

Article

[Guillaume Rongier](#) · Déc 5, 2022 15m de lecture

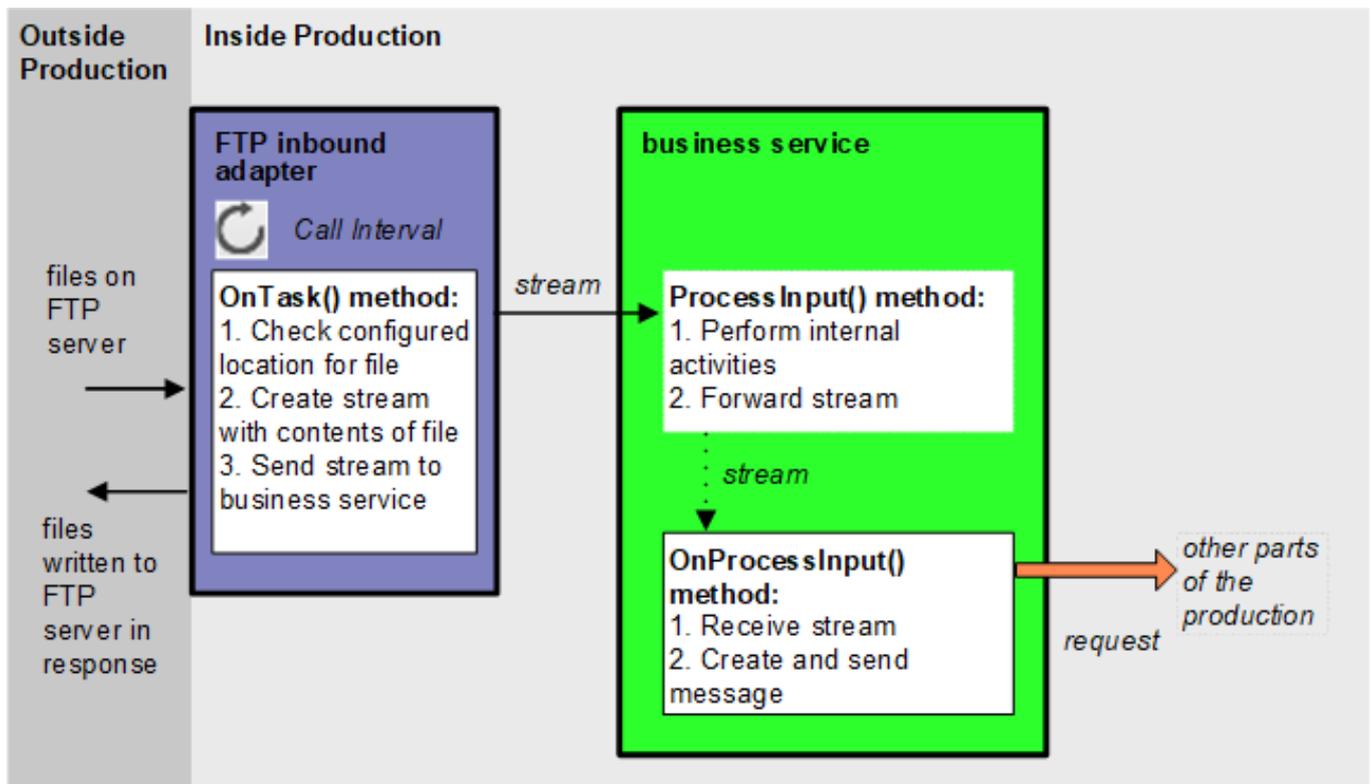
[Open Exchange](#)

Comment utiliser l'adaptateur FTP pour produire et consommer des messages FTP

FTP (Protocole de transfert de fichiers) est un protocole de réseau permettant de transmettre des fichiers sur des connexions TCP/IP dans un réseau (y compris l'Internet) configuré pour transférer des fichiers via ce protocole. Dans une transaction FTP, un expéditeur de fichiers est appelé hôte local. Un récepteur de fichiers impliqué dans le FTP est un hôte distant, et il s'agit généralement d'un serveur. Bien que de nombreux transferts de fichiers puissent être effectués à l'aide du protocole HTTP (Hypertext Transfer Protocol), le FTP est encore couramment utilisé pour transférer des fichiers en coulisse pour d'autres applications, comme les services bancaires. Par conséquent, nous pouvons dire que FTP est une excellente option pour promouvoir l'interopérabilité basée sur les fichiers entre des systèmes situés dans le même réseau local ou dans l'Internet (un serveur FTP est accessible via l'Internet). Cette interopérabilité peut être enrichie, traitée et orchestrée par le système d'interopérabilité IRIS Interoperability. Pour comprendre comment IRIS peut aider, il est crucial de comprendre la fonctionnalité de l'Architecture d'Interopérabilité IRIS pour FTP, centrée sur les adaptateurs FTP d'entrée et de sortie.

