

Utiliser QNS dans StrataFrame

Query Notification Services (QNS) est une fonctionnalité proposée par SQL Server depuis la version 2005.

Ce service permet aux clients SQL d'être automatiquement avisés par le serveur SQL de tout changement apporté à des données qu'ils ont requêté.

Au moment où il requête un jeu d'enregistrements, le client va s'abonner à une demande de notification. Cette requête peut être directe dans un SELECT, ou bien être incluse dans une procédure stockée appelée par le client.

Si le client abonné est déconnecté, le serveur SQL le reconnectera automatiquement pour l'aviser de l'existence de modifications !

I. Préalables

En arrière plan, sur le serveur SQL, c'est le Service Broker qui fait fonctionner QNS. Côté client, QNS repose sur les classes SqlDependency et SqlNotificationRequest de .Net. Il y a donc des prérequis côté serveur et côté client.

A. Côté Serveur

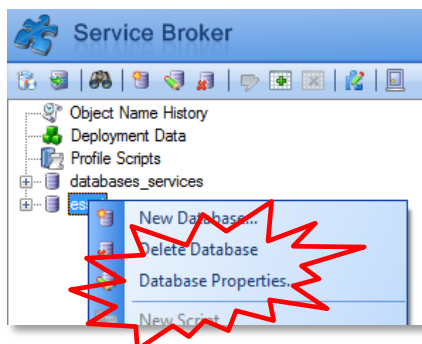
1. Activation de Service Broker

Ce service doit être activé sur la base de données cible.

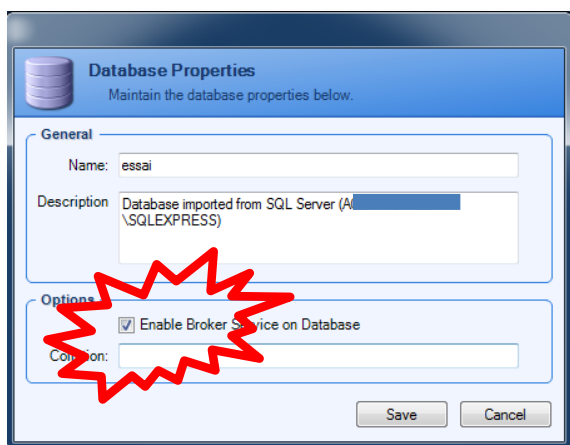
a) En utilisant DDT

Si vous disposez de DDT, il vous suffit de cocher la case appropriée dans les propriétés du database :

