

Data Tier Application avec SQL Server 2008 R2



Article par David BARBARIN (MVP SQL Server)

David BARBARIN est actuellement consultant en bases de données pour une entreprise Suisse GOLD Partner Microsoft et participe au développement d'offres à valeur ajoutée de data management autour de la technologie Microsoft SQL Server. Intervenant en tant que speaker à deux reprises lors des sessions [GUSS](#) et [SQLPass Suisse](#) en 2010, David a également écrit de nombreux articles et billets de blog. De plus, il participe activement aux communautés SQL Server sur [Developpez.com](#), [Technet SQL Server](#) et [Beyond Relational](#).

Expert en bases de données et performances SQL Server, David met son expérience au service des différents clients pour lesquels il a pu travailler.

QU'EST-CE QUE DAC ?

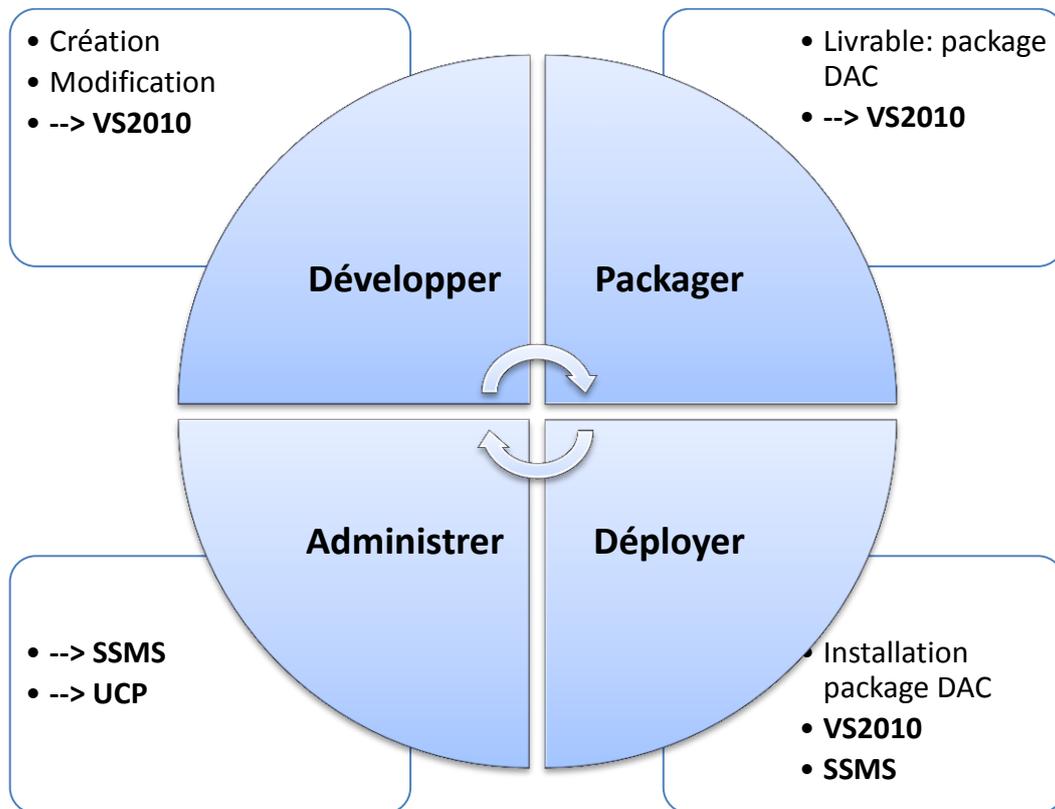
« Data Tier Application » est une nouvelle fonctionnalité introduite avec SQL Server 2008 R2 qui permet de rationaliser les différentes étapes du cycle de vie d'une application. Elle concerne surtout les applications qui reposent sur une architecture dirigée par les données (Data Driven Architecture).

Il va s'en dire que cette fonctionnalité ne sera d'une grande utilité que pour les entreprises ayant déjà mis en place un processus de gestion des changements de leur application. Data Tier Application facilite la mise à jour d'une application en proposant une automatisation de tâches à différents niveaux de ce processus (détection des changements, développement, packaging, déploiement en production ...). Cette fonctionnalité concerne aussi bien les développeurs que les administrateurs de bases de données avec une amélioration de l'interopérabilité entre Visual Studio 2010 et SQL server.

QU'EST-CE QU'UN PACKAGE DAC ?

Une application gouvernée par les données peut être comparée à une entité qui peut contenir un certain nombre d'objets au niveau instance ou au niveau base de données comme les tables, les vues, les index, les procédures stockées ou encore des fichiers, des scripts qui définissent les exigences d'un déploiement. Cet ensemble de composant est communément appelé package DAC (Data Tier Application Components) et sera utilisé par les différents acteurs (développeurs et administrateurs de bases de données) tout au long du cycle de vie de l'application. Un développeur utilisera Visual Studio 2010 (Edition Professional, Premium ou Ultimate) pour opérer les modifications nécessaires dans l'application alors qu'un administrateur de bases de données préférera utiliser SQL Server Management Studio pour la déployer et la gérer. Enfin une application DAC installée sur une instance SQL Server est appelée instance DAC.

LE CYCLE DE VIE D'UNE APPLICATION DAC



- VS210 : Visual Studio 2010
- SSMS : SQL Server Management Studio
- UCP : Utility Control Point

DAC s'inscrit dans le schéma traditionnel du cycle de vie de conception d'une application. Il existe deux méthodes de génération de package DAC et deux méthodes de déploiement.

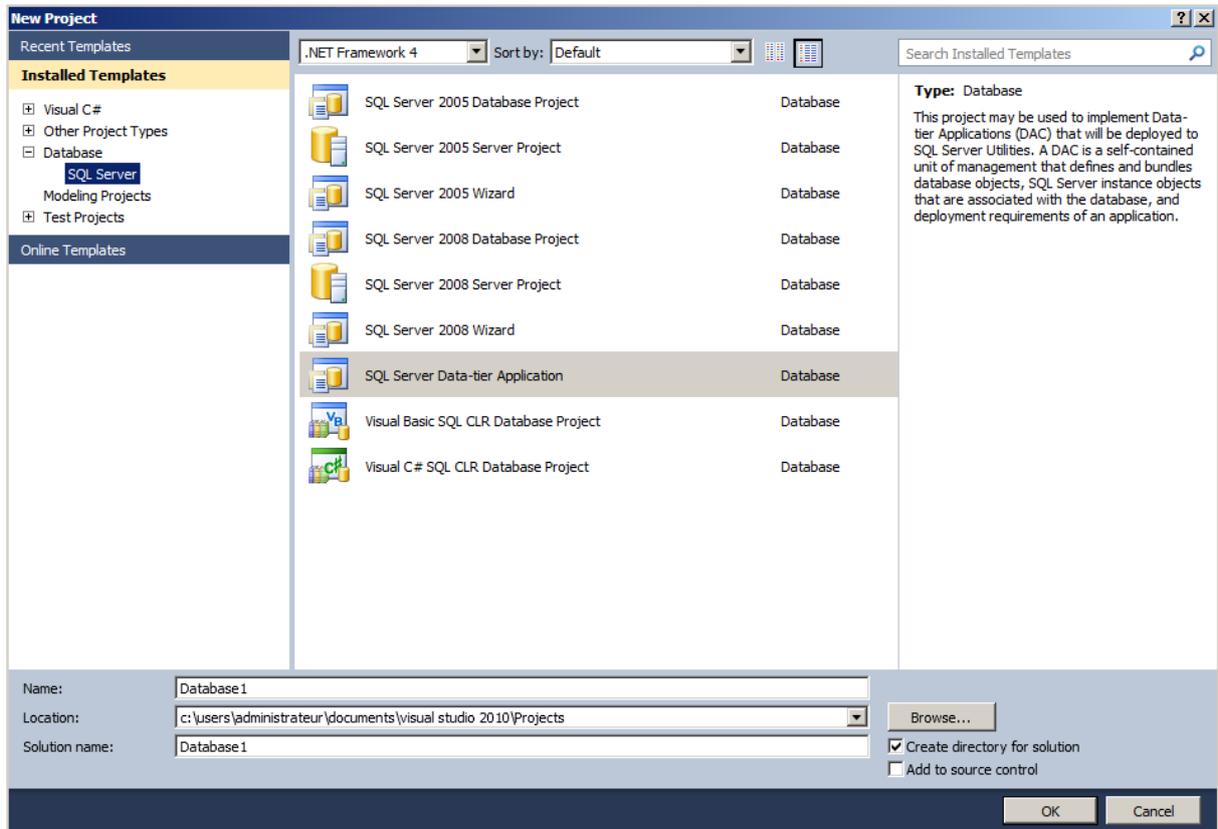
Un package DAC peut être généré depuis SQL Server Management Studio pour une base de données existante ou depuis Visual Studio 2010.

Chaque package peut être ensuite déployé soit par un administrateur de bases de données par le biais de l'assistant prévu à cet effet dans SQL Server Management Studio soit directement par un développeur qui possède les droits nécessaires depuis Visual Studio 2010.

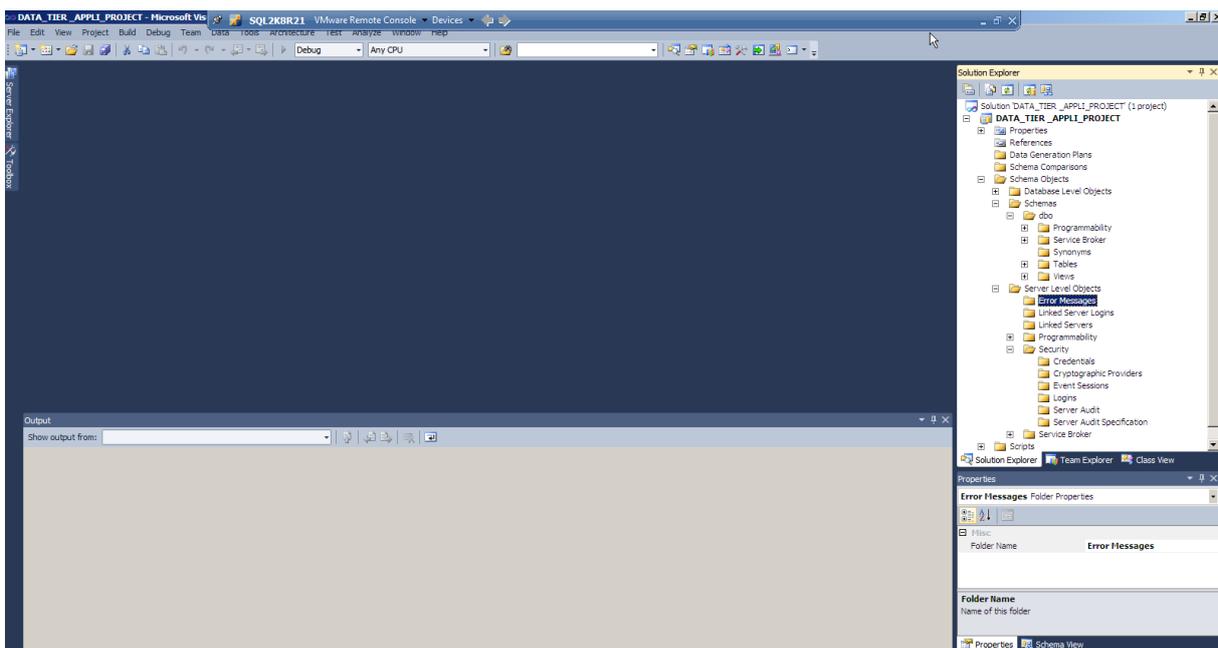
Le résultat d'un déploiement se traduit par une application DAC associée à une base de données et un enregistrement des métadonnées concernant l'application DAC dans la base de données msdb.

CREATION D'UN PROJET DAC AVEC VISUAL STUDIO 2010

Visual Studio 2010 propose un modèle de création pour les projets de DAC :



Nous créerons un projet nommé DATA_TIER_APPLI_PROJECT avec le chemin par défaut proposé par Visual Studio. L'espace de travail devient le suivant :

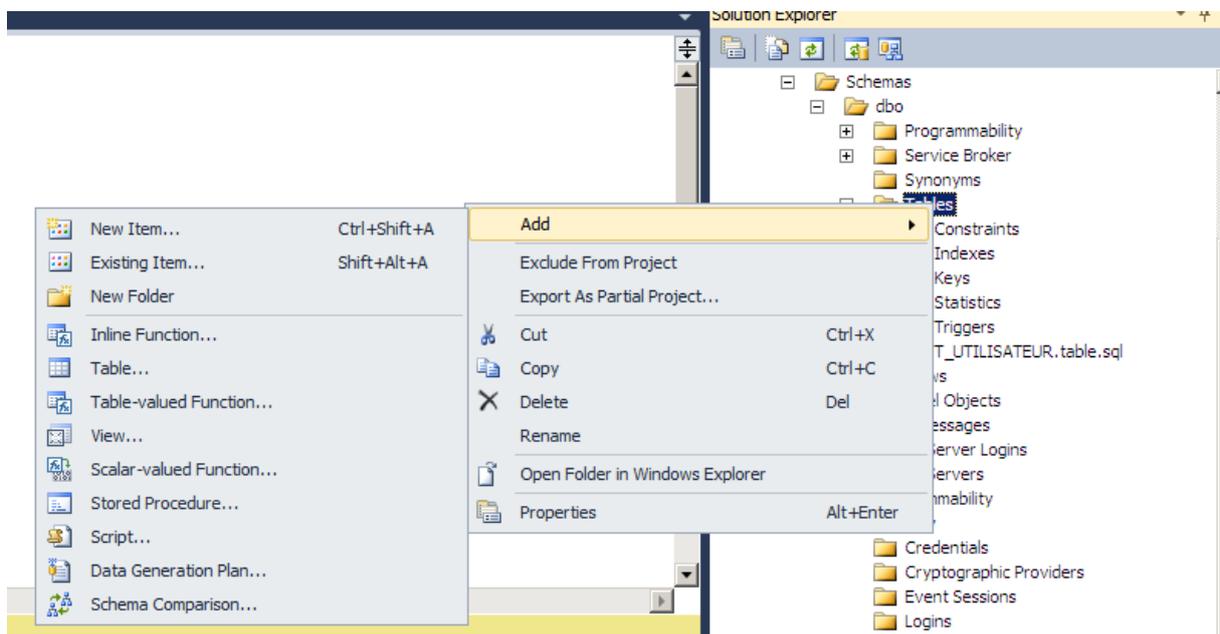


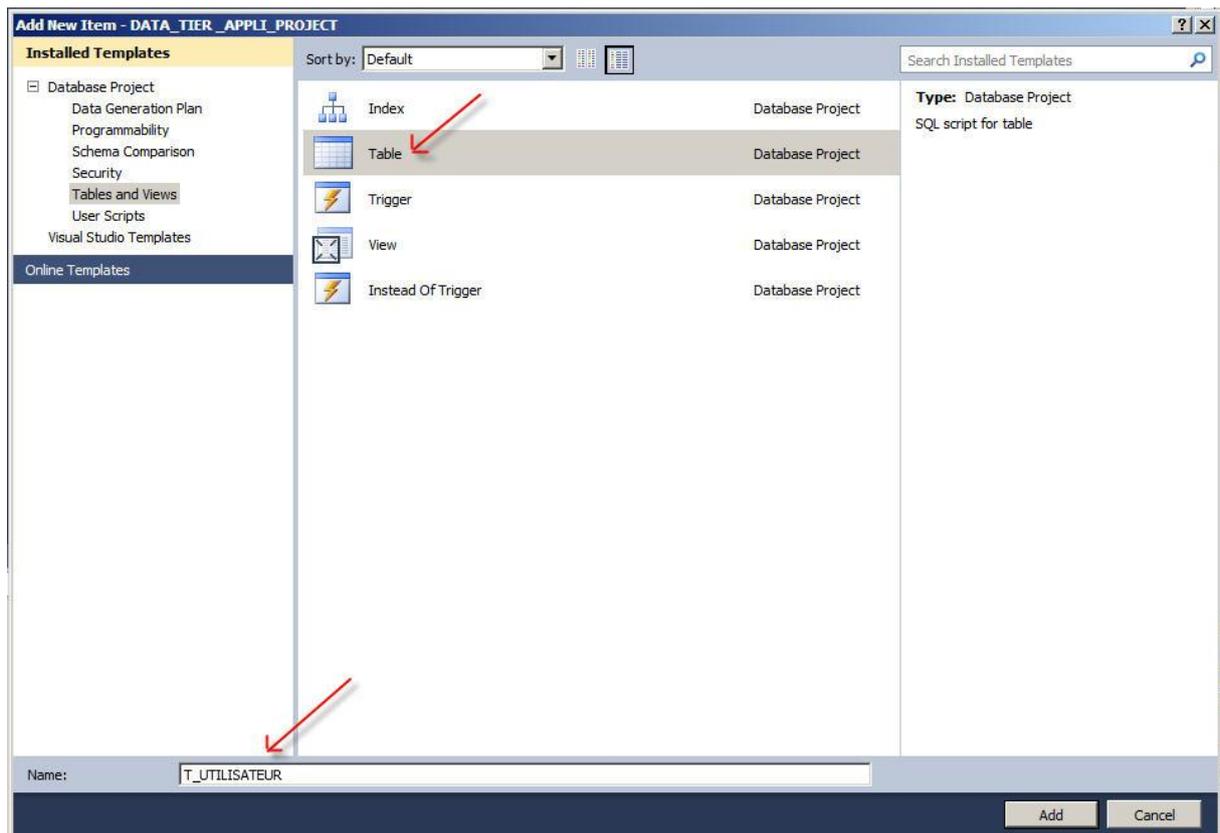
La partie de droite concerne l'explorateur d'objets. Dans ce projet les objets suivants seront créés :