Adaptateur FTP d'entrée

Selon [la documentation InterSystems](#), l'EnsLib.FTP.InboundAdapter permet à InterSystems IRIS de recevoir des fichiers via le protocole FTP. L'adaptateur reçoit une entrée FTP à partir de l'emplacement configuré, la lit et l'envoie sous forme de flux au service métier associé. Le service métier créé et configuré par vos soins utilise ce flux et communique avec le reste de la production. La figure suivante montre le flux global des messages entrants (à l'exclusion des réponses) :



En détail :

1. Chaque fois que l'adaptateur rencontre une entrée provenant de sa source de données configurée, il appelle la méthode interne ProcessInput() de la classe de service métier, en passant le flux comme argument d'entrée.
2. Celle-ci est exécutée par la méthode interne ProcessInput() de la classe de service métier. Cette méthode effectue des tâches de production de base liées à la gestion des informations internes d'une manière requise par tous les services métier. Il n'est pas nécessaire de personnaliser ou de remplacer la méthode héritée par votre classe de service métier.
3. La méthode ProcessInput() appelle ensuite votre méthode OnProcessInput() personnalisée, en transmettant l'objet Stream en tant qu'entrée. Maintenant, avec le Stream, il est possible de développer les règles requises par le métier pour les fichiers reçus.
4. Le message de réponse suit le même chemin mais en sens inverse.

Adaptateur FTP de sortie

Selon la documentation d'InterSystems

(<https://docs.intersystems.com/iris20221/csp/docbook/DocBook.UI.Page.cls?..>), l'adaptateur FTP de sortie EnsLib.FTP.OutboundAdapter permet à la production d'envoyer des fichiers via le protocole FTP. Pour utiliser cet adaptateur, vous devez créer et configurer une opération métier spécialement conçue pour ce dernier. Cette opération métier recevra alors un message de la production, recherchera le type de message et exécutera la méthode appropriée. Cette méthode exécute généralement des règles pour générer des fichiers avec le contenu et le format requis par le métier à envoyer à une application.

Scénarios d'utilisation FTP les plus courants pour l'interopérabilité IRIS

Le tableau suivant résume les principaux scénarios d'utilisation de FTP pour l'interopérabilité d'IRIS :

Type d'adaptateur FTP	Scénario
D'entrée	Réception de lots de données métier dans des fichiers CSV, XML, TXT et JSON.
De sortie	Envoi des résultats du traitement des données métier par lots en CSV, XML, TXT et JSON à partir des données envoyées précédemment
D'entrée	Processus ETL (extraction, transformation et chargement) de gros fichiers comme parquet, ORC et avro pour les lacs de données Data Lakes et les dépôts de données Data Marts.
De sortie	Génération de fichiers parquet, ORC et avro à partir des bases de données IRIS pour les envoyer vers les lacs de données Data Lakes et les dépôts de données Data Marts externes.
D'entrée	Consommation asynchrone de messages et de fichiers provenant de systèmes externes
De sortie	Production de messages et de fichiers asynchrones vers des systèmes externes
De sortie	Envoi/publication des fichiers générés (rapports PDF, fichiers numérisés, images, etc.) vers les dossiers distants de certains organismes.

Application modèle

Le modèle IRIS-FTP (<https://openexchange.intersystems.com/package/iris-ftp-sample>) est un exemple d'utilisation de l'adaptateur FTP (d'entrée et de sortie).

Production

Ce modèle a une production d'interopérabilité vers :

Recevoir des données CSV à l'aide d'un adaptateur FTP d'entrée dans un service métier

LoadCSVFTPBusinessService ; envoyer son contenu à une opération métier LoadCSVFTPBusinessOperation ; persister le contenu dans une classe persistante à l'aide de CSVgen.