Un click droit sur le nœud du database, demandons les propriétés

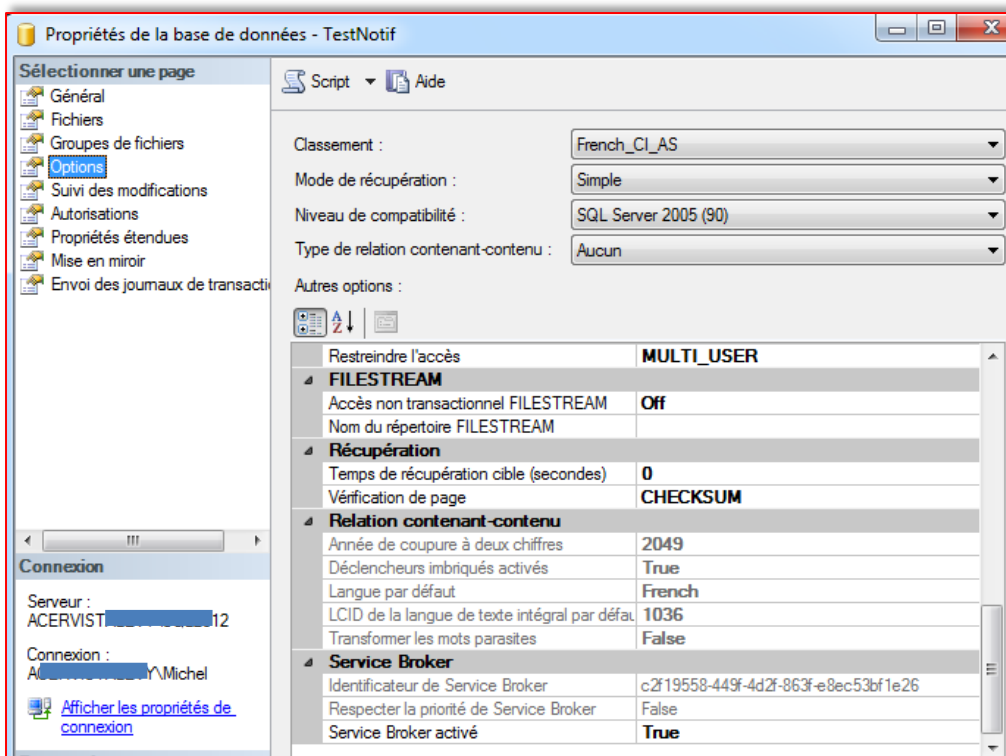


Pour activer le Service Broker sur cette database, cochez la case !



b) Sans utiliser DDT

(1) Dans SSMS



(2) Ou bien par script

Il suffit de lancer l'instruction suivante :

```
ALTER DATABASE [ma_database] SET ENABLE_BROKER
```

2. Donner les droits appropriés

Assurez-vous que les utilisateurs de la base de données auront les droits nécessaires pour faire fonctionner QNS. Ils doivent pouvoir s'abonner aux notifications, créer des procédures, créer des queues et des services :

```
GRANT SUBSCRIBE QUERY NOTIFICATIONS TO Test
GRANT CREATE PROCEDURE to Test
GRANT CREATE QUEUE TO test
GRANT CREATE SERVICE TO Test
```

B. Côté Client

1. Formulation de la requête

Pour pouvoir s'abonner à une notification, une requête doit respecter les mêmes règles que pour une vue indexée ; vous devez donc **IMPÉRATIVEMENT** qualifier les objets au minimum sur 2 parties. Le SELECT * est donc interdit (le nom de chaque colonne est obligatoire), et le nom de la table doit être précédé du nom du schéma.

Avec StrataFrame, si vous utilisez les 2 propriétés **AllFieldsName** et **TableNameAndSchema** pour construire vos requêtes, elles seront correctes pour QNS.

