

## 記事

[Toshihiko Minamoto](#) · 2021年2月22日 4m read

## IRIS Interoperability機能を使ったファイル連携

皆さん、こんにちは。

他のシステムとファイル連携を行う場合、Cachéでは、Jobコマンドを使った常駐プロセスやタスクを作成し、特定のディレクトリにあるファイルを定期的に監視、データを取り込むといった機能を手作りされていたかと思います。

これにはファイルの監視や、常駐プロセスの監視、プロセスの制御（起動、停止）を行う機能を用意する必要がありましたが、IRISではInteroperability機能が使えますので、そのようなプログラムを省略することが可能です。

今回は、

**既にファイ**

**ル読み込み処理を行うルー**

**チンが存在し、ルーチン呼出時の引数にファ**

**イル名があるという前提**

で、その処理をInteroperability機能を使ってどのように呼び出すかについて説明したいと思います。

### 手順

作業手順は以下の通りとなります。

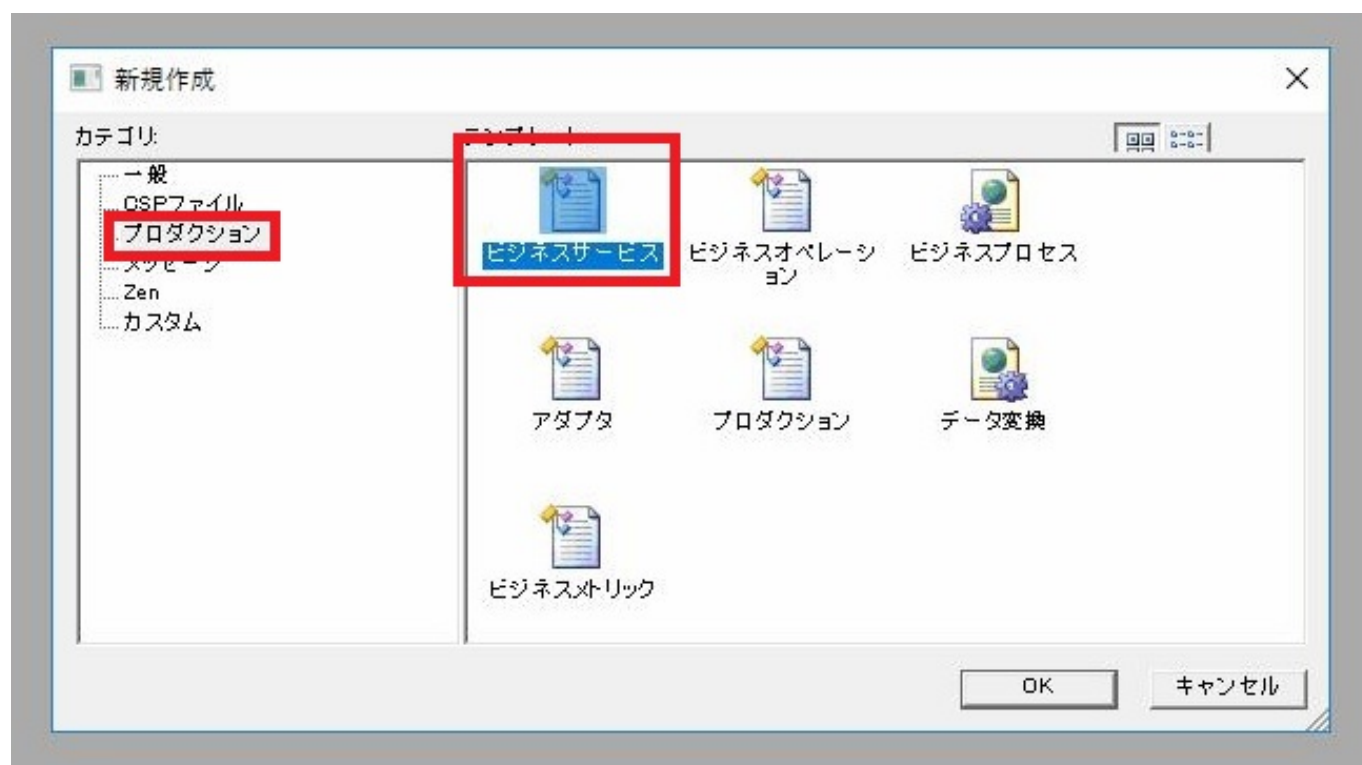
1. ビジネスサービスクラスの作成
2. OnProcessInput()メソッドの実装
3. プロダクションの作成
4. ビジネスサービスの登録
5. プロダクションの起動