- Une table T_UTILISATEUR
- Une table T_NEWS
- Une vue V_UTILISATEUR_NEWS
- Une procédure stockée PS_POST_NEWS

Création des tables

Pour créer un script de table il suffit de sélectionner le nœud **Tables** du projet et de faire **clic droit > add > table**

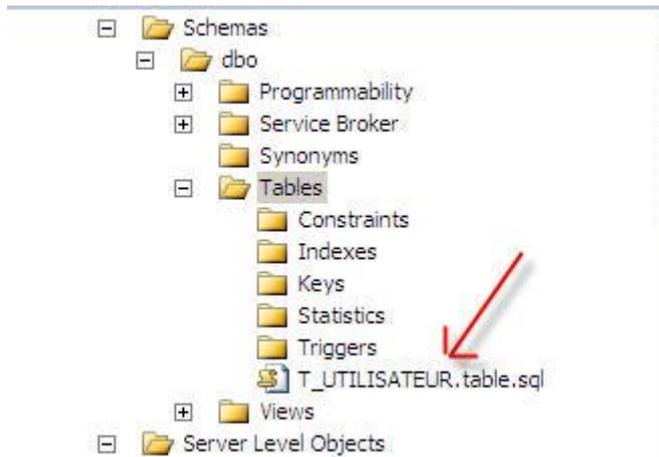




Le script de création de la table est le suivant :

```
CREATE TABLE [dbo].[T_UTILISATEUR]
(
    id INT IDENTITY PRIMARY KEY,
    nom VARCHAR(50) NOT NULL,
    prenom VARCHAR(50) NOT NULL
)
```

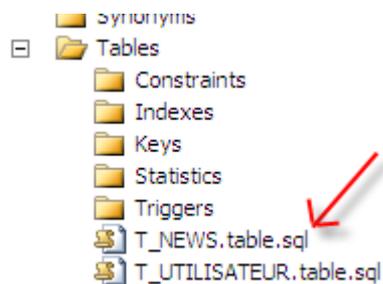
Un fichier **T_UTILISATEUR** est maintenant présent sous le nœud **tables** avec la commande DDL précédente :



Le script de création pour la table T_NEWS est le suivant :

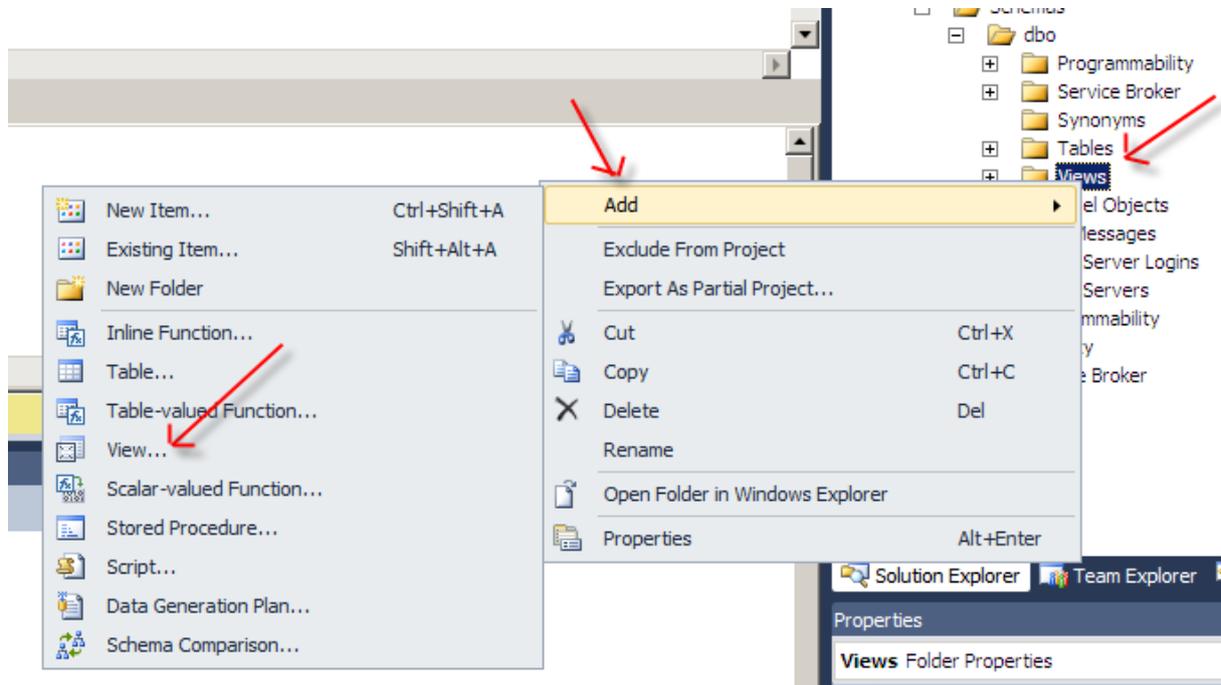
```
CREATE TABLE [dbo].[T_NEWS]
(
    id INT IDENTITY PRIMARY KEY,
    id_parent INT NULL,
    usr_id INT NOT NULL,
    titre VARCHAR(20) NOT NULL,
    commentaire VARCHAR(max) NOT NULL,
    CONSTRAINT FK_T_NEWS_id_parent FOREIGN KEY (id_parent) REFERENCES T_NEWS(id),
    CONSTRAINT FK_T_UTILISATEUR_id FOREIGN KEY (usr_id) REFERENCES T_UTILISATEUR(id)
);
```

Un deuxième fichier apparaît sous le nœud Tables :



Création de la vue

La création d'une vue s'effectue de la même façon que pour une table. Il suffit de sélectionner le bon objet dans la fenêtre :



Note : se positionner sur le nœud View n'est pas une obligation. Il est possible par la suite de déplacer le fichier script généré avec un drag & drop par exemple.

Comme précédemment il suffit d'insérer la DDL suivante et d'enregistrer pour créer le fichier script associé.

```
CREATE VIEW [dbo].[V_UTILISATEUR_NEWS]
AS

SELECT
    U.prenom + ' ' + U.nom AS [user_name],
    N.titre AS titre,
    N.commentaire
FROM T_NEWS AS N
INNER JOIN T_UTILISATEUR AS U
ON N.usr_id = U.id;
```

Note : la fonctionnalité IntelliSense peut être utilisée en mode déconnecté.

```
CREATE VIEW [dbo].[V_UTILISATEUR_NEWS]  
AS
```

```
SELECT *
```

```
FROM T_NEWS AS T
```

```
INNER JOIN T
```

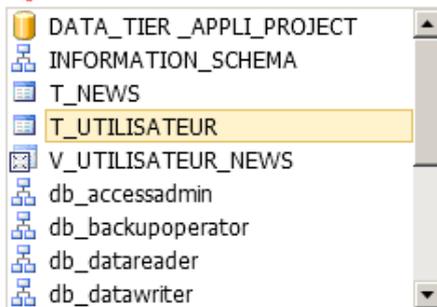


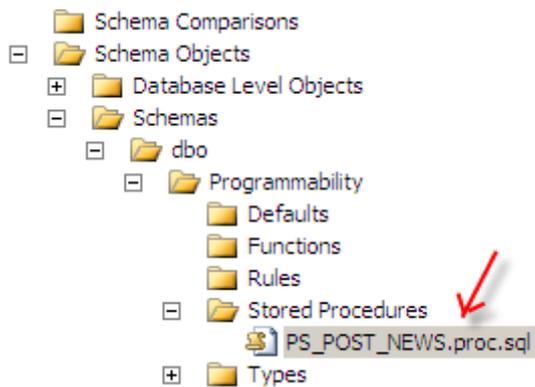
table DATA_TIER_APPLI_PROJECT.dbo.T_UTILISATEUR

Création de la procédure stockée

Comme vous l'avez sans doute compris la création d'objets est relativement simple. La DDL de la procédure stockée à enregistrer est la suivante :

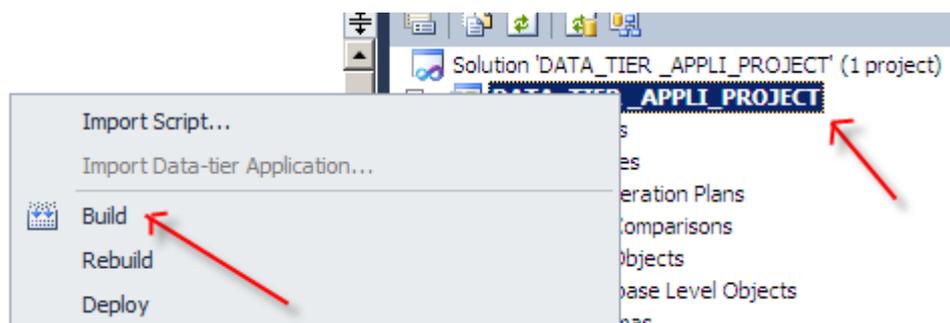
```
CREATE PROCEDURE [dbo].[PS_POST_NEWS]  
(  
    @id_user INT,  
    @id_parent_news INT = NULL,  
    @titre VARCHAR(50),  
    @commentaire VARCHAR(MAX)  
)  
AS  
  
INSERT INTO dbo.T_NEWS (id_parent, usr_id, titre, commentaire)  
VALUES (@id_parent_news, @id_user, @titre, @commentaire)
```

Le fichier de script SQL est présent sous le nœud « Stored Procedures » :

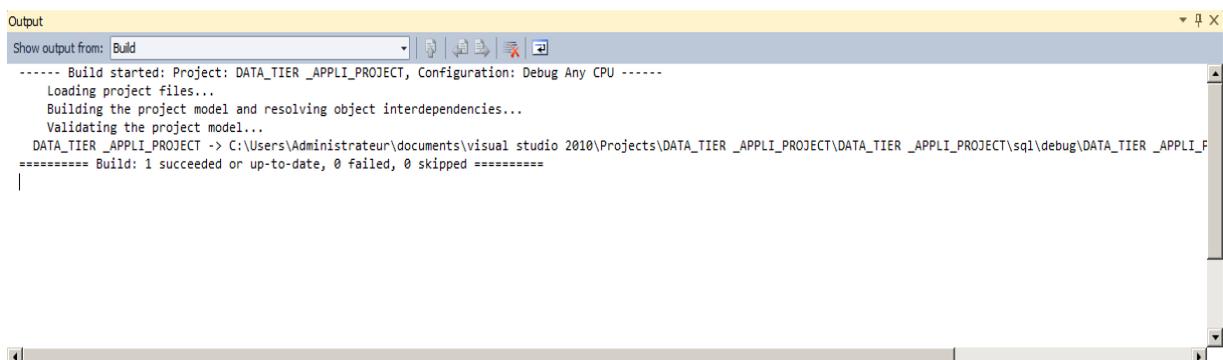


GENERATION D'UN PACKAGE DAC

La phase suivante consiste à générer le package DAC. Cette génération s'effectue au moyen du menu BUILD du projet.

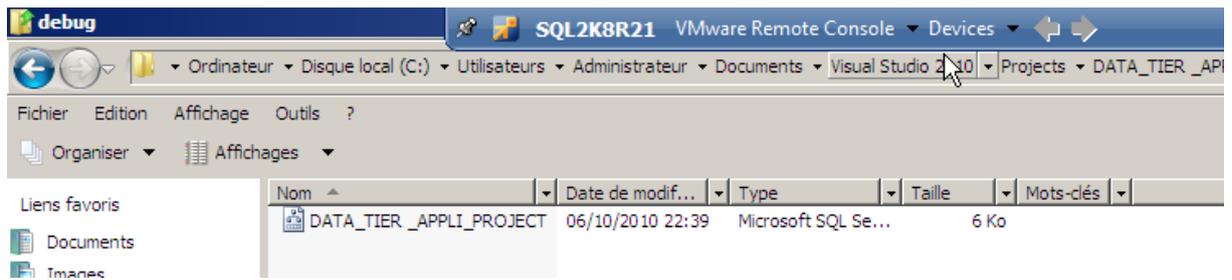


La génération du package produit la sortie suivante dans Visual Studio 2010 :



Le fichier a été généré sans erreurs dans le répertoire par défaut :

C:\Users\Administrateur\documents\visual studio 2010\Projects\DATA_TIER _APPLI_PROJECT\DATA_TIER _APPLI_PROJECT\sql\debug et porte le nom *DATA_TIER _APPLI_PROJECT.dacpac*



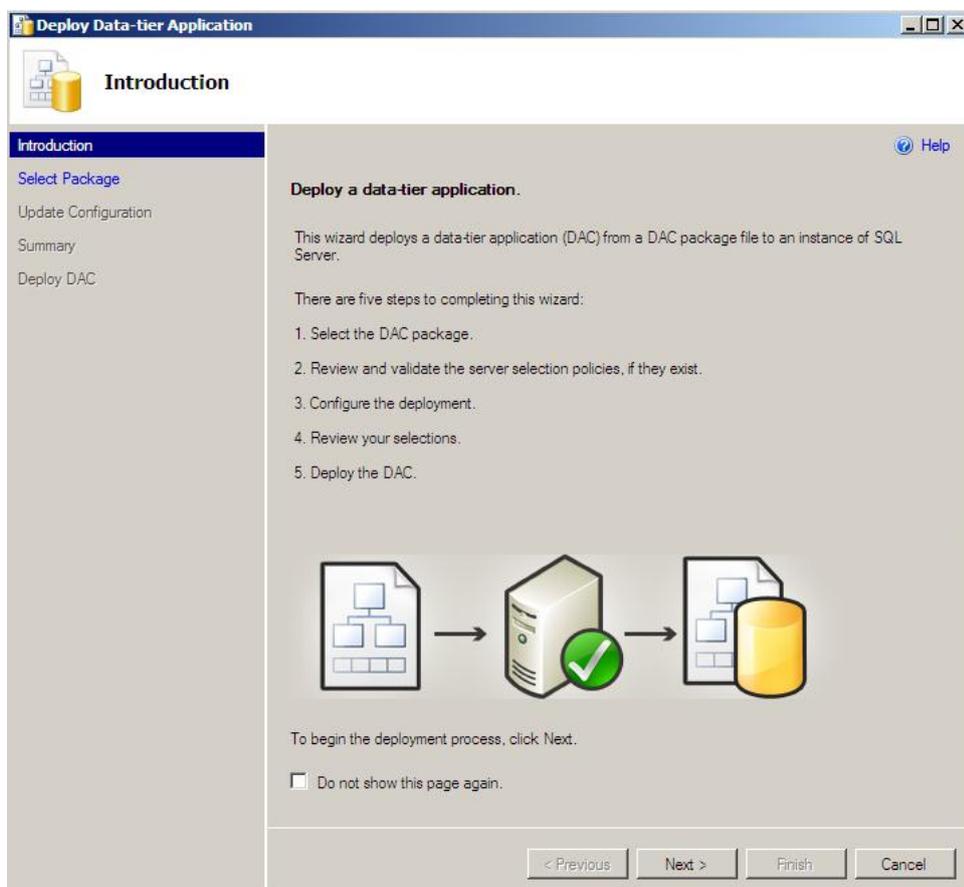
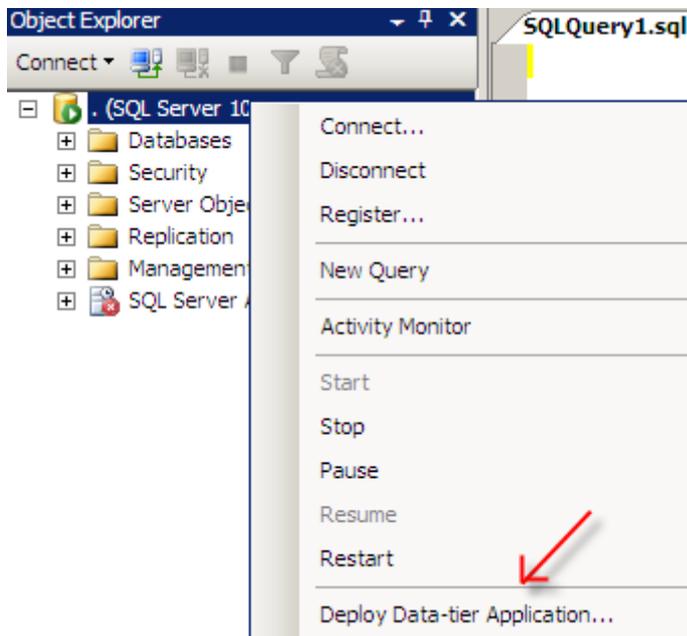
Un package DAC est en réalité une archive constituée de plusieurs fichiers.

