU", 3, "RD サーバ版");
```

```
if (code < 0)
    System.out.println("error: VrsOut(KIKAKU,3) = " + code);

code = svf.VrsOutn("SHOHIN_CODE", 3, "000000001001");
if (code < 0)
    System.out.println("error: VrsOut(SHOHIN_CODE,3) = " + code);

code = svf.VrsOutn("SURYO", 3, "4");
if (code < 0)
    System.out.println("error: VrsOut(SURYO,3) = " + code);

code = svf.VrsOutn("TANKA", 3, "500000");
if (code < 0)
    System.out.println("error: VrsOut(TANKA,1) = " + code);

code = svf.VrsOutn("KIKAKU", 4, "RD ユーティリティ版");
if (code < 0)
    System.out.println("error: VrsOut(KIKAKU,4) = " + code);

code = svf.VrsOutn("SHOHIN_CODE", 4, "000000001003");
if (code < 0)
    System.out.println("error: VrsOut(SHOHIN_CODE,4) = " + code);

code = svf.VrsOutn("SURYO", 4, "60");
if (code < 0)
    System.out.println("error: VrsOut(SURYO,4) = " + code);

code = svf.VrsOutn("TANKA", 4, "50000");
if (code < 0)
    System.out.println("error: VrsOut(TANKA,4) = " + code);

//
//データ出力終了(改ページ)
//

code = svf.VrEndPage();
if (code < 0)
    System.out.println("error: VrEndPage() = " + code);
else
    ok = true;
}
}

catch (Exception e) {
    // io error
    e.printStackTrace();
}

}

//
// ※ VrInit() 成功後は、必ず呼び出してください。
//
code = svf.VrQuit();
```

```
        if (code < 0)
            System.out.println("error: VrQuit() = " + code);

    }

    catch (Exception e) {
        // VrInit() error ?
        e.printStackTrace();
    }

    finally {
        //
        // ※ new SvfrClient() 成功後は、必ず呼び出してください。
        //
        svf.close();
    }

}

catch (Exception e) {
    // connection failed
    e.printStackTrace();
}

}

}
```

# 5 注意事項

SVF Connect for Java APIを使用する際の注意事項について記述します。

- VrInit()関数は 1 個のオブジェクトに対して 1 回のみ使用可能となります。  
同時に複数個実行する場合は、別オブジェクトとして生成をおこなってください。
- VrSetForm()関数等のファイル名を指定する時のパラメータは各プラットフォームに依存します。  
プラットフォームを変更する際は、以前使用していたプラットフォームと同様のフォルダ構成にするか、アプリケーションで指定しているファイル名のパス等を変更する必要があります。
- svfrclient.jar を他の jar に取り込んで使用することは誤動作の原因となるため、おこなわないでください。
- 処理終了後は、close()をかならず発行するようにしてください。





● 索引

**A**

---

Application Launcher.....11

**C**

---

CHAIN.java.....22

**J**

---

Javaプログラム用クラスライブラリ .....2, 4  
Javaプログラム用クラスライブラリ側オプション ....4  
Java言語用メソッドリスト .....2

**L**

---

SVF実行部 (.NET版) 接続用ランタイム.....9  
SVF実行部 (.NET版) 接続用ランタイム起動オプション...9

**S**

---

Sample.frm .....2  
Sample.java .....2, 17  
Sample.vrq.....2  
sample/sample.frm .....2  
sample/Sample.java.....2  
sample/sample.vrq.....2  
SVF 実行部 (C版) 用ランタイム起動オプション ....6  
svf.jar .....7, 14, 15  
svfrclient.jar .....1, 2, 4  
svfrclient.properties .....2, 4  
SvfrServer.exe.....9  
svfrsvr.exe .....5  
SVF実行部 (C版) 接続用ランタイム .....2, 5  
SVF実行部 (Java版) 接続用ランタイム.....7  
SVF実行部 (Java版) 用ランタイム起動オプション...7

**か**

---

ガベージコレクション .....7

**く**

---

クエリー.....17

**こ**

---

固定帳票.....22

**さ**

---

サービス登録.....11  
サービス.....6  
サンプル.....2  
サンプルプログラム.....17

**せ**

---

制限事項.....27  
製品概要.....1  
セットアップ.....4

**た**

---

タイムアウト.....6, 7, 9

**ち**

---

注意事項.....27

**と**

---

動作環境.....3

**は**

---

バインドアドレス .....6, 7, 9

**ふ**

ファイル構成 .....2

**へ**

ヘルプ .....6, 7

**ら**

ライブラリ .....13

**り**

リソースポートアドレス ..... 6, 7, 9

**れ**

レポートライター ..... 17

**ろ**

ログ ..... 9

ログ出力..... 6, 8, 10

ログファイル..... 6

ログファイル名 ..... 7



2008年 1月 第1版

**販売元：ウイングアーク テクノロジーズ株式会社**

[本社] 〒108-0075 東京都港区港南 1-8-15 Wビル 14F

TEL:03-6710-1700(代表) FAX:03-6710-1701

ホームページ:<http://www.wingarc.co.jp>

**開発元：株式会社エフ・アイ・ティ**

〒102-0071 東京都千代田区富士見 2-14-37 FUJIMI EAST 2F

---

※本マニュアルに記載されている社名および商品名は、一般に各社の商標および登録商標です。

※本マニュアルは予告なく変更することがあります。