### ビジネスサービスクラスの作成

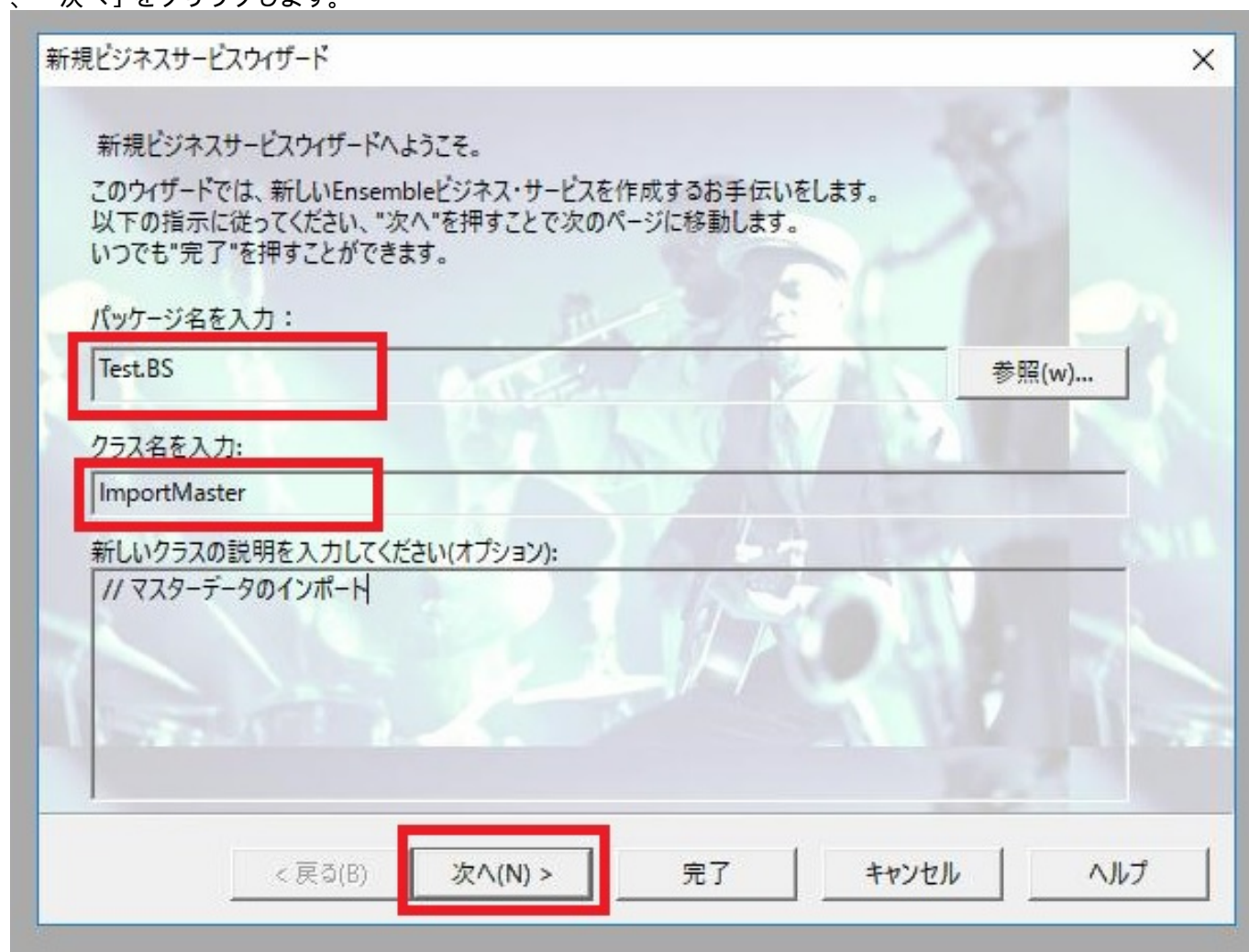
スタジオを起動し、ファイル読み込み処理を行うルーチンが存在するネームスペースに接続します。