Effectuer un changement de données CDC (Change Data Capture) dans le tableau Paiement en utilisant un adaptateur SQL d'entrée dans une opération métier PaymentSQLBusinessService et envoyer les données du tableau (dans un fichier JSON) à une opération métier SendSQLDataToFTPBusinessOperation pour mettre le fichier JSON dans un serveur FTP.

Vous pouvez voir cette production ici :

Installation du modèle

Pour installer le modèle en utilisant ZPM, suivez les étapes suivantes :

1. Ouvrez l'espace de nommage IRIS Namespace avec l'interopérabilité activée.
2. Ouvrez le terminal et lancez un appel : `USER>zpm "install iris-ftp-sample"`

Pour installer le modèle en utilisant Docker, il faut procéder comme suit :

1. Clone/git tire le repo dans n'importe quel répertoire local :

```
$ clone git https://github.com/yurimarx/iris-ftp-sample.git
```

2. Ouvrez le terminal dans ce répertoire et exécutez :

```
$ compilation de docker-compose
```

3. Lancez le conteneur IRIS avec votre projet :

```
$ docker-compose up -d
```

Lancement du modèle

Pour exécuter le modèle, suivez les procédures suivantes :

1. Créez les informations d'identification pour accéder au serveur FTP (Allez vers Interopérabilité > Configurer > Informations d'identification) :

- ID : FTPCredentials
- Nom de l'utilisateur : irisuser
- Mot de passe : sys

Voici le résultat :

2. Ouvrez la production et lancez-la, comme indiqué dans l'image ci-dessous :

Envoi d'un fichier à l'aide du client FTP et visualisation des données chargées dans un tableau SQL

1. Ouvrez un client FTP (j'utilise généralement Filezilla pour le faire) et mettez le fichier input/countries.csv dans le dossier FTP racine :

- Hôte : localhost
- Nom de l'utilisateur : irisuser
- Mot de passe : sys

Ouvrez un client FTP (j'utilise généralement Filezilla pour le faire) et mettez le fichier input/countries.csv dans le dossier FTP racine :

2. Accédez à System Explorer > SQL et tapez la sélection du pays, de la latitude, de la longitude et du nom
SELECT country, latitude, longitude, name FROM dcirisftpsample.Country . Consultez l'image ci-dessous pour voir à quoi devraient ressembler les données CSV chargées dans le tableau SQL :

Insertion de données dans un tableau et un fichier avec le contenu inséré dans un fichier sur le serveur FTP

1. Accédez à System Explorer > SQL et tapez les valeurs de paiement (montant payeur, destinataire, date de transaction): insert into dcirisftpsample.Payment(amount payer, receiver, transactiondate) values(100.0,'Yuri','Fabiana', CURRENTTIMESTAMP) . Vous pouvez voir ici la représentation à l'écran du fichier JSON contenant les données insérées dans le serveur FTP :

En coulisses : le code source

Fichiers et dossiers principaux

Dans le dossier iris-ftp-sample nous trouvons les éléments suivants :

Fichier	Description
Readme.md	Informations sur le modèle et instructions d'installation/utilisation
Module.xml	Manifeste d'installation de ZPM
Dockerfile	Commandes Docker pour créer un conteneur IRIS d'InterSystems
Docker-compose.yml	Commandes pour créer une composition de 2 instances docker : 1 instance docker de serveur FTP et 1 instance docker d'InterSystems IRIS dans le même réseau Docker
Input/countries.csv	Fichier modèle à utiliser pour tester le chargement des données par FTP et csvgen avec une production FTP modèle.
src/dc/irisftpsample	Le dossier du code source dans le paquet dc.irisftpsample
FTPSampleProduction.cls	Le code source de la Production FTP (conteneur permettant d'exécuter des services et des opérations métier à l'aide d'adaptateurs FTP d'entrée et de sortie).
LoadCSVFTPBusinessService.cls	Classe ObjectScript qui utilise un adaptateur FTP d'entrée pour recevoir des fichiers CSV et envoyer leur contenu à l'opération LoadCSVFTPBusinessOperation.cls.
LoadCSVFTPBusinessOperation.cls	Reçoit le contenu d'un fichier CSV et charge ses données dans un tableau SQL en utilisant CSVgen
SendSQLDataToFTPBusinessOperation.cls	Reçoit un fichier JSON avec le contenu du tableau "Paiement" de PaymentSQLBusinessService et envoie ce fichier JSON au serveur FTP.
Payment.cls	Classe persistante (tableau SQL "Paiement") pour les

données de paiement persistantes à envoyer au serveur FTP.