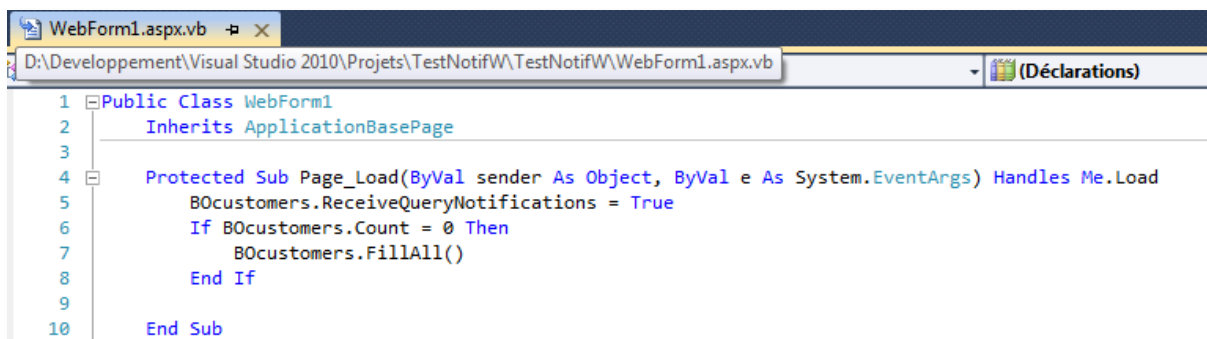
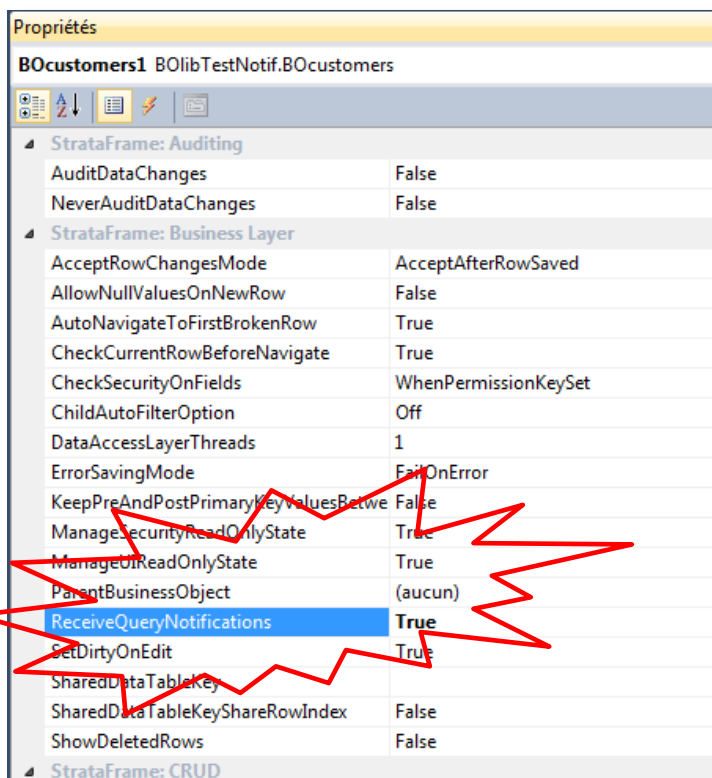
II. Utilisation

Sur le client, l'utilisation de QNS se passe en 2 étapes asynchrones :

- La première étape consiste à « abonner » une requête ; c'est cet abonnement qui va instancier les objets complémentaires au SQLCommand, ces objets envoyés avec la requête créeront les services, les messages et les files d'attente sur le Service Broker de la base de données impactée.
- La deuxième étape consiste à récupérer les messages retournés au client par Service Broker l'informant que les données qu'il avait sélectionné ont été modifiées par un autre abonné.

A. Abonnement du Business Object

Il suffit de mettre à **True** la propriété **ReceiveQueryNotifications** du BO, soit en mode de design, soit dans le code. Pour les applications WinForms, vous le ferez au niveau du Form, et pour les applications en Aspx, vous le ferez au niveau de la page elle-même.



Et c'est tout ! StrataFrame se charge de créer les objets nécessaires à l'abonnement.

B. Traitement

Les messages retournés par QNS sont routés par StrataFrame jusqu'à l'événement de BO **ServerDataChanged**. La méthode associée à cet événement reçoit l'évènement en paramètre, et la propriété Info de cet évènement est de

type `System.Data.SqlClient.SqlNotificationInfo`. Cette propriété peut prendre les valeurs **Insert** (1), **Update** (2), ou **Delete** (3).

Par exemple, dans ce WinForm en Vb.Net, (à condition que l'utilisateur ne soit pas déjà en train de modifier les données, seulement s'il est en train de les consulter), j'ai choisi de :