「ファイル」「新規作成」メニューをクリックします。

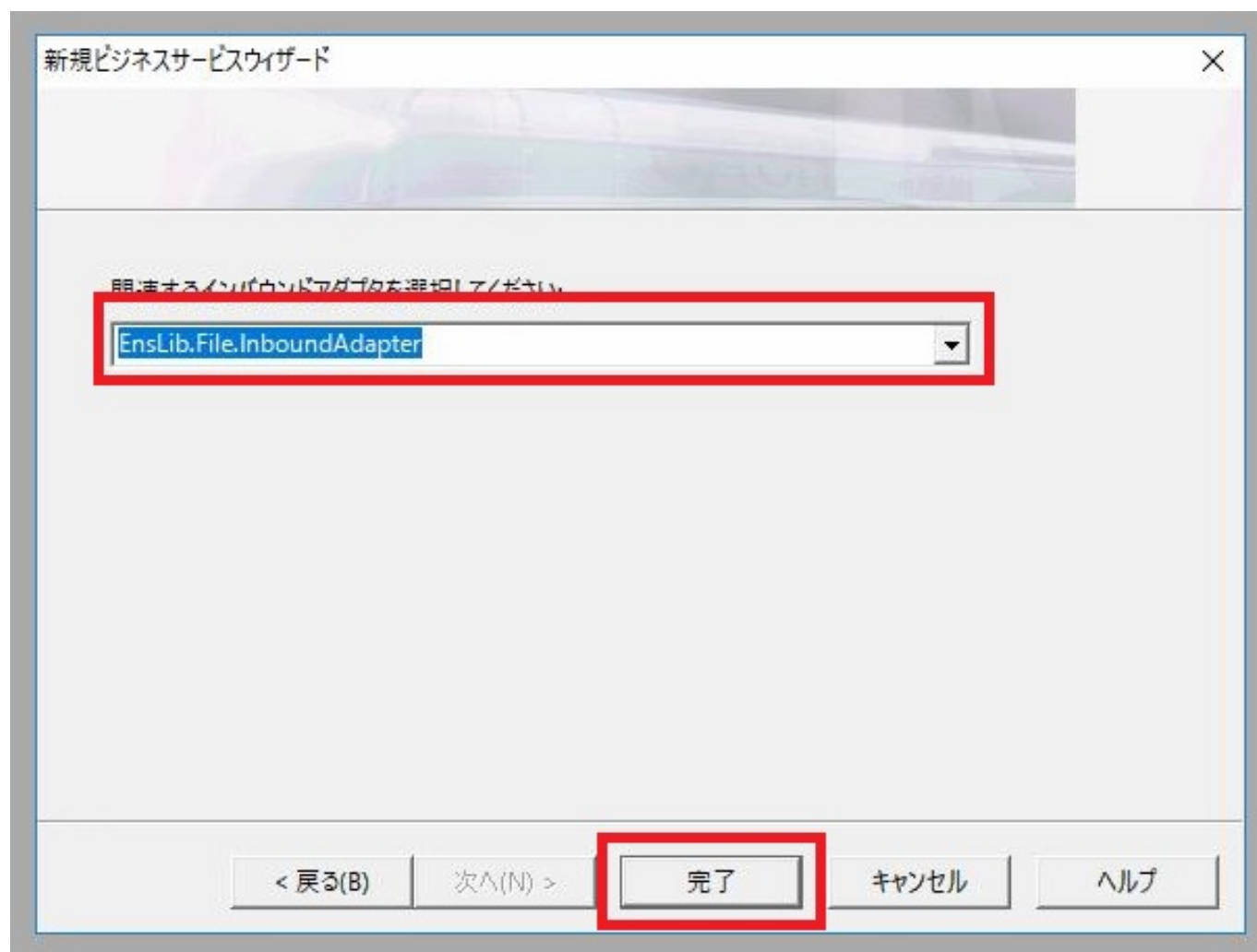
以下のダイアログボックスが表示されますので、「プロダクション」カテゴリをクリックし「ビジネスサービス」アイコンをクリック、「OK」ボタンをクリックします。



