


```
        // ??????Credentials????????????????????????????????????????
        Set tSC=$$$EnsError($$$EnsErrNoCredentials,..Credentials) Quit tSC
    }
    // ?????????????????????????????????????????????????????????????
    set tSC=..ExecNetUse($piece(..Adapter.FilePath,"\",1,4),
tCredential.Username,tCredential.Password)
    if $$$ISERR(tSC) {
        $$$LOGSTATUS(tSC) // ?????????????????????????????????????????
        set tSC2=..ExecNetUse($piece(..Adapter.FilePath,"\",1,4),,,1)
        if $$$ISERR(tSC2) $$$LOGSTATUS(tSC2)
        set %objlasterror=tSC
        quit tSC
    }
    $$$LOGINFO("?????????? "_$piece(..Adapter.FilePath,"\",1,4)" ??????????")
}
Quit $$$OK
}
```

ExecNetUseメソッドは以下の通りです。ここではエラー出力をファイルに出力し、戻り値が0でない場合、ファイルから出力メッセージを読み込み、エラーメッセージとして返しています。

```
ClassMethod ExecNetUse(sharefolder As %String, username As %String = "",
password As %String = "", deleteflg As %Boolean = 0) As %Status
{
    // ?????????????????????????????????????????????????????????????
    set curdir=$SYSTEM.Process.CurrentDirectory()
    set errlog=curdir_"netuse_"_$_job_.log" set tSC=$$$OK
    try {
        if deleteflg {
            // Net use????????????????????????????????????????????
            set sts=$zf(-1,"net use "_sharefolder_" /delete 2> ""_errlog_"")
        } else {
            // Net use????????????????????????????????????????????
            set sts=$zf(-1,"net use "_sharefolder_" "_password_$select(
username'="" ":" /user:"_username,1:"")
                _" /persistent:NO 2> ""_errlog_"")
        }
        quit:sts=0 // ?????????????????????????????????????????
        set stream=##class(%FileCharacterStream).%New()
        set stream.Filename=errlog
        set msg=$translate(stream.Read(),$char(10,13)," ") // ??????????
        kill stream
        set tSC=$$$ERROR(5001,"net use ?????????????????????????????? ???"_sts_" Msg="_msg")
    } catch {
        set tSC=$$$ERROR(5001,"net use ?????????????????????????????? Err="_$zerror)
    }
    // ?????????????????????????????????????????????????????????????
    do ##class(%File).Delete(errlog)
    quit tSC
}
```

OnTearDown()メソッドの作成

OnTearDown()メソッドはビジネスサービス停止時に呼ばれます。ファイルパスがWindows共有フォルダ(ファイルパスが//で始まる)の場合、ExecNetUseメソッドを呼び出し、Net useコマンドで共有ディレクトリの切断を行い、継承元の処理を行います。

```
Method OnTearDown() As %Status  
{  
    // ??????????  
    set tSC=..ExecNetUse($piece(..Adapter.FilePath,"\",1,4),,,1)  
    if $$$ISERR(tSC) $$$LOGSTATUS(tSC)  
    if $$$ISOK(tSC) $$$LOGINFO("????????? "_$piece(..Adapter.FilePath,"\",1,4)_  
" ?????????")  
    // ??????????  
    quit ##super()  
}
```