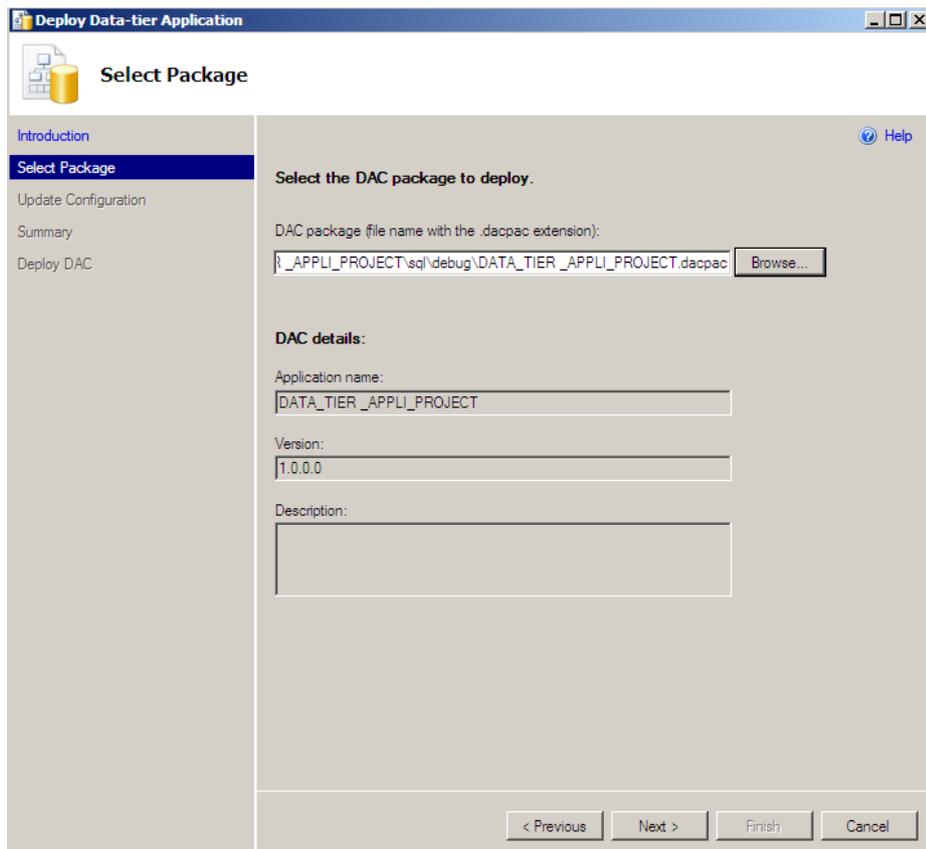
- Un fichier de script SQL qui contient les commandes de création des objets de la base de données
- Deux autres fichiers de script SQL concernant les commandes de pré et post déploiement
- Fichiers XML de configuration

Nom	Date de modif...	Type	Taille	Mots-clés
DacMetadata.xml	16/11/2010 22:46	Document XML	1 Ko	
DATA_TIER_APPLI_PROJECT.sql	16/11/2010 22:46	Microsoft SQL Se...	3 Ko	
LogicalObjectStream.xml	16/11/2010 22:46	Document XML	24 Ko	
PhysicalObjectStream.xml	16/11/2010 22:46	Document XML	6 Ko	
Script.PostDeployment.sql	16/11/2010 22:46	Microsoft SQL Se...	1 Ko	
Script.PreDeployment.sql	16/11/2010 22:46	Microsoft SQL Se...	1 Ko	
TargetSelection.xml	16/11/2010 22:46	Document XML	4 Ko	

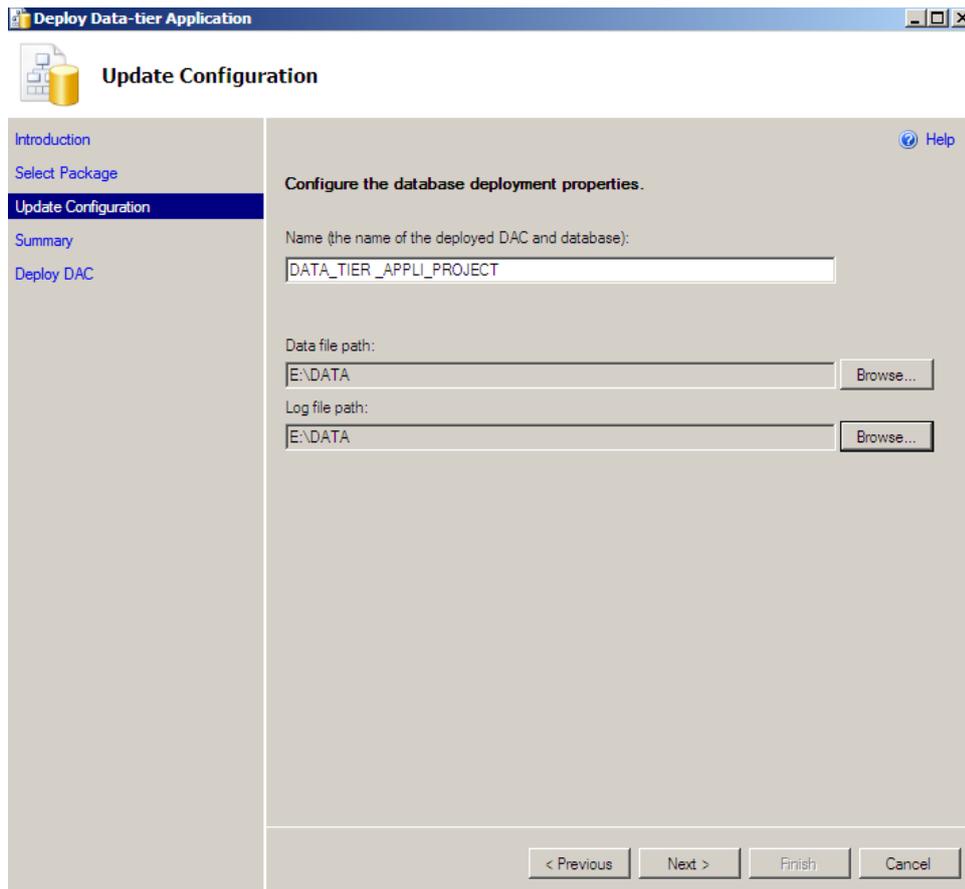
DEPLOIEMENT D'UN PACKAGE DAC

Le déploiement d'un package DAC par un administrateur de bases de données s'effectue grâce à l'assistant graphique prévu à cet effet dans SQL Server Management Studio.





Il faut choisir le package DAC à déployer. Une fois que celui-ci est sélectionné les informations le concernant (nom de l'application et la version) apparaissent dans la section *DAC details*.



La fenêtre suivante permet de configurer les propriétés de la base de données associée. Il est possible ici de définir les répertoires d'installation de la base de données de l'application DAC.



Summary

- Introduction
- Select Package
- Update Configuration
- Summary**
- Deploy DAC

Help

Summary

The following settings will be used to deploy your DAC.

```
[-] DAC properties
    ... Deploy from file: C:\Users\Administrateur\Documents\Visual Studio 2010\Projects\DATA_T
    ... Application name: DATA_TIER_APPLI_PROJECT
    ... Version: 1.0.0.0
    ... Description: None
[-] Database settings
```

< Previous Next > Finish Cancel

Deploy Data-tier Application

Deploy DAC

Introduction
Select Package
Update Configuration
Summary
Deploy DAC

[Help](#)

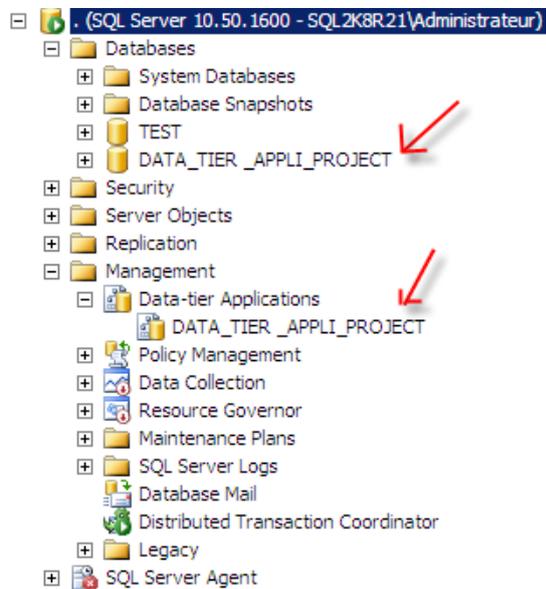
Deploying the DAC

Action	Result
✓ Preparing DAC metadata in the SQL Server instance 'SQL2K8R21'	Success
✓ Preparing deployment script	Success
✓ Creating database 'DATA_TIER_APPLI_PROJECT_c810cd12-02f4-4...	Success
✓ Creating schema objects in database 'DATA_TIER_APPLI_PROJECT...	Success
✓ Renaming database 'DATA_TIER_APPLI_PROJECT_c810cd12-02f4-4...	Success
✓ Registering the DAC in DAC metadata.	Success

[Save Report](#)

[< Previous](#) [Next >](#) [Finish](#) [Cancel](#)

La base de données nommée DATA_TIER_APPLI_PROJECT a bien été créé ainsi que l'application qui sont visibles respectivement sous le nœud *Databases* et *Management > Data-tier Applications*

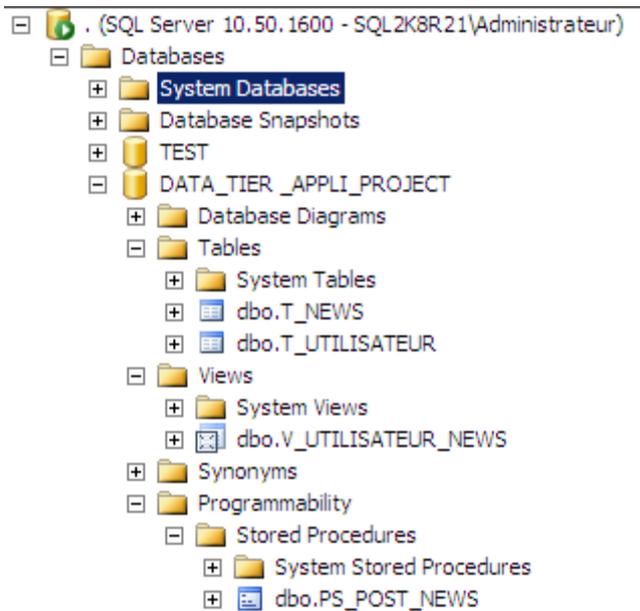


La définition de l'application est également stockée dans la base msdb. La table système *sysdac_instances_internal* permet de connaître le nom des applications installées.

```
USE msdb;  
GO  
SELECT *  
FROM dbo.sysdac_instances_internal
```

instance_id	instance_name	type_name	type_version	des
C81CCD12-02F4-4E9E-9DF5-AC8459E0A627	DATA_TIER_APPLI_PROJECT	DATA_TIER_APPLI_PROJECT	1.0.0.0	NL

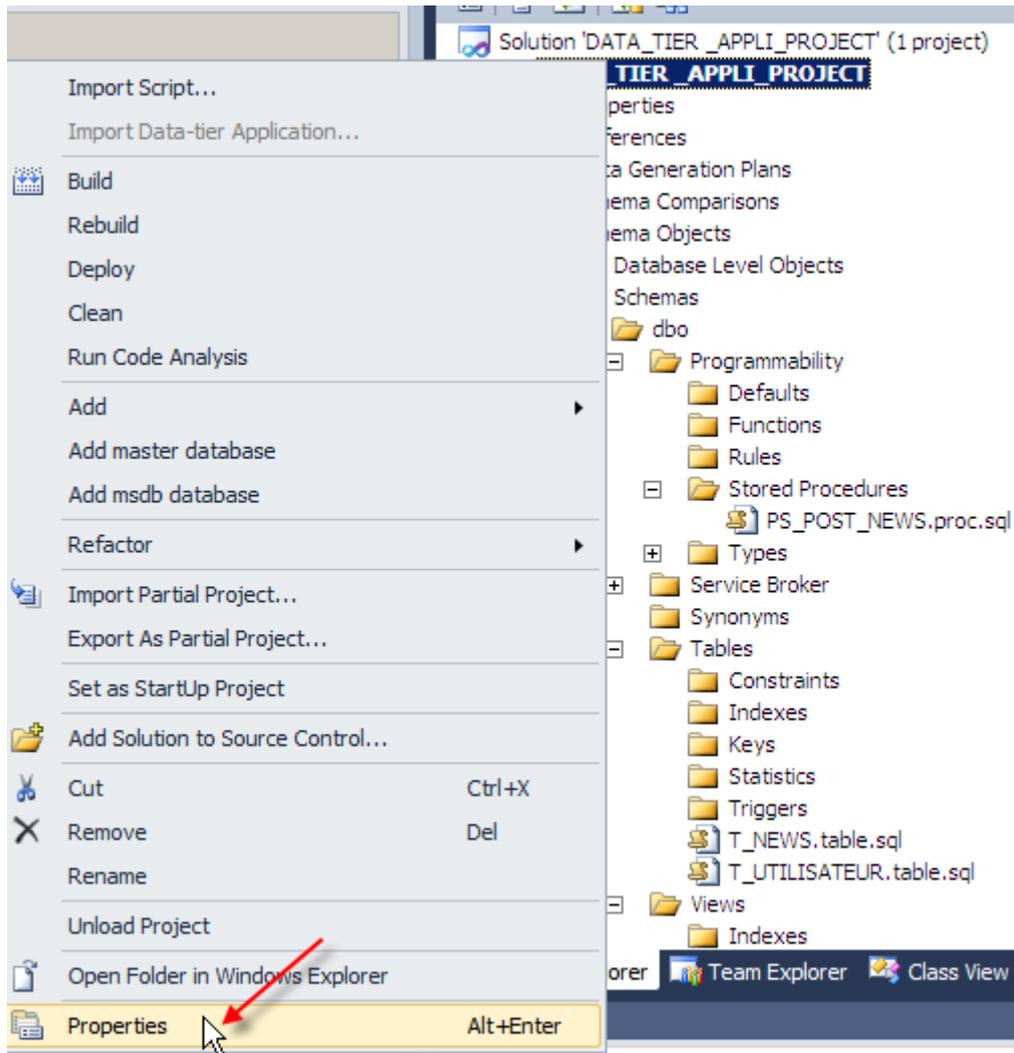
On peut retrouver les objets créés dans notre projet dans la base de données DATA_TIER_APPLI_PROJECT :



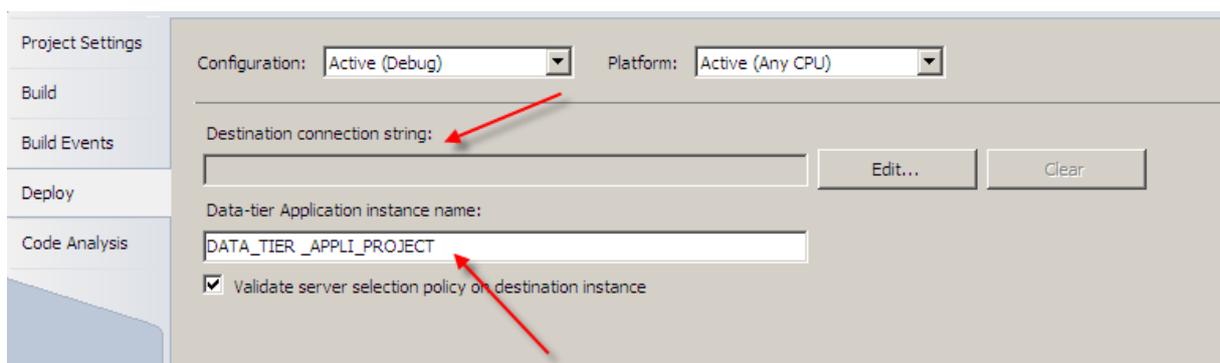
DEPLOIEMENT DU PACKAGE DAC DEPUIS VISUAL STUDIO 2010

Le déploiement d'une application DAC peut également s'effectuer depuis Visual Studio 2010 à condition d'avoir les droits minimums requis d'une part (à savoir db_creator) et d'avoir configuré les paramètres de déploiement liés au projet d'autre part.