以下のウィザードが表示されますので、作成するビジネスサービスのクラス名（パッケージ名、クラス名）を入力、「次へ」をクリックします。



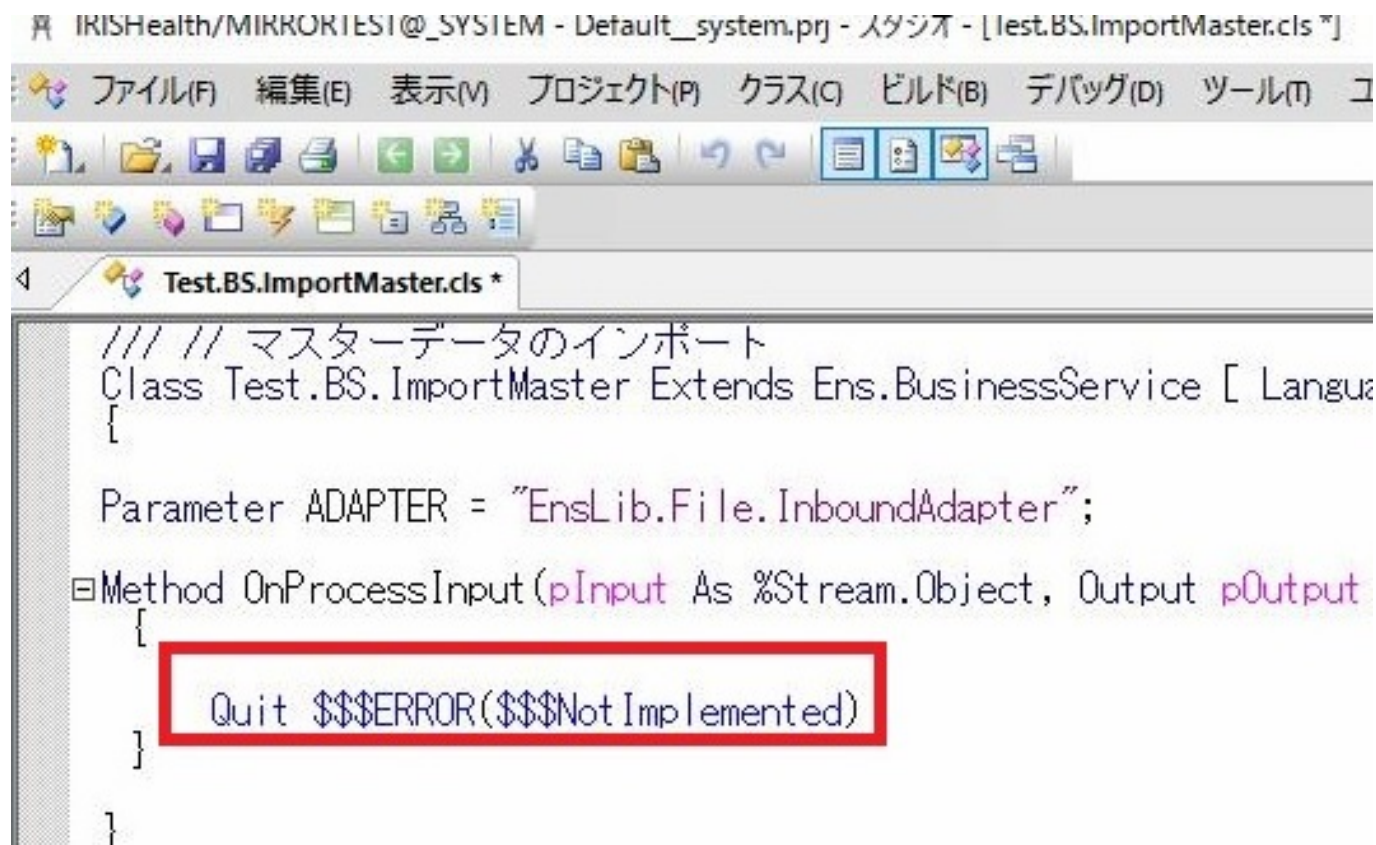
以下のようにアダプターを選択する画面が表示されますので、「EnsLib.File.InboundAdapter」を選択し、「完了」ボタンをクリックします。