- pour un **Insert** ou un **Delete**, recharger automatiquement les données en affichant juste un WaitWindow pour l'en informer (le Sleep dans le code ci-dessous n'est là que pour permettre à l'utilisateur de lire le WaitWindow, sinon, le rechargement des données est tellement rapide qu'on n'a pas le temps !)
- pour un Update, demander à l'utilisateur s'il veut actualiser les données par le biais d'une InfoBox à menu contextuel.

```
Private Sub B0customers1_ServerDataChanged(  
    e As System.Data.SqlClient.SqlNotificationEventArgs  
    ) Handles B0customers1.ServerDataChanged  
  
    If Me.B0customers1.EditingState =  
        MicroFour.StrataFrame.Business.BusinessEditingState.Idle Then  
  
        Select Case e.Info  
            Case Data.SqlClient.SqlNotificationInfo.Insert  
                With Me.WaitWindow1  
                    .Title =  
                        "De nouveaux enregistrements ont été ajoutés sur le serveur !"  
                    .ShowWaitWindow()  
                    Me.B0customers1.FillAll()  
                    System.Threading.Thread.Sleep(1500)  
                    .HideWaitWindow()  
                End With  
  
            Case Data.SqlClient.SqlNotificationInfo.Delete  
                With Me.WaitWindow1  
                    .Title =  
                        "Des enregistrements ont été supprimés sur le serveur !"  
                    .ShowWaitWindow()  
                    Me.B0customers1.FillAll()  
                    System.Threading.Thread.Sleep(1500)  
                    .HideWaitWindow()  
                End With  
  
            Case Data.SqlClient.SqlNotificationInfo.Update  
  
                With InfoBox.AlertBoxSettings  
                    .CardinalPosition = MessagingCardinalPosition.SouthEast  
                    .Timeout = 3000  
                    .Sound = MessagingSounds.Warning  
                End With  
  
                InfoBox.AlertBox("Attention !",  
                    "Les données ayant été modifiées sur le serveur" &  
                    vbCrLf &  
                    "par un autre utilisateur," & vbCrLf &  
                    "veuillez indiquer si vous voulez les actualiser localement :",  
                    Me, ContextMenuStrip1)  
  
            End Select  
  
        End If  
  
    End Sub
```

C. Conseils d'utilisation

QNS ne remplace pas la gestion des accès concurrentiels, et n'est pas non plus un service de communication interpersonnel comme le chat ou le mail. Un usage abusif de QNS va rajouter un grand nombre de queues et de Timer au Service Broker de votre database, risquant de faire baisser les performances de votre serveur.

Définissez attentivement les utilisateurs qui en ont réellement besoin, et les données sur lesquelles portent les abonnements. Il est parfois préférable de passer par un abonnement à une table dédiée qui contiendra des informations résumées.

Annexes

1. Vous trouverez plus d'information sur MSDN en ligne :

Notifications de requête dans SQL Server :

[http://msdn.microsoft.com/fr-fr/library/t9x04ed2\(VS.90\).aspx](http://msdn.microsoft.com/fr-fr/library/t9x04ed2(VS.90).aspx)

Activation des notifications de requête :

[http://msdn.microsoft.com/fr-fr/library/ms172133\(VS.90\).aspx](http://msdn.microsoft.com/fr-fr/library/ms172133(VS.90).aspx)

Détection des modifications à l'aide de SqlDependency :

[http://msdn.microsoft.com/fr-fr/library/62xk7953\(VS.90\).aspx](http://msdn.microsoft.com/fr-fr/library/62xk7953(VS.90).aspx)

Exécution de SqlCommand avec un SqlNotificationRequest :

[http://msdn.microsoft.com/fr-fr/library/wd2x83zk\(VS.90\).aspx](http://msdn.microsoft.com/fr-fr/library/wd2x83zk(VS.90).aspx)

SqlDependency dans une application ASP.NET :

[http://msdn.microsoft.com/fr-fr/library/9dz445ks\(VS.90\).aspx](http://msdn.microsoft.com/fr-fr/library/9dz445ks(VS.90).aspx)

2. Dans le code source de StrataFrame

Dans la classe SqlDataSourceItem (dans le namespace Data), vous verrez que la méthode GetDataTable est overridee pour créer un objet SqlDependency. Vous retrouverez cet objet dans la classe DataLayer, ce sera le Sender qui est passé à la méthode NotificationReceived.

3. Requêtes utiles

Pour observer le fonctionnement de ce service sur votre serveur SQL, voici quelques requêtes à exécuter dans le contexte de la database cible :

```
select * from sys.service_queues
select * from sys.services
select * from sys.dm_broker_queue_monitors
select * from sys.dm_qn_subscriptions
select * from sys.internal_tables
```