Pour configurer les paramètres du projet il faut se rendre dans les propriétés de celui-ci (clic droit sur le projet et menu *propriétés*).



La fenêtre des propriétés s'ouvre dans la partie centrale :



... avec deux paramètres importants à configurer :

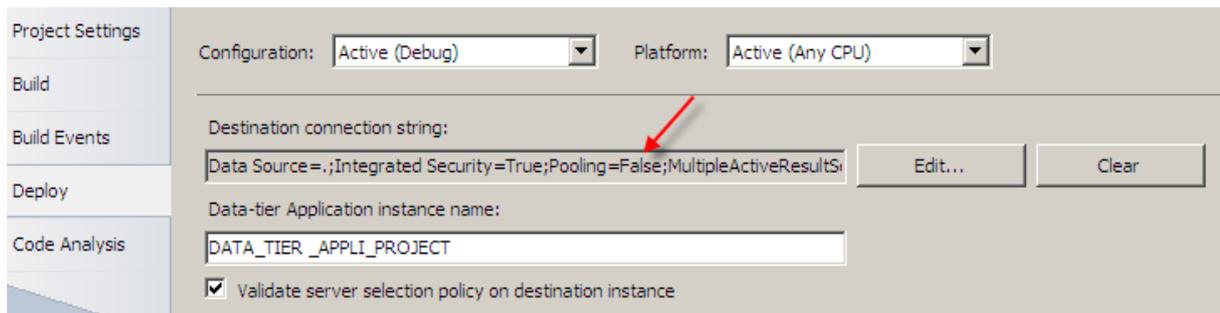
- La chaîne de connexion permettant de se connecter au serveur cible
- Le nom de l'application à déployer

Un 3^{ème} paramètre facultatif concerne la vérification des règles de déploiement avant le déploiement.

Le paramétrage de la chaîne de connexion s'effectue en cliquant sur le bouton « Edit ». Une fenêtre de connexion s'ouvre alors :



Le nom du serveur cible doit être renseigné ainsi que les informations de connexion.



Avant de déployer à nouveau l'application et de la mettre à jour nous ajouterons une table T_FORUM et modifierons la définition de la table T_NEWS.

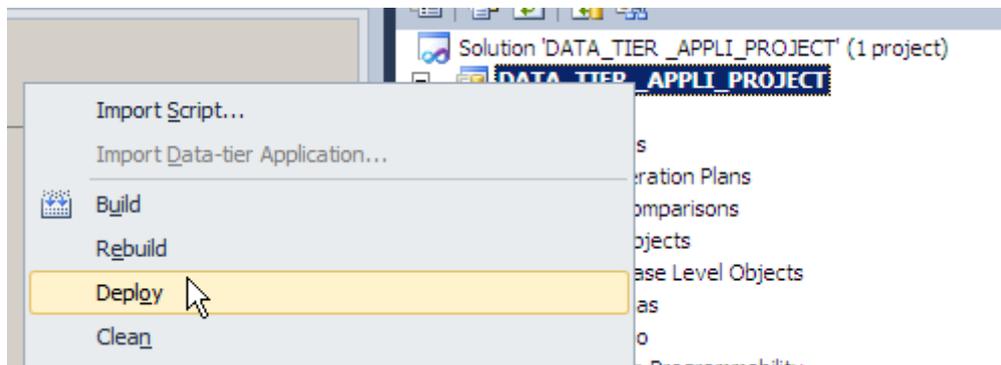
Le script de création de la table T_FORUM :

```
CREATE TABLE [dbo].[T_FORUM]
(
    id INT IDENTITY(1,1) PRIMARY KEY,
    nom VARCHAR(50) NOT NULL
)
```

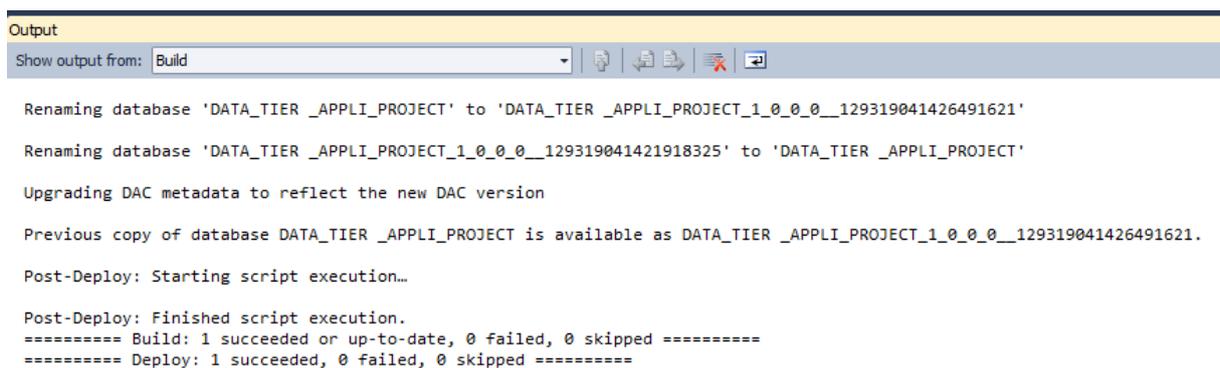
Le script de création de la table T_NEWS :

```
CREATE TABLE [dbo].[T_NEWS]
(
    id INT IDENTITY PRIMARY KEY,
    id_parent INT NULL,
    usr_id INT NOT NULL,
    forum_id INT NOT NULL,
    titre VARCHAR(20) NOT NULL,
    commentaire VARCHAR(max) NOT NULL,
    CONSTRAINT FK_T_NEWS_id_parent FOREIGN KEY (id_parent) REFERENCES T_NEWS(id),
    CONSTRAINT FK_T_UTILISATEUR_id FOREIGN KEY (usr_id) REFERENCES T_UTILISATEUR(id),
    CONSTRAINT FK_T_FORUM_id FOREIGN KEY (forum_id) REFERENCES T_FORUM(id)
);
```

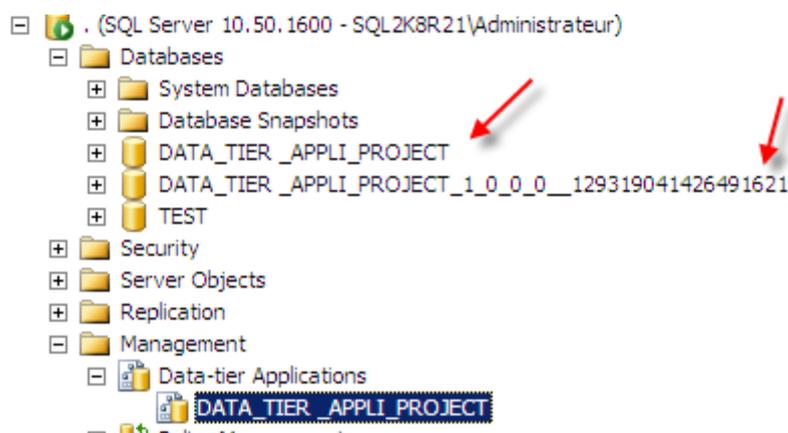
Pour déployer une application, il suffit d'effectuer un clic droit sur le projet et de choisir le menu « Deploy »



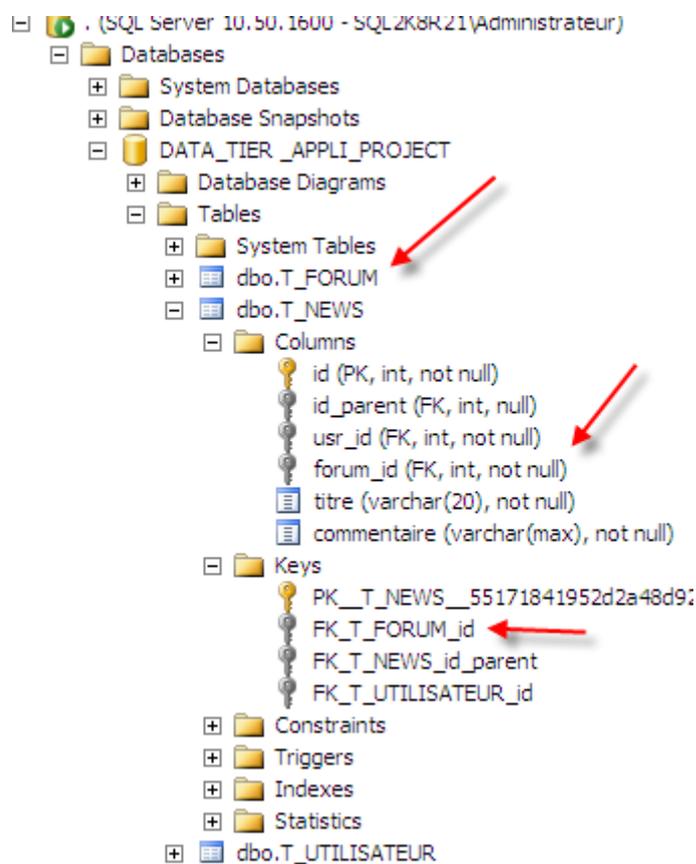
La fenêtre de sortie affiche la progression du déploiement et les éventuels messages d'erreurs rencontrés.



Si l'on se rend dans SQL Server Management Studio, on peut remarquer qu'il existe maintenant deux bases de données ou plutôt deux versions de bases de données :



Les objets de la base ont bien été modifiés :



En interrogeant la table *sysdac_history_internal* dans la base de données *msdb* on peut visualiser les différentes étapes exécutées par SQL Server pour effectuer un déploiement de l'application DAC.

```

] SELECT
    date_modified,
    sequence_id,
    action_type_name,
    dac_object_type_name,
    dac_object_name_pretran AS dac_object_before,
    dac_object_name_posttran AS dac_object_after,
    sqlscript
FROM msdb.dbo.sysdac_history_internal
-WHERE date_modified > '20101018'

```

... qui donne le résultat suivant :

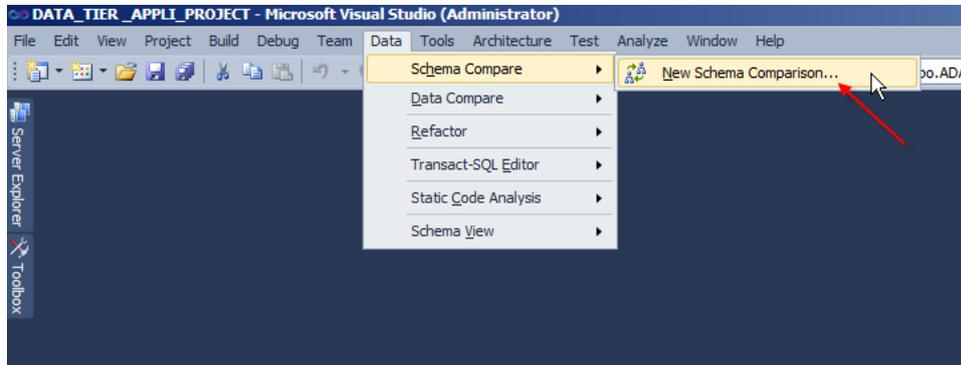
	date_modified	sequ...	action_type_name	dac_object...	dac_object_before	dac_object_after	
1	2010-10-18 21:36:00.550	0	upgrade	dacpac	DATA_TIER_APPLI_PROJECT	DATA_TIER_APPLI_PROJECT	U
2	2010-10-18 21:35:54.737	1	create	database	DATA_TIER_APPLI_PROJECT_1_0_0_0__129319041421918325	DATA_TIER_APPLI_PROJECT_1_0_0_0__12931904142191...	C
3	2010-10-18 21:35:54.817	2	create objects	database	DATA_TIER_APPLI_PROJECT_1_0_0_0__129319041421918325	DATA_TIER_APPLI_PROJECT_1_0_0_0__12931904142191...	S
4	2010-10-18 21:35:54.937	3	register	dacpac	DATA_TIER_APPLI_PROJECT_1_0_0_0__129319041421918325	DATA_TIER_APPLI_PROJECT_1_0_0_0__12931904142191...	E
5	2010-10-18 21:35:55.143	4	set readonly	database	DATA_TIER_APPLI_PROJECT	DATA_TIER_APPLI_PROJECT	A
6	2010-10-18 21:35:55.153	5	disconnect users	database	DATA_TIER_APPLI_PROJECT	DATA_TIER_APPLI_PROJECT	A
7	2010-10-18 21:36:00.040	6	disable constraints	database	DATA_TIER_APPLI_PROJECT	DATA_TIER_APPLI_PROJECT_1_0_0_0__12931904142191...	A
8	2010-10-18 21:36:00.100	7	move data	database	DATA_TIER_APPLI_PROJECT	DATA_TIER_APPLI_PROJECT_1_0_0_0__12931904142191...	S
9	2010-10-18 21:36:00.113	8	enable constraints	database	DATA_TIER_APPLI_PROJECT	DATA_TIER_APPLI_PROJECT_1_0_0_0__12931904142191...	A
10	2010-10-18 21:36:00.213	9	set readwrite	database	DATA_TIER_APPLI_PROJECT	DATA_TIER_APPLI_PROJECT	A
11	2010-10-18 21:36:00.433	10	rename	database	DATA_TIER_APPLI_PROJECT	DATA_TIER_APPLI_PROJECT_1_0_0_0__12931904142649...	E
12	2010-10-18 21:36:00.473	11	rename	database	DATA_TIER_APPLI_PROJECT_1_0_0_0__129319041421918325	DATA_TIER_APPLI_PROJECT	E
13	2010-10-18 21:36:00.547	12	update registration	dacpac	DATA_TIER_APPLI_PROJECT_1_0_0_0__129319041421918325	DATA_TIER_APPLI_PROJECT	E

Les étapes exécutées sont les suivantes :

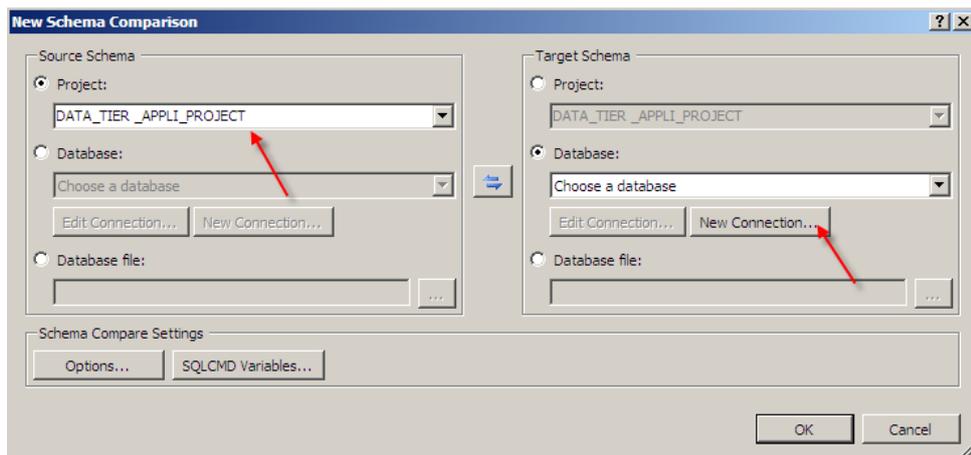
- Création d'une base de données supplémentaire
- Création des objets à l'image de l'application DAC initiale
- Enregistrement de la nouvelle application DAC sur la nouvelle base créée
- Mise en lecture seule de la base de données initiale
- Déconnexion des utilisateurs de la base de données initiale
- Désactivation des contraintes présentes dans la nouvelle base de données
- Transfert des données depuis la base de données initiale vers la nouvelle base de données
- Réactivation des contraintes présentes dans la nouvelle base de données
- Réaction du mode lecture / écriture de la base de données initiale
- Renommage de la base de données initiale
- Renommage de la nouvelle base de données avec le nom d'origine de l'application DAC
- Mise à jour de la nouvelle application DAC

COMPARATEUR DE SCHEMA AVEC VS2010

Le comparateur de schéma est une fonctionnalité intéressante à utiliser avec une application DAC. Un développeur peut rapidement voir les différences qu'il existe entre une version de production et une version en développement.