以上でビジネスサービスクラスは作成できました。

### OnProcessInput()メソッドの実装

以下のようなテンプレートが作成されますので、OnProcessInput ( ) メソッド部分(赤枠部分)を修正します。



今回は、以下のようにファイルのストリームを表す変数pInputの属性にあるファイル名を取得し、引数として読込ルーチン ImportData^Productsを呼び出しています。

```

/// ファイルを検知した際の処理
Method OnProcessInput(pInput As %Stream.Object, Output pOutput As %RegisteredObject)
{
    // 検知したファイルのファイル名を取得
    set filename=pInput.Attributes("Filename")

    // 商品情報を読み込むルーチンを呼び出す
    do ImportData^Products(filename)
    Quit $$$OK
}
}

```


修正したクラスをコンパイルします。

## プロダクションの作成

プロダクションはInteroperability機能を実行するのに必要な設定で、ネームスペースごとに1つだけ実行することが可能です。

プロダクションを作成するにはシステム管理ポータルを起動し、「Interoperability」をクリックすると以下のようなネームスペース一覧が表示されますので、先ほど作成したクラスがあるネームスペースをクリックします。









管理ポータル


ホーム


サーバ SAM-DB01    ネームスペース %SYS [変更](#)    ユーザ [SYSTEM](#)    ライセンス先 S


ようこそ, **\_SYSTEM** 表示:  