Cas d'utilisation modèle 1 : Recevoir un fichier CSV du serveur FTP et charger les données CSV dans un tableau SQL.

Pour ce cas d'utilisation, nous avons la séquence suivante :

1. L'adaptateur FTP d'entrée lit le fichier CSV et le supprime du serveur FTP en utilisant ces paramètres :

Le Chemin du fichier (File Path) est l'endroit où l'adaptateur FTP se connectera pour trouver les fichiers.

La fonction de suppression du serveur (Delete From Server) permet de supprimer le fichier du serveur après sa lecture.

File Spec est le modèle qui correspond aux fichiers à lire.

L'intervalle d'appel (Call Interval) est l'intervalle du tirage (vérification des nouveaux fichiers).

Le serveur FTP est le nom de l'IP du serveur dans le réseau ou l'Internet.

Port FTP est le port à connecter

Les informations d'identification constituent la référence aux informations d'identification pour la configuration des noms d'utilisateur et des mots de passe.

2. Envoyez le flux d'un fichier CSV à la méthode OnProcessInput du service LoadCSVFTPBusinessService.
3. Envoyez le flux d'un fichier CSV dans un message Ens.StreamContainer dans l'opération LoadCSVFTPBusinessOperation.

```
Class dc.irisftpsample.LoadCSVFTPBusinessService Extends Ens.BusinessService
{
Parameter ADAPTER = "EnsLib.FTP.InboundAdapter";
Method OnProcessInput(pInput As %Stream.Object, pOutput As %RegisteredObject) As %Status
{
Set tSC = $$$OK,
tSource = pInput.Attributes("Filename"),
tFileLocation = pInput.Attributes("FTPSDir"),
pInput=##class(Ens.StreamContainer).%New(pInput)

Set tSC = ..SendRequestSync("LoadCSVFTPBusinessOperation", pInput, .pOutput, -1)

Quit tSC
}
}
```

4. Envoyez le flux d'un fichier CSV dans un message Ens.StreamContainer à la méthode LoadCSVIntoTable. L'adaptateur utilise le mappage XData pour déterminer la méthode à appeler pour chaque type de message dans la classe dc.irisftpsample.LoadCSVFTPBusinessOperation :

```
XData MessageMap
{

LoadCSVIntoTable
```

```
}
```

5. Envoyez le flux comme un fichier CSV temporaire et appelez la méthode Generate pour placer le fichier CSV dans le tableau SQL dc.irisftpsample.Country.

```
Class dc.irisftpsample.LoadCSVFTPBusinessOperation Extends Ens.BusinessOperation
{
    Parameter ADAPTER = "Ens.OutboundAdapter";
    Parameter INVOCATION = "Queue";
    Property Header As %Boolean [ InitialExpression = 1 ];
    Property Delimiter As %String [ InitialExpression = "," ];
    Property Classname As %String(MAXLEN = "");
    Property Prowtype As %String(MAXLEN = "");
    Property GuessTypes As %Boolean [ InitialExpression = 1 ];
    Property Append As %Boolean [ InitialExpression = 1 ];
    Property UseLoadData As %Boolean [ InitialExpression = 1 ];
    Parameter SETTINGS = "Header:Basic,Delimiter:Basic,Classname:Basic,Prowtype:Basic,Gue
ssTypes:Basic,Append:Basic,UseLoadData:Basic";
    Method LoadCSVIntoTable(pInput As Ens.StreamContainer, Output pOutput As Ens.StringRe
sponse) As %Status
    {
        Set tSC = $$$OK
        Set pOutput = ##class(Ens.StringResponse).%New()

        Try {
            Set tSC = ##class(community.csvgen).StreamToFile(pInput.Stream, .pFile)
            Set tSC = ##class(community.csvgen).Generate(pFile, ..Delimiter, ..Classname,
..Prowtype, ..GuessTypes, ..Append, ..UseLoadData, ..Header)
            Set pOutput.StringValue = "OK"
        } Catch err {
            Set pOutput.StringValue = $System.Status.GetOneErrorText(tSC, 1)
        }

        Quit tSC
    }
}

XData MessageMap
{
    LoadCSVIntoTable
}
}
```

- Remarque 1 : Le paramètre Adapter est utilisé pour choisir l'adaptateur à utiliser par la Business Operation.
- Remarque 2 : les paramètres tels que Header, Delimiter, Classname, Prowtype, GuessTypes, Append et UseLoadData sont requis par csvgen et configurés par l'utilisateur de Production car ces paramètres sont référencés dans le paramètre SETTINGS de la section Basic :

Consultez les détails sur les informations requises par CSVgen ici : <https://github.com/evshvarov/csvgen>.