クラスのコンパイル

プロパティやメソッドを追加した Test.BS.ImportMaster クラスをコンパイルします。

認証情報の設定

システム管理ポータル内の「Interoperability」「構成」「認証情報」メニューをクリックし、認証情報を設定します。

以下の画面が表示されますので、右下の「新規」ボタンをクリックし、「ID」「ユーザ名」「パスワード」を入力し、「保存」ボタンをクリックします。



余談になりますが、当初「ユーザ名」欄をリモートサーバのユーザ名のみにしてしまうとnet useコマンドが時々失敗し、接続できなくなってハマってしまいました 😞

「ユーザ名」欄は<NetBIOSコンピュータ名>/<ユーザ名> または <ドメイン名>/<ユーザ名> にしたほうがよいと思います。

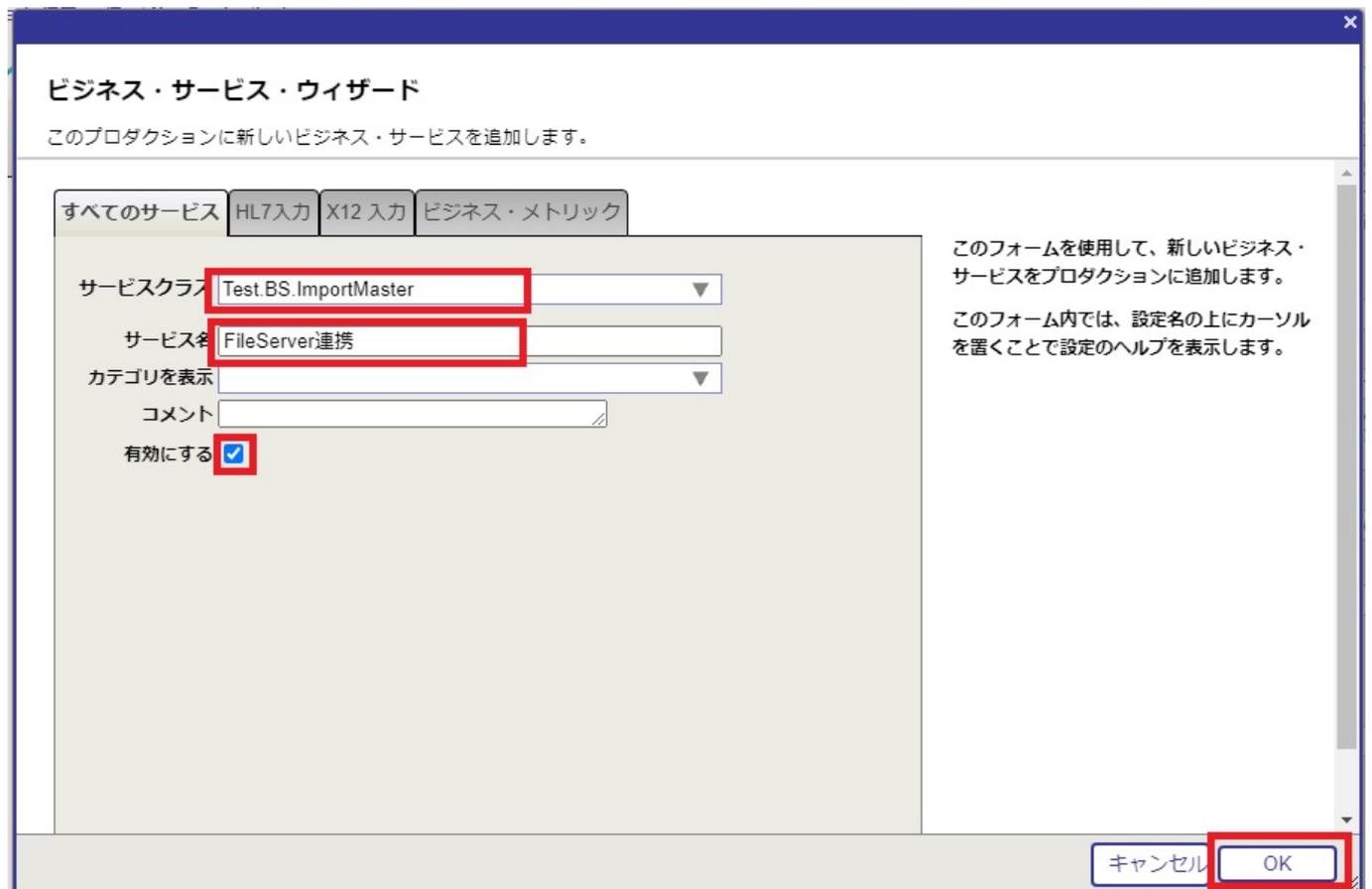
ビジネスサービスの追加

最後にビジネスサービスの設定を行います。

管理ポータル内の「InterOperability」「構成」「プロダクション」をクリックすると、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、サービスの右にある「+」ボタンをクリックします。



以下のダイアログボックスが表示されますので、「サービスクラス」には、先ほど修正したクラス「Test.BS.ImportMaster」とサービス名「FileServer連携」を入力、「有効にする」をチェックし、「OK」ボタンをクリックします。（サービス名は任意の名前で構いません）



ビジネスサービスが追加されましたので、以下の画面にて、作成したビジネスサービス「FileServer連携」をクリックし、「設定」タブをクリック、設定を行います。



ここで、以下の項目を入力し「適用」ボタンをクリックします。

- **ファイルパス** 監視するリモートサーバのディレクトリ。
- **アーカイブパス** 読み込んだファイルを格納するディレクトリ。
- **認証情報** 先ほど作成した認証情報のIDをドロップダウンリストから選択します。

プロダクションの開始

最後にプロダクション構成画面の「開始する」ボタンをクリックし、プロダクションを開始します。

これで、共有ディレクトリに対してファイルが出力されると、Test.BS.ImportMasterクラスのOnProcessInputメソッドが呼び出され、共有ディレクトリにあるファイルの取り込み処理が行えるようになります。取り込み処理については[前回の記事](#)をご参照ください。

また、今回作成したクラスやルーチンは[こちら](#)にアップロードしています。

ご質問等ございましたら、お気軽に書き込んでください。
よろしくお願いいたします。

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

ソースURL:

<https://jp.community.intersystems.com/post/iris%E3%81%AEinteroperability%E6%A9%9F%E8%83%BD%E3%82%92%E4%BD%BF%E3%81%A3%E3%81%9Fwindows%E5%85%B1%E6%9C%89%E3%83%95%E3%82%A9%E3%83%AB%E3%83%80%E3%81%A7%E3%81%AE%E3%83%95%E3%82%A1%E3%82%A4%E3%83%AB%E9%80%A3%E6%90%BA>