 **ホーム**

 **Health**

 **Analytics**

 **Interoperability**

 **システムオペレーション**

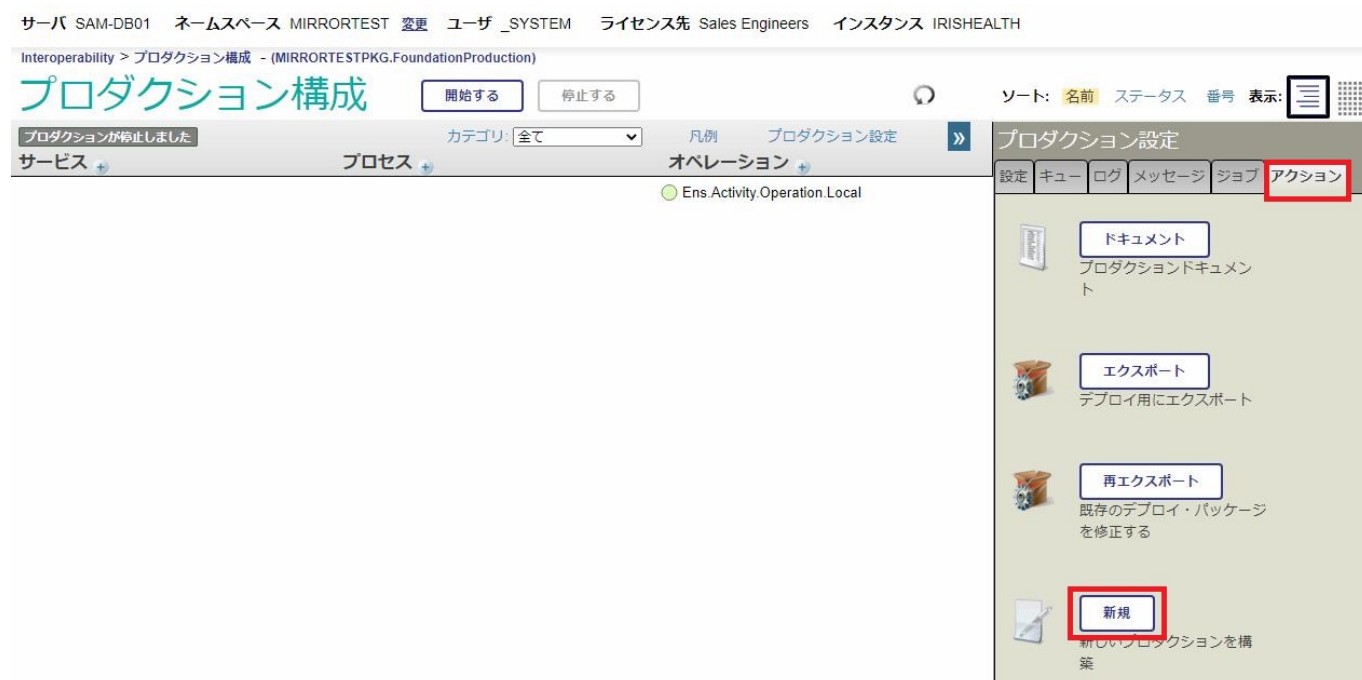
%SYS ネームスペースはフ

別のネームスペースを選択してください

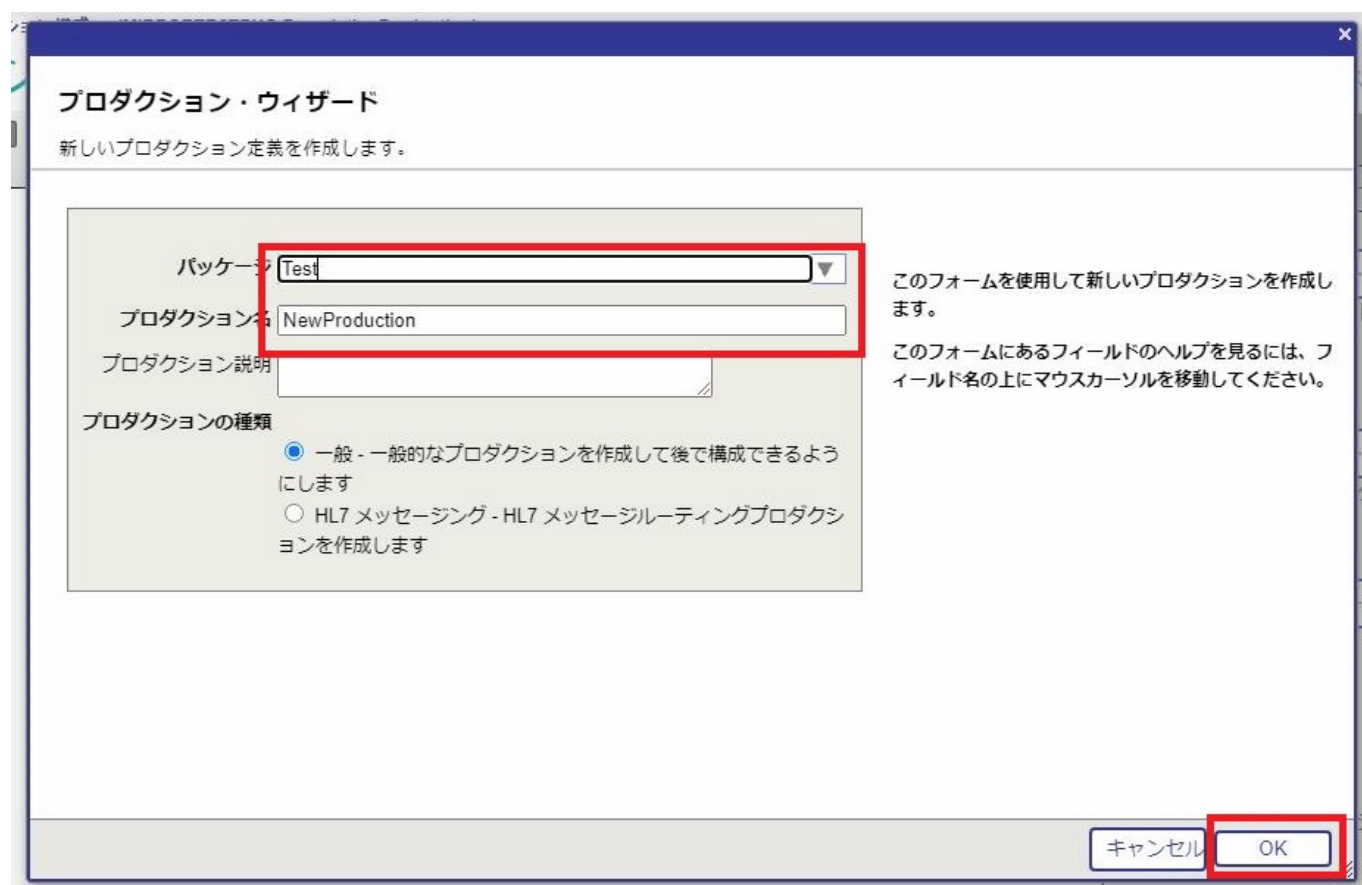
プロダクションに使用可能なネームスペース

HSSYS	
<b>MIRRORTTEST</b>	
TEST	

さらに「構成」メニューの「プロダクション」を選択すると、以下のようなプロダクション構成画面が表示されますので、画面右上の「アクション」タブをクリックし、その下にある「新規」ボタンをクリックします。



以下のダイアログが表示されますので、作成するプロダクションのクラス名を入力し、「OK」ボタンをクリックします。



これでプロダクションが作成できました。

## ビジネスサービスの登録

プロダクションが作成できましたので、先ほど作成したビジネスサービスクラスを登録します。まず、以下のように「サービス」の右にある「+」ボタンをクリックします。

サーバ SAM-DB01    ネームスペース MIRRORTEST    変更    ユーザ \_SYSTEM    ライセン:

Interoperability > プロダクション構成 - (Test.NewProduction)

## プロダクション構成

開始する    停止する

プロダクションが停止しました    カテゴリ: 全て ▼

サービス +    プロセス +

以下の画面が表示されますので、サービスクラス欄に先ほど作成したビジネスサービスクラス名（Test.BS.Import Master）を選択し、「有効にする」をチェック、「OK」ボタンをクリックします。

ビジネス・サービス・ウィザード

このプロダクションに新しいビジネス・サービスを追加します。

すべてのサービス    HL7入力    X12 入力    ビジネス・メトリック