Cas d'utilisation modèle 2 : Application de la méthode CDC (Change Data Capture) sur le tableau des paiements en utilisant l'adaptateur SQL d'entrée et en envoyant les données du tableau à une opération métier pour les mettre sur le serveur FTP.

Pour ce cas d'utilisation, nous avons la séquence suivante :

1. IRIS effectue une opération CDC dans le tableau "Paiement" et génère un fichier JSON avec le contenu du tableau, en supprimant les données lues. L'adaptateur utilise ces paramètres pour déterminer le tableau et les données à lire :

L'intervalle d'appel (Call Interval) : les nouvelles données sont vérifiées en quelques secondes

DSN : l'espace de nom si le tableau existe.

Target Config Names : le composant de la production qui recevra les données (dans ce cas, SendSQLDataToFTPBusinessOperation).

Requête : la commande de sélection pour obtenir les données à lire (dans ce cas, il s'agit de données relative au paiement (montant payeur, destinataire, date de transaction), à savoir SELECT ID, amount, payer, receiver, transactiondate FROM dcurisftpsample.Payment).

Requête de suppression : la commande de suppression utilisée pour supprimer les données lues, si nécessaire (dans ce cas DELETE FROM dcurisftpsample.Payment).

Nom du champ clé : la clé primaire du tableau.

2. Envoyez le flux de données au format JSON au PaymentSQLBusinessService.
3. Envoyer le flux du fichier JSON dans un message Ens.StreamContainer dans l'opération métier SendSQLDataToFTPBusinessOperation.
4. Envoyez le flux du fichier JSON à l'adaptateur FTP de sortie :

```
Class dc.irisftpsample.SendSQLDataToFTPBusinessOperation Extends Ens.BusinessOperation
{
Parameter ADAPTER = "EnsLib.FTP.OutboundAdapter";
Property Filename As %String(MAXLEN = 1000);
Parameter SETTINGS As %String = "Filename";
Method OnMessage(pRequest As Ens.StreamContainer, Output pResponse As %Persistent) As %Status
{
Set tFilename=
..Adapter.CreateTimestamp(##class(%File).GetFilename("/tmp/sqltojson"),..Filename)
Quit ..Adapter.PutStream(tFilename, pRequest.Stream)
}
}
```

- Remarque 1 : Le paramètre ADAPTER définit que l'opération métier utilisera l'adaptateur FTP de sortie.
- Remarque 2: Le paramètre Filename définit le nom du fichier qui se trouve sur le serveur FTP.

5. Placez le fichier JSON dans le dossier FTP racine avec le nom SQLTOJSONYYYY-MM-DDHH.mm.ss.zzz.json en utilisant les paramètres suivants :

Le serveur FTP est le nom de l'IP du serveur dans le réseau ou l'Internet.

Port FTP est le port à connecter

Les informations d'identification constituent la référence aux informations d'identification pour la configuration du nom d'utilisateur et du mot de passe

Le chemin du fichier est l'endroit où l'adaptateur FTP se connectera pour trouver les fichiers

Nom du fichier est le modèle du nom du fichier sur le serveur FTP

Protocole : FTP ou SFTP (FTP avec certificat SSL)

Utiliser PASV : connexions passives avec le serveur FTP

Pour en savoir plus

Documentation officielle pour les adaptateurs FTP :

<https://docs.intersystems.com/iris20221/csp/docbook/DocBook.UI.Page.cls?KEY=EFTP>

Apprentissage officiel en ligne sur la production, les services métiers et les opérations :

<https://learning.intersystems.com/course/view.php?id=1437>

Dépôt Git pour CSVGEN : <https://github.com/evshvarov/csvgen>

À propos de FTP : <https://www.techtarget.com/searchnetworking/definition/File-Transfer-Protocol-FTP>

[#FTP](#) [#Interopérabilité](#) [#InterSystems IRIS](#) [#InterSystems IRIS for Health](#)

[Voir l'application sur InterSystems Open Exchange](#)

URL de la source: <https://fr.community.intersystems.com/post/comment-utiliser-ladaptateur-ftp-pour-produire-et-consommer-des-messages-ftp>