On peut comparer notre application DAC (Project) et la base de données en production



Connection Properties

Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.

Data source: Microsoft SQL Server (SqlClient) Change...

Server name: localhost Refresh

Log on to the server

Use Windows Authentication

Use SQL Server Authentication

User name:

Password:

Save my password

Connect to a database

Select or enter a database name:

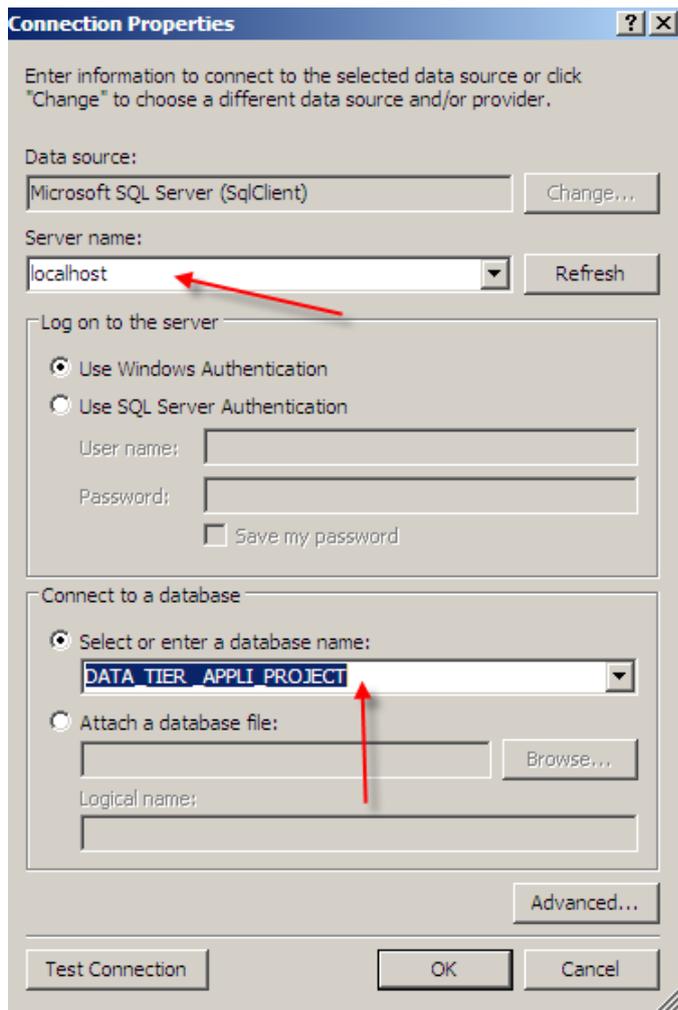
DATA_TIER_APPLI_PROJECT

Attach a database file: Browse...

Logical name:

Advanced...

Test Connection OK Cancel



New Schema Comparison

Source Schema

Project: DATA_TIER_APPLI_PROJECT

Database: Choose a database Edit Connection... New Connection...

Database file: ...

Target Schema

Project: DATA_TIER_APPLI_PROJECT

Database: [SQL2K8R21.DATA_TIER_APPLI_PROJECT] Edit Connection... New Connection...

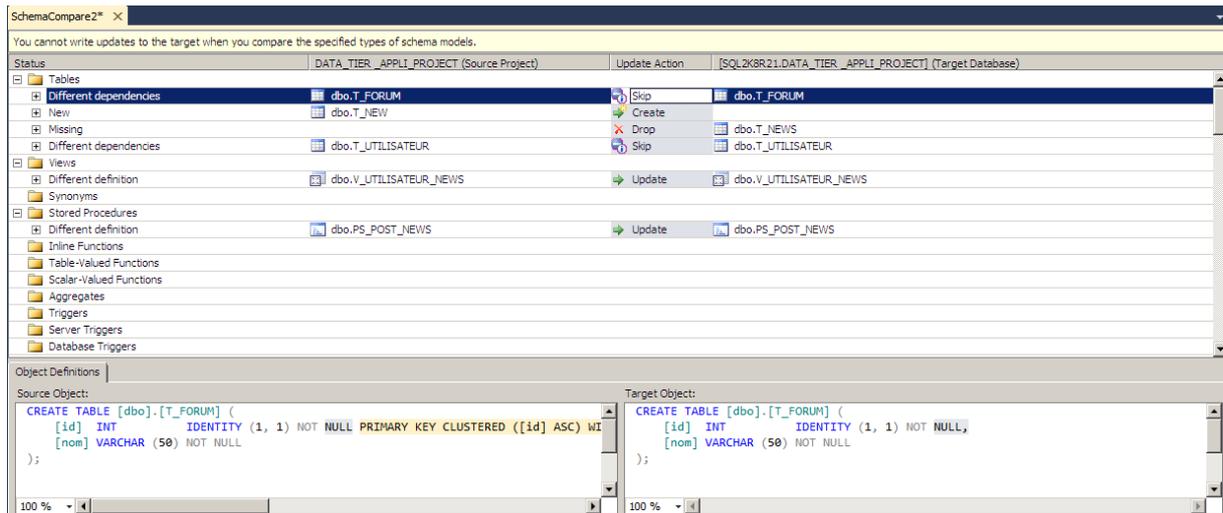
Database file: ...

Schema Compare Settings

Options... SQLCMD Variables...

OK Cancel

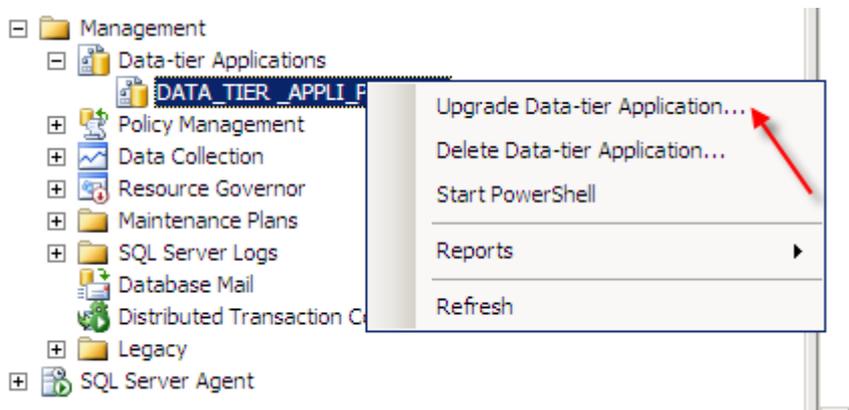


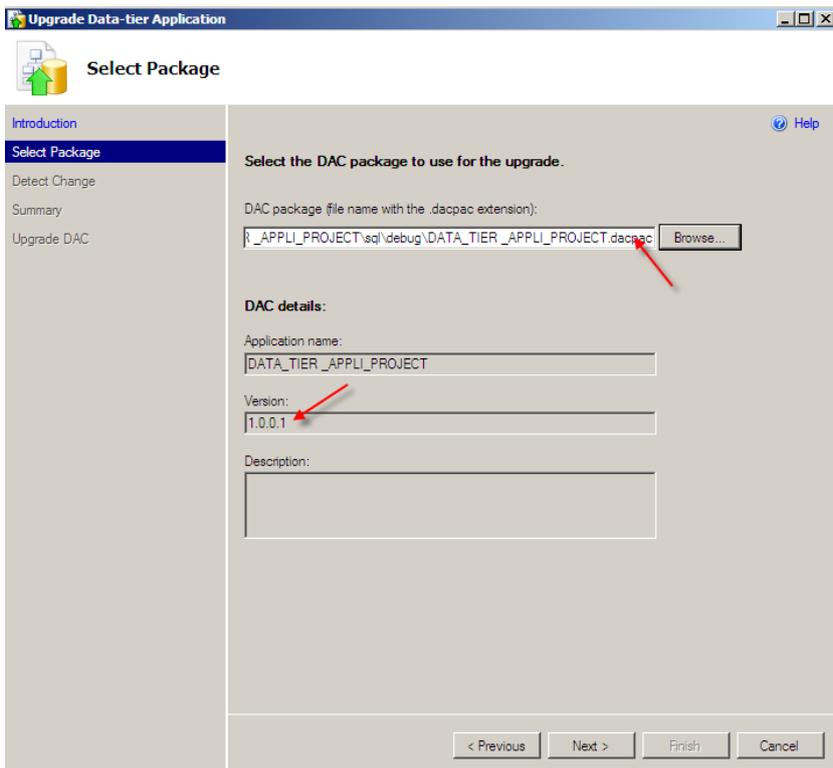
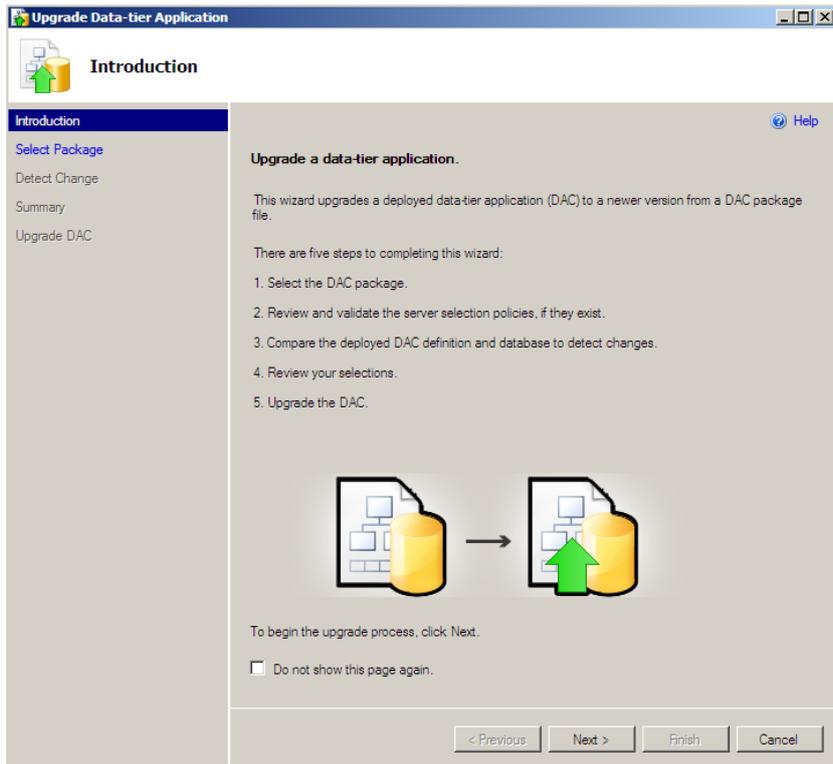


On peut voir par exemple que la table *dbo.T_NEW* doit être créée et la table *dbo.T_NEWS* doit être supprimée ou encore que la vue *dbo.V_UTILISATEUR_NEWS* doit être mise à jour car elle référence la table *dbo.T_NEW*.

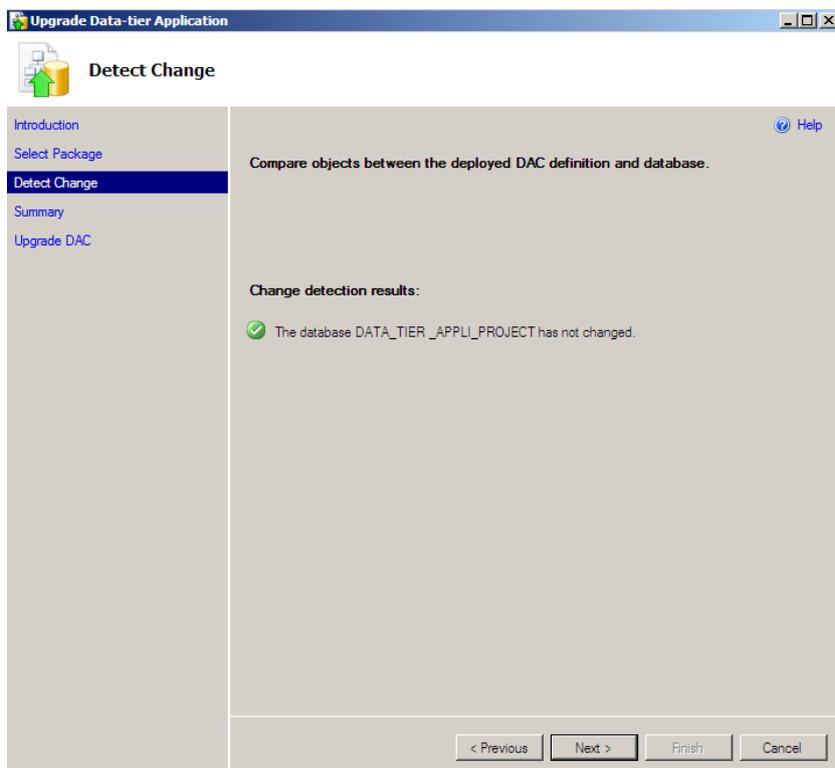
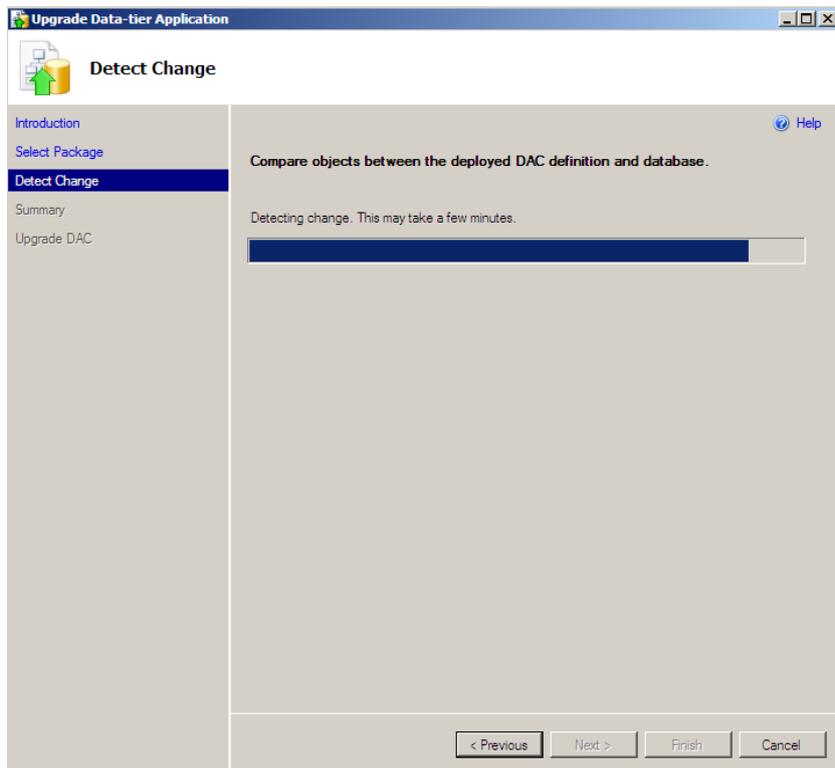
MISE A JOUR D'UNE APPLICATION DAC AVEC SSMS

Nous avons vu un peu plus haut la mise à jour d'une application DAC depuis VS2010. Dans cette section nous allons voir cette même mise à jour depuis SSMS. Après avoir généré le package DAC de l'application, il faut lancer l'assistant de mise à jour prévue à cet effet.

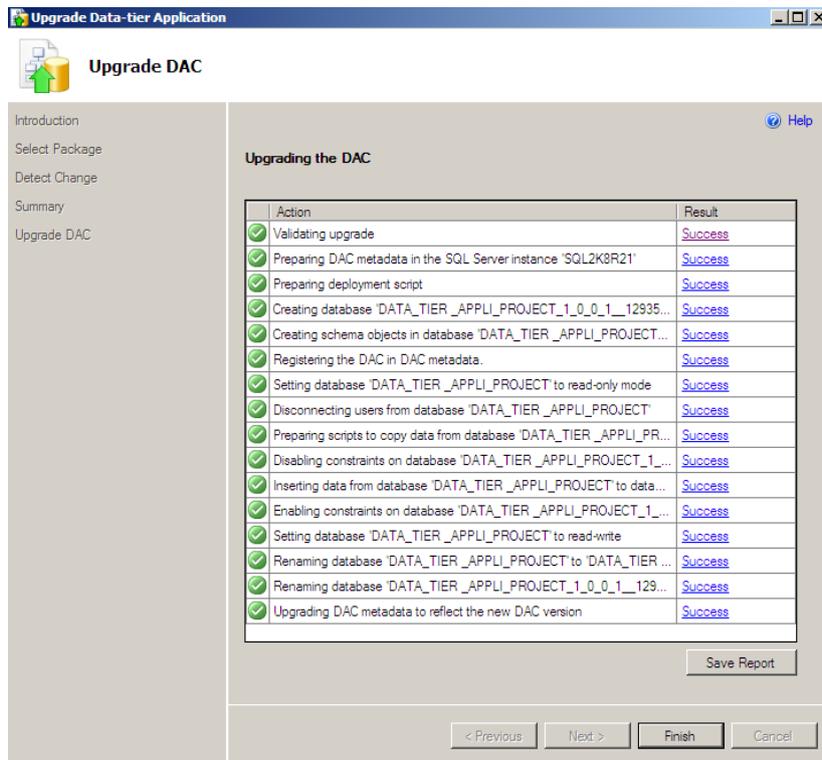




Une détection et une comparaison des changements est effectuée.



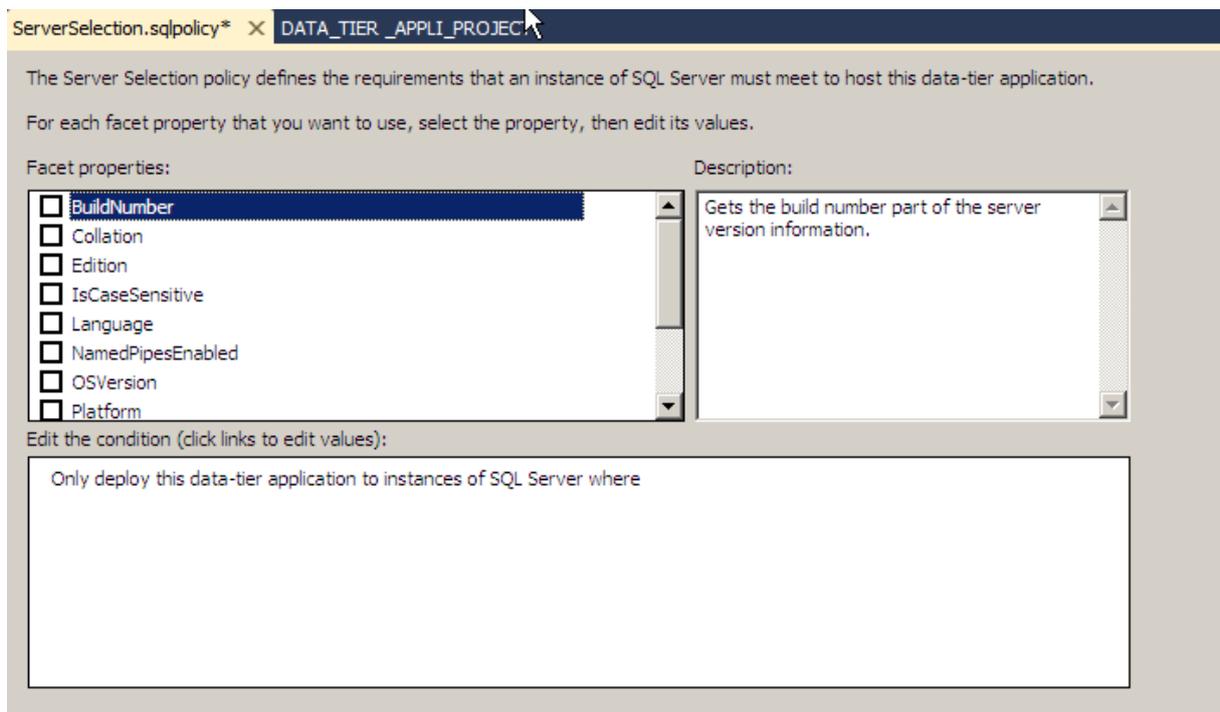
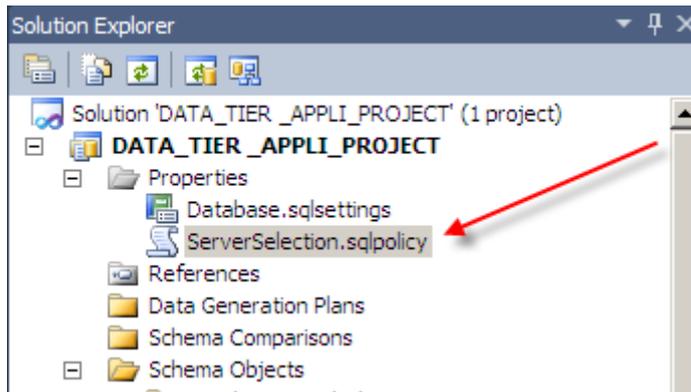
SQL Server procède ensuite à la vérification des objets du package afin de savoir si ceux-ci sont supportés par DAC. Cette étape passée, la mise à jour de l'application peut commencer.



REGLES DE GESTION DE DEPLOIEMENT D'UNE APPLICATION DAC

Il est possible de créer des règles de gestion de déploiement des packages DAC depuis Visual Studio 2010, ce qui permet de concevoir une véritable politique d'entreprise de déploiement sur les instances SQL Server. Par exemple, il est possible contrôler la mise à jour de versions des applications sur les serveurs et de ne pas déployer accidentellement une version ancienne en production. De la même manière il est possible de contrôler sur quels éditions de SQL Server ou sur quelle version du système d'exploitation votre package peut être déployé. La sélection des règles de gestion de déploiement dans Visual Studio 2010 permet de réaliser un certain nombre de scénarios de contrôle pour le déploiement d'une application DAC.

Dans notre cas nous ne voulons pas que l'application DAC DATA_TIER_APPLI_PROJECT puisse être déployée sur une instance SQL Server Enterprise Edition. La règle de déploiement se traduit de la façon suivante sur Visual Studio 2010 (nœud *Properties* > *ServerSelection.sqlpolicy*).

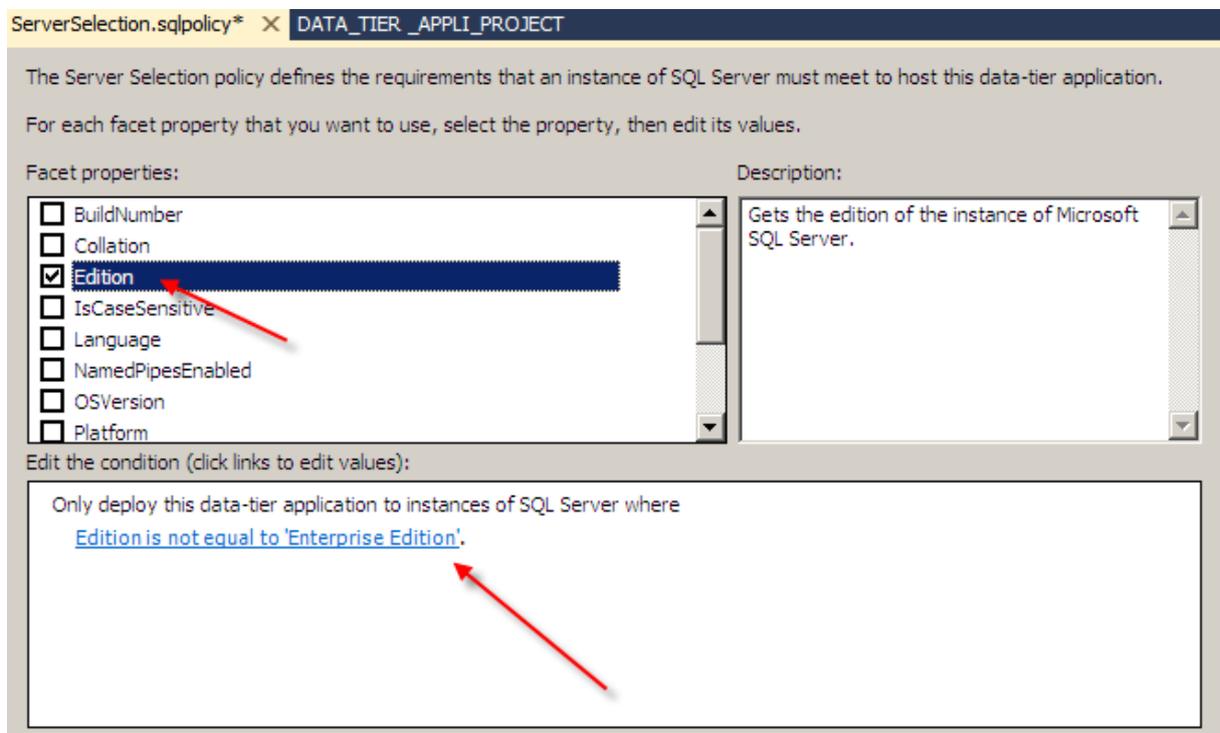
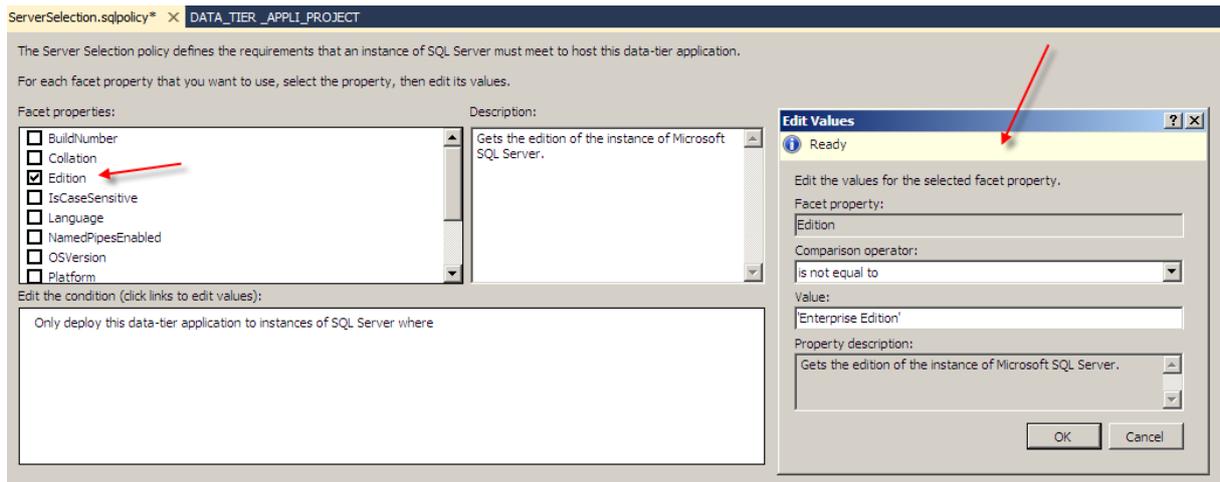


Une règle de déploiement est composée de facettes et de conditions. Dans notre cas celle-ci sera paramétrée de la façon suivante :

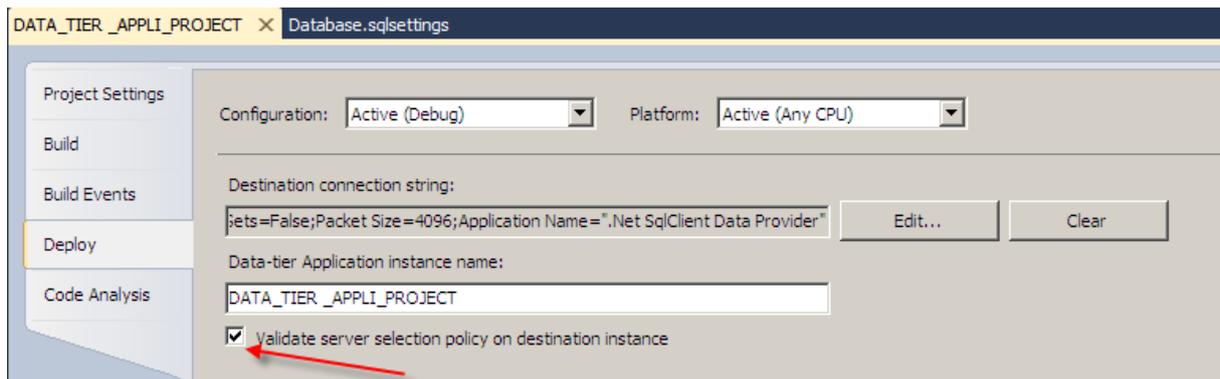
[Edition] (Facet)

[is not equal to] (Comparison operator)

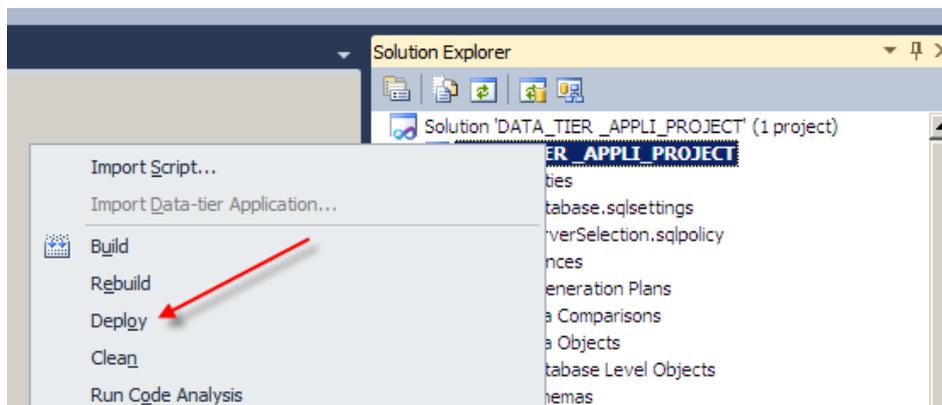
[Enterprise Edition] (Condition value)



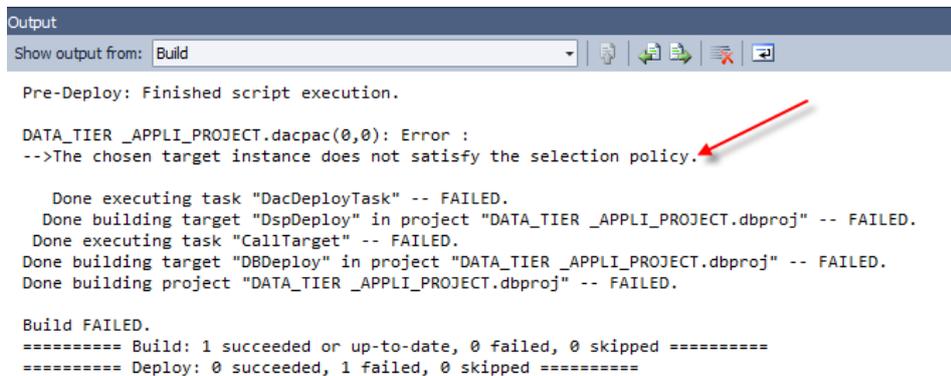
Une fois la fois la règle de sélection définie nous pouvons déployer le package DAC depuis Visual Studio 2010 dans un premier temps et en utilisant SQL Server Management Studio sur une instance SQL Server Enterprise Edition dans un deuxième temps. Il faut au préalable avoir activé l'option de vérification des règles de sélection pour le déploiement dans les propriétés du projet.



Déploiement via Visual Studio 2010



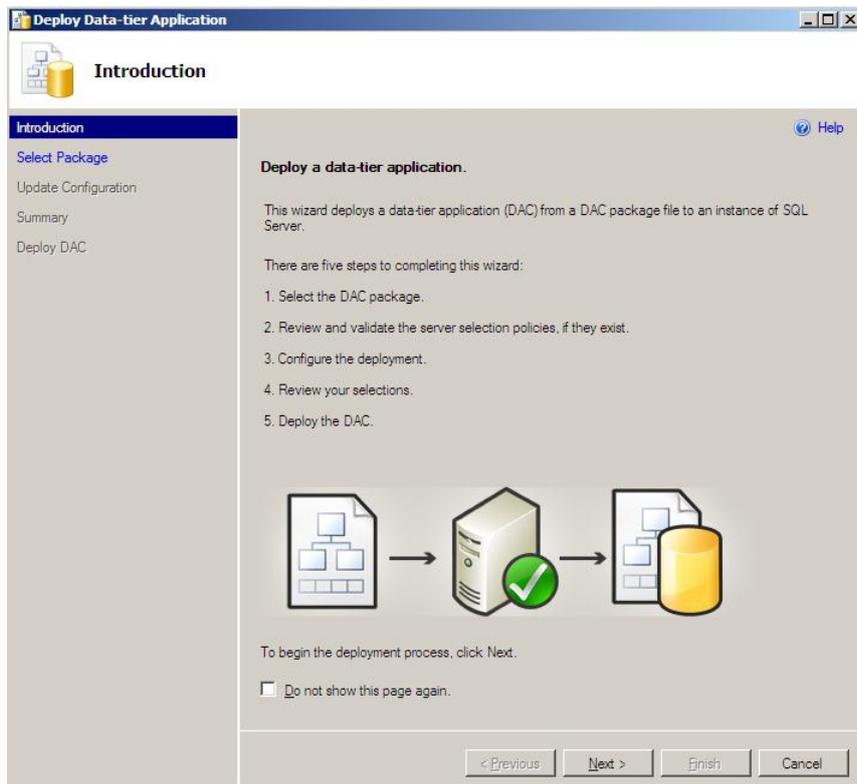
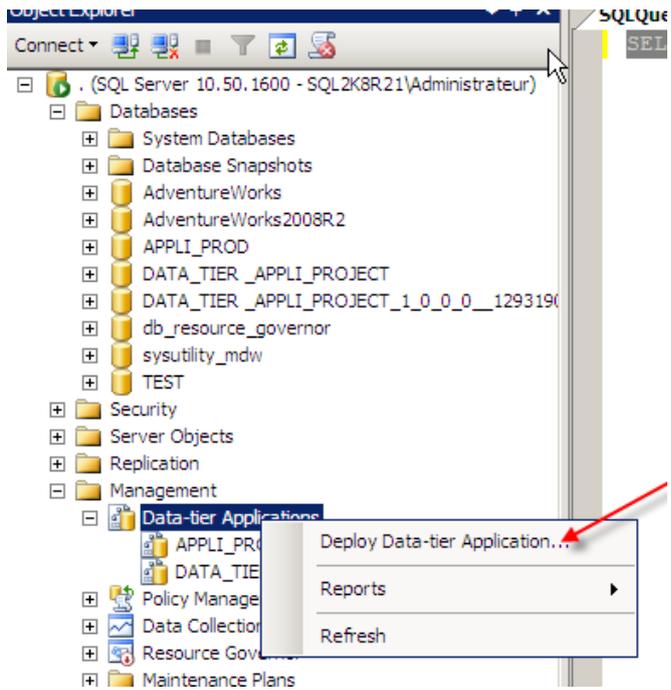
La fenêtre de sortie de déploiement est la suivante :



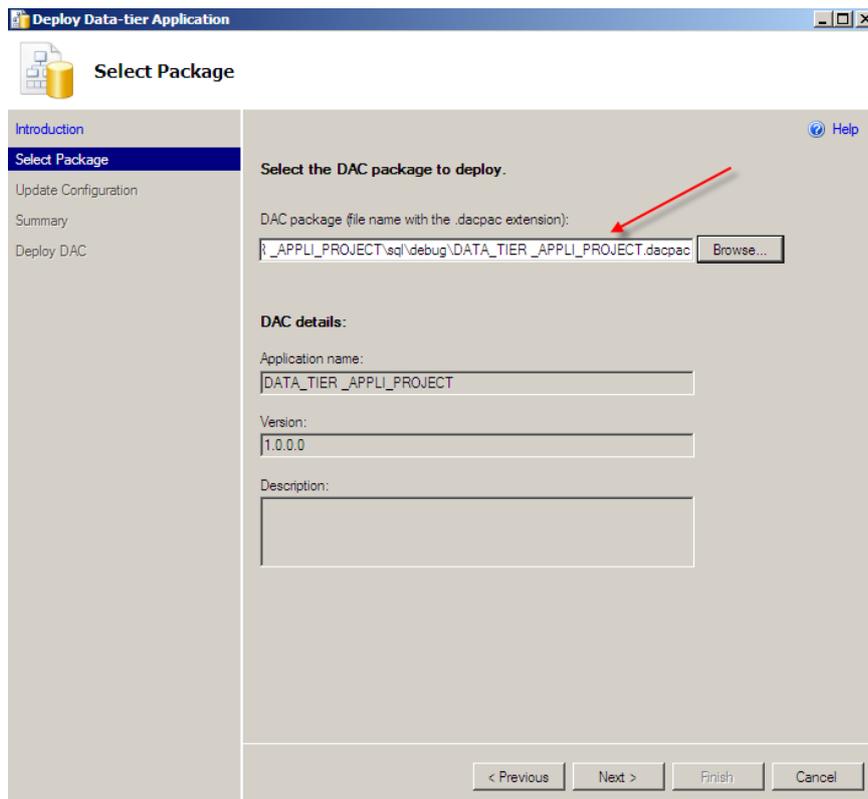
Le déploiement a échoué car l'instance SQL Server cible ne respecte pas la règle de déploiement paramétrée pour l'application DAC DATA_TIER_APPLI_PROJECT

Déploiement via SQL Server Management Studio

Dans ce scénario, il faut d'abord générer le package DAC. (cf. section génération package DAC). L'assistant DAC nous permet ensuite de déployer l'application DAC à l'aide du package généré.



Sélection du package



Une erreur est levée dans l'écran d'examen des règles de déploiement. A ce moment du processus de déploiement, l'administrateur de bases de données a la possibilité de passer outre les erreurs existantes et de forcer le déploiement (option *ignore policy violations*).



Review Policy

- Introduction
- Select Package
- Review Policy**
- Update Configuration
- Summary
- Deploy DAC

Help

Review the results of evaluating the server selection policy.

Evaluation results of policy conditions:

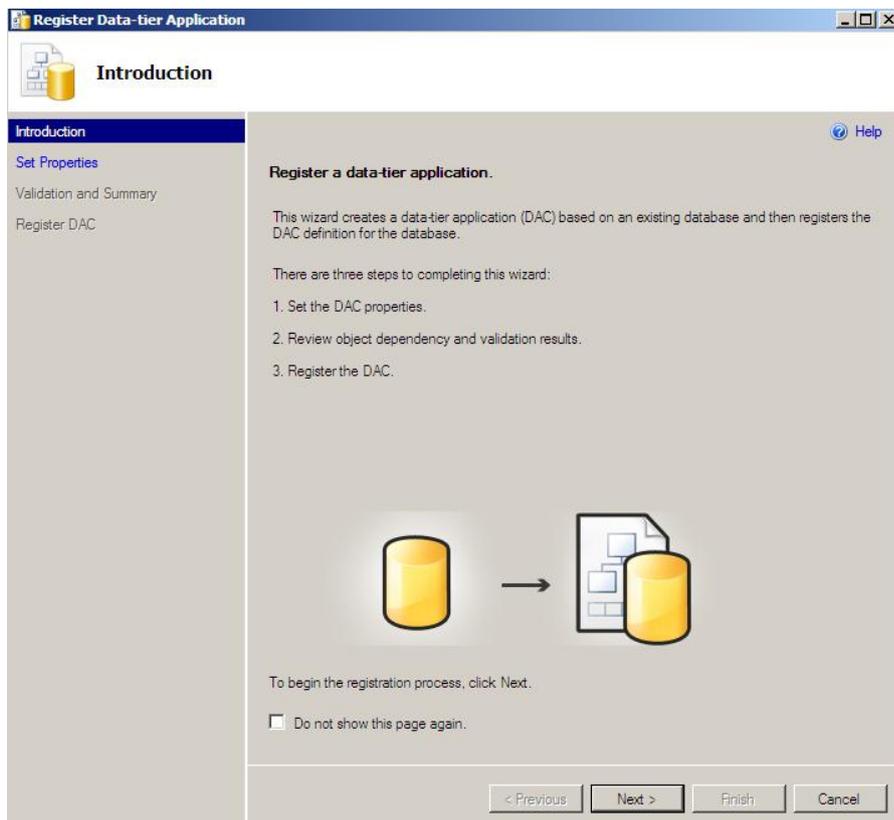
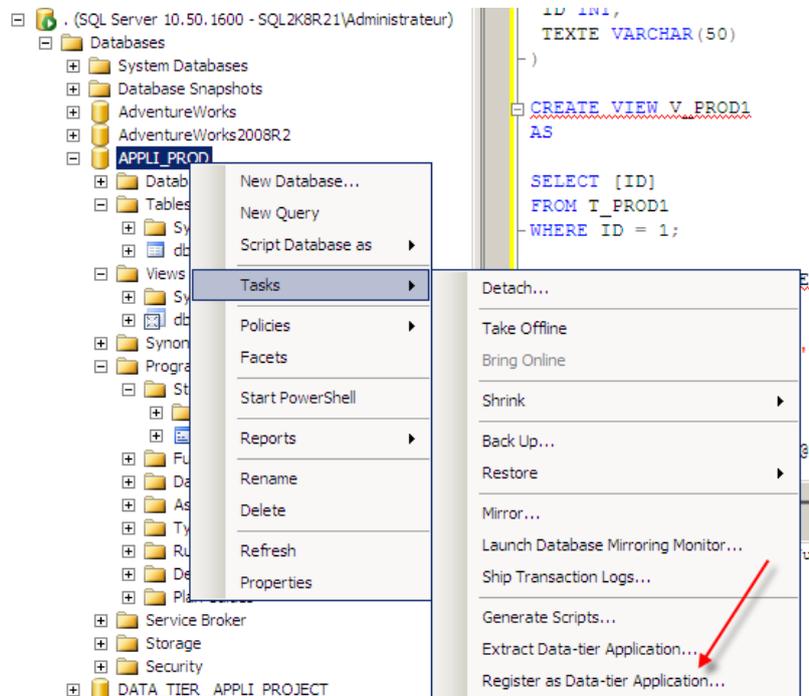
AndOr	Result	Field	Operator	Expected Value	Actual Value
		@Edition	NOT L...	'Enterprise Edition'	'Enterprise Edition'

Ignore policy violations.

< Previous Next > Finish Cancel

CONVERSION D'UNE BASE DE DONNEES EN APPLICATION DAC

Une base de données existante peut être convertie en une application DAC. Cette nouvelle application peut également être alors déployée sur d'autres instances SQL Server ou modifiée par les équipes de développement dans Visual Studio 2010.



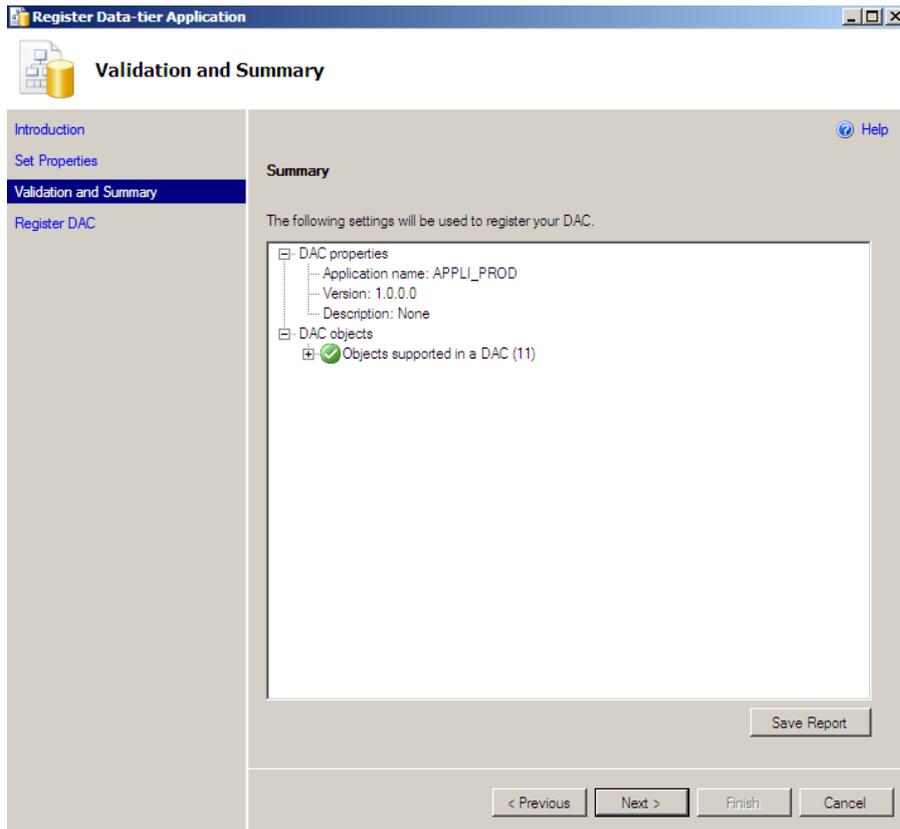
Il faut donner un nom pour la nouvelle application DAC ainsi qu'un numéro de version

The screenshot shows the 'Register Data-tier Application' wizard in SQL Server Enterprise Manager. The current step is 'Set Properties'. The form contains the following fields:

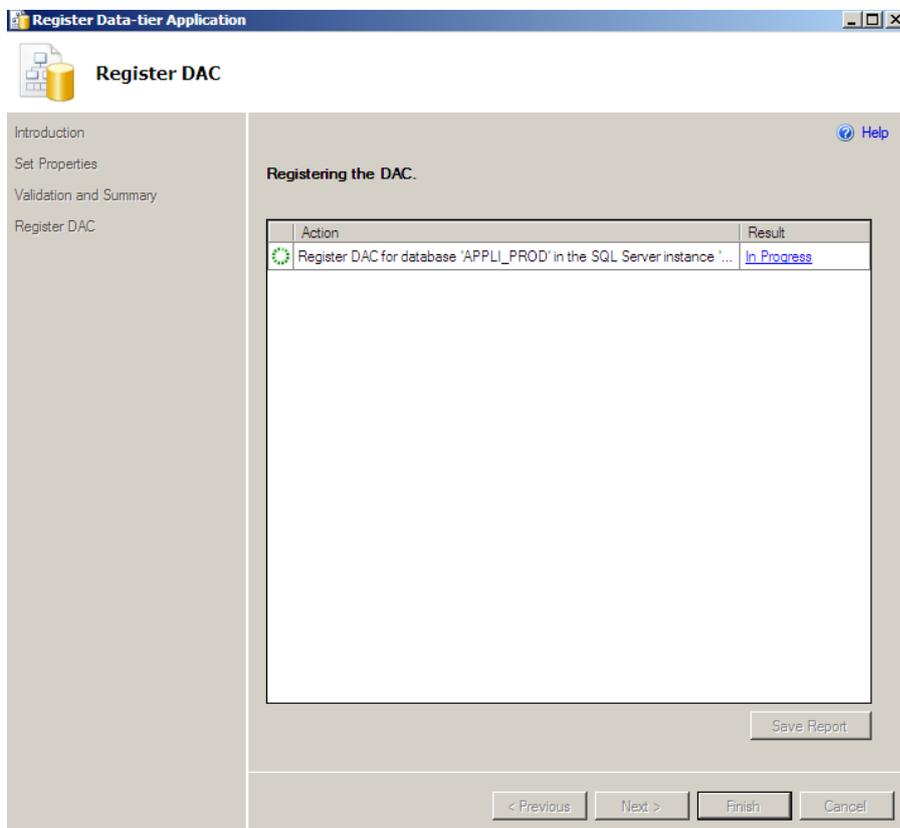
- DAC instance name: APPLI_PROD
- Application name: APPLI_PROD
- Version (use x.x.x.x where x is a number): 1.0.0.0
- Description: (empty)

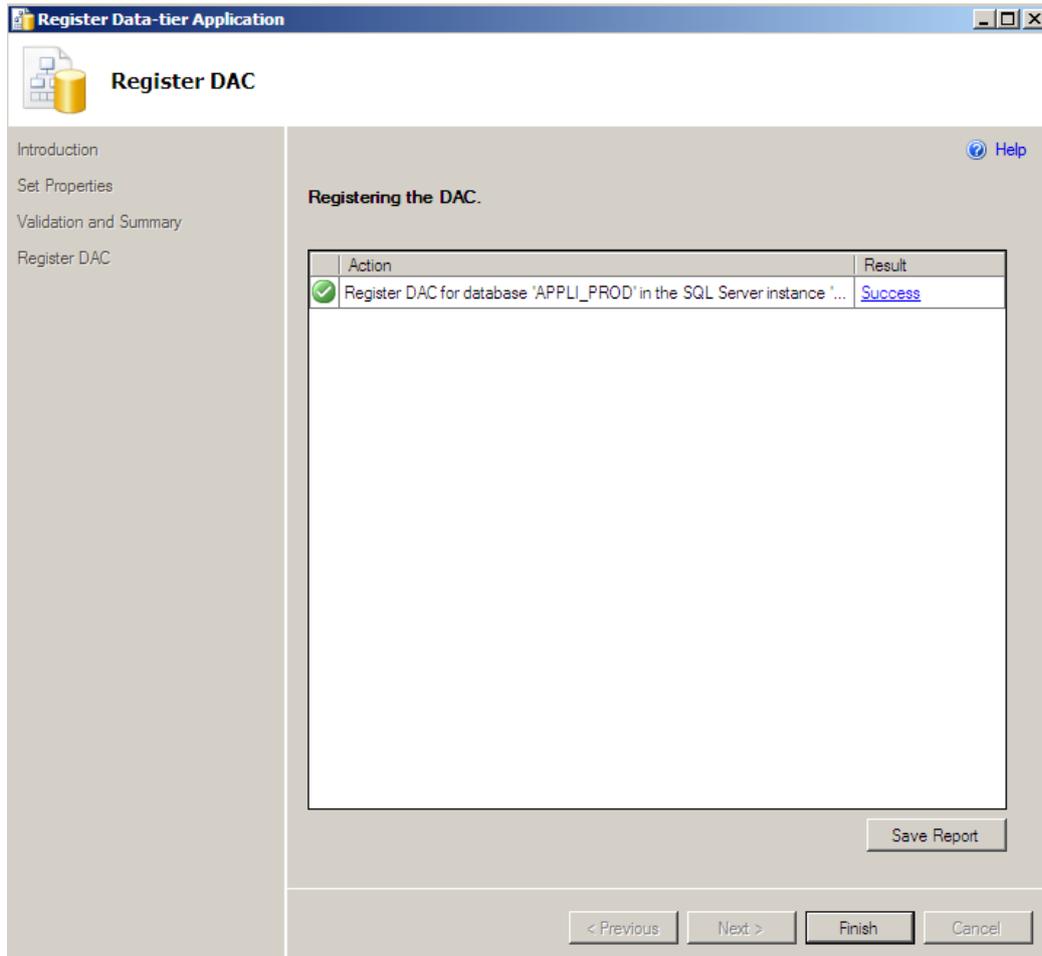
Red arrows point to the 'Application name' and 'Version' fields. At the bottom, there are buttons for '< Previous', 'Next >', 'Finish', and 'Cancel'. A 'Help' link is visible in the top right corner.

Une tâche est ensuite démarrée pour vérifier si les objets que contient la base de données APPLI_PROD sont supportés par DAC. Si une erreur survient pendant cette phase de vérification à cause d'un objet non supportée par DAC, la base de données ne peut pas être convertie. L'administrateur de bases de données est alors invité à corriger les erreurs rencontrées pendant le processus de conversion.

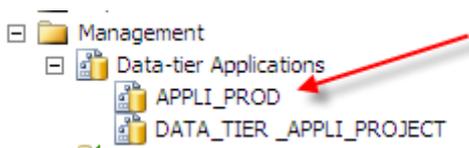


Dans notre cas, aucun problème n'a été rencontré. Tous les objets de la base APPLI_PROD sont supportés par DAC.

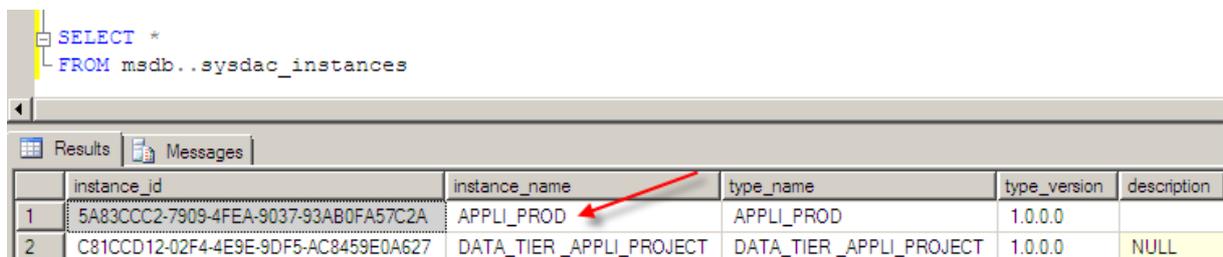




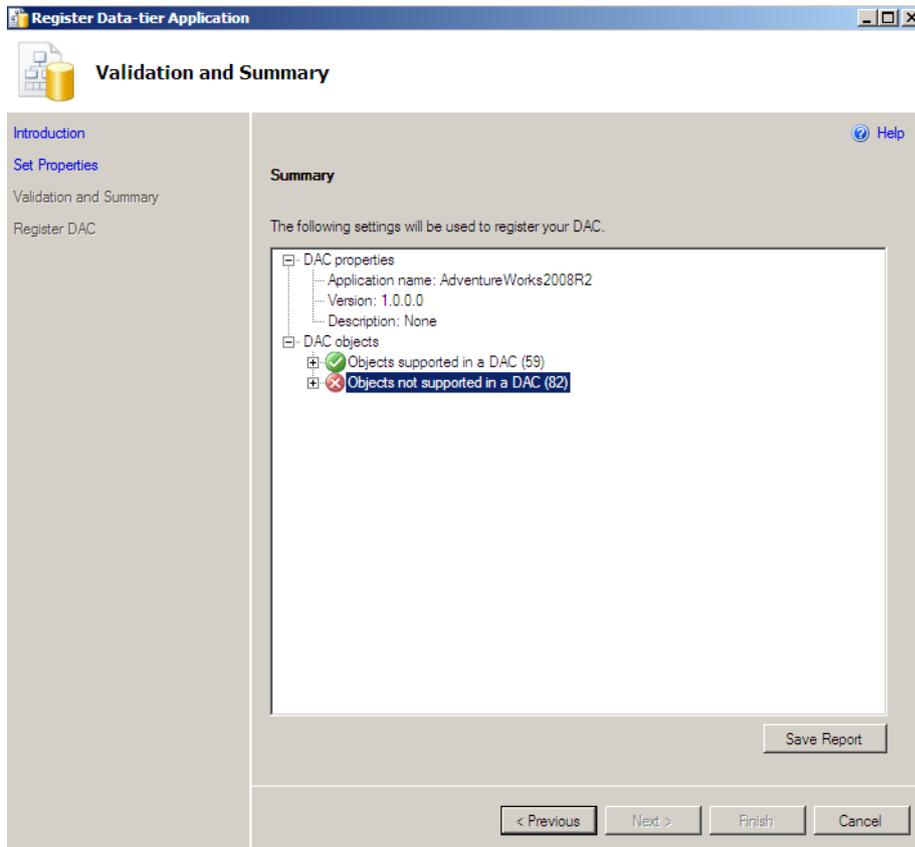
La base de données APPLI_PROD est maintenant inscrite en tant que qu'application DAC. On peut le vérifier via le nœud « Data-tier Applications »



... ou la table système *sysdac_instances* dan la base de données *msdb* :



Par exemple la base de données exemple AdventureWorks2008R2 ne peut pas être convertie en application DAC :



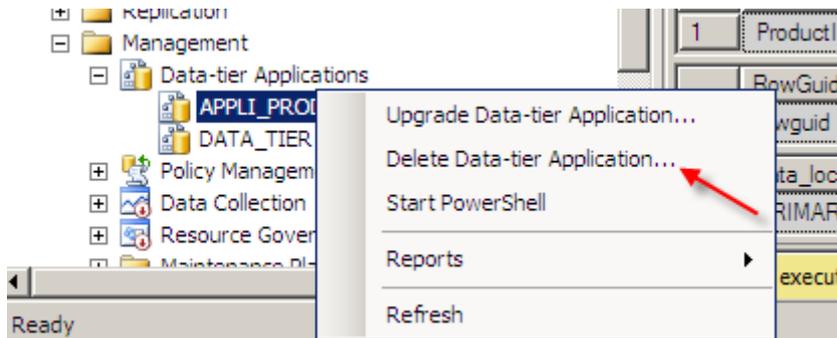
Voici quelques exemples d'objets présents dans la base de données AdventureWorks2008R2 et qui posent problème :

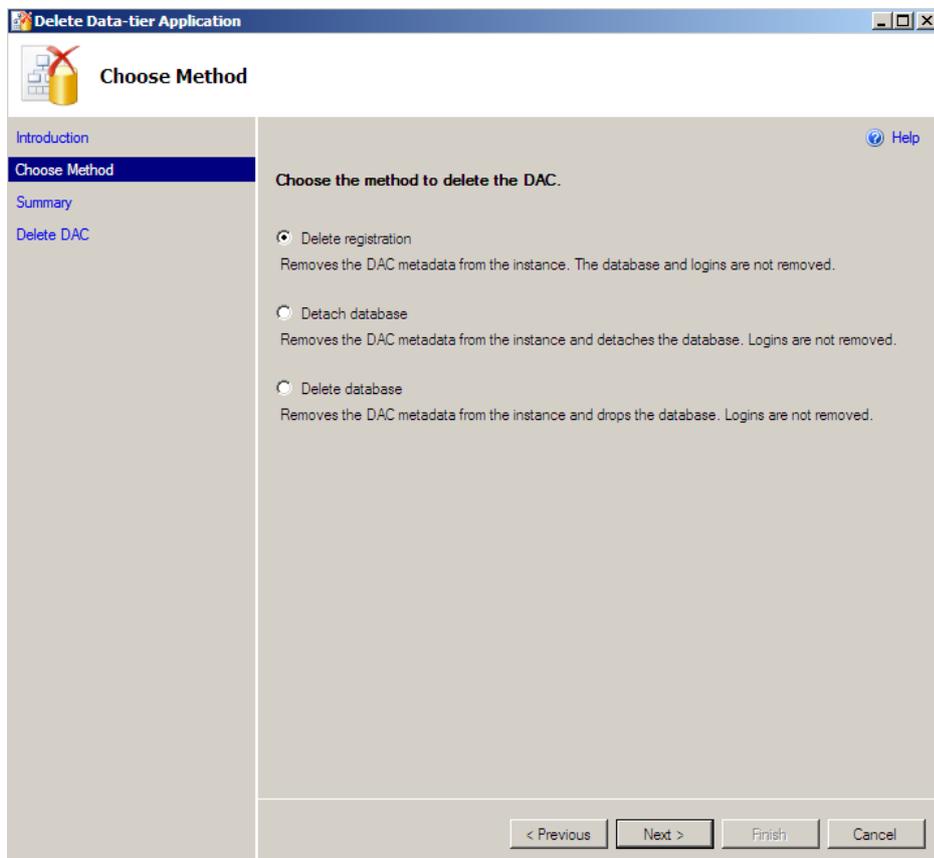
- Index FULLTEXT
- Type de données non supportés :
 - o Hierarchy,
 - o Geometry,
 - o XML,
 - o XML schemas collection
 - o FILESTREAM
- Triggers DDL
- Propriétés étendues

SUPPRESSION D'UNE APPLICATION DAC

Un administrateur de bases de données peut être amené à supprimer une application DAC d'une instance SQL Server. Cette suppression peut s'effectuer à l'aide de l'assistant prévu à cet effet. Ce dernier propose 3 options de suppression :

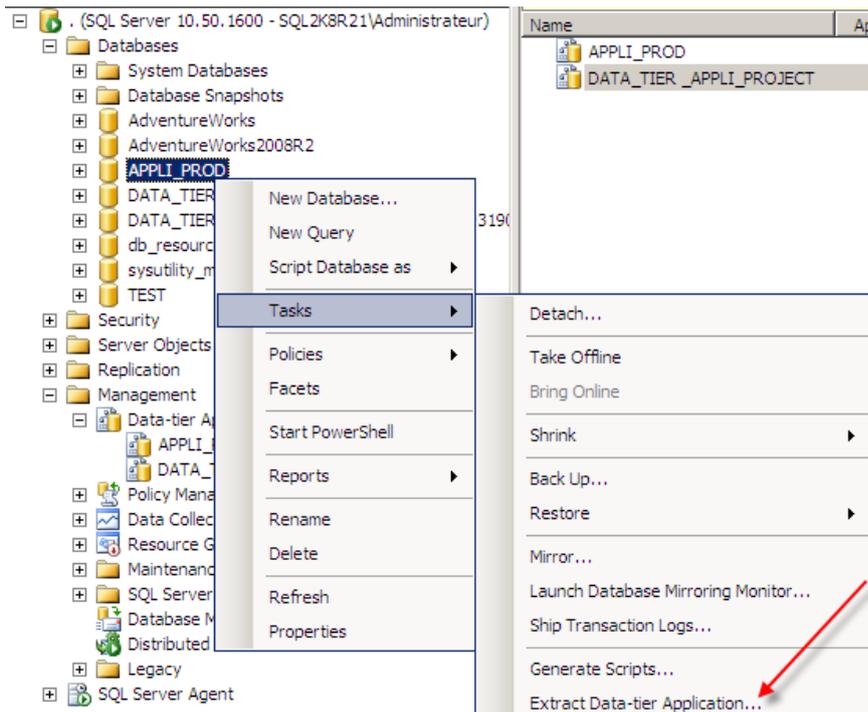
- Par désinscription de l'application DAC : Cette option permet de supprimer les métadonnées de l'application DAC. La base de données associée et les comptes de connexion sont préservés.
- Par détachement : Cette option supprime également les métadonnées associées à l'application DAC. A l'instar d'une opération de détachement d'une base de données classique, les fichiers de l'application DAC sont également détachés. Les comptes de connexion ne sont pas supprimés.
- Par suppression : Les métadonnées ainsi que la base de données associée sont supprimés. Les comptes de connexion sont préservés.
-