サービスクラス Test.BS.ImportMaster ▼

サービス名

カテゴリを表示 ▼

コメント

有効にする ☒

このフォームを使用して、新しいビジネス・サービスをプロダクションに追加します。

このフォーム内では、設定名の上にカーソルを置くことで設定のヘルプを表示します。

キャンセル    OK

これでビジネスサービスが登録されましたので、プロダクション構成画面にて、作成されたビジネスサービスをクリックし、右側の「設定」タブをクリックします。



設定には「基本の設定」の中に「ファイルパス」と「アーカイブパス」がありますので、以下のように「ファイルパス」には他のシステムにて書き込まれるファイルのディレクトリ、「アーカイブパス」はIRISが読み込んだファイルを保管するディレクトリを指定し、「適用」ボタンをクリックします。ここで指定したディレクトリは必ず作成しておいてください。



## プロダクションの起動

設定が完了しましたので、プロダクションを起動し、ファイルを監視する常駐プロセスを起動します。プロダクション構成画面の上にある「開始する」ボタンをクリックします。





以上で完了です！

ファイルパスで指定されたディレクトリ(今回の例ではc:\import\products\in)に他システムからファイルが出力されると、ImportData^Productsが起動され、引数にファイル名が渡されるようになります。

ぜひ、お試しください 😊

ご意見、ご質問等ございましたら、お気軽に返信いただけるとありがたいです。  
よろしくお願いいたします。

[#ObjectScript](#) [#相互運用性](#) [#InterSystems IRIS](#)

ソースURL:<https://jp.community.intersystems.com/post/iris-interoperability%E6%A9%9F%E8%83%BD%E3%82%92%E4%BD%BF%E3%81%A3%E3%81%9F%E3%83%95%E3%82%A1%E3%82%A4%E3%83%AB%E9%80%A3%E6%90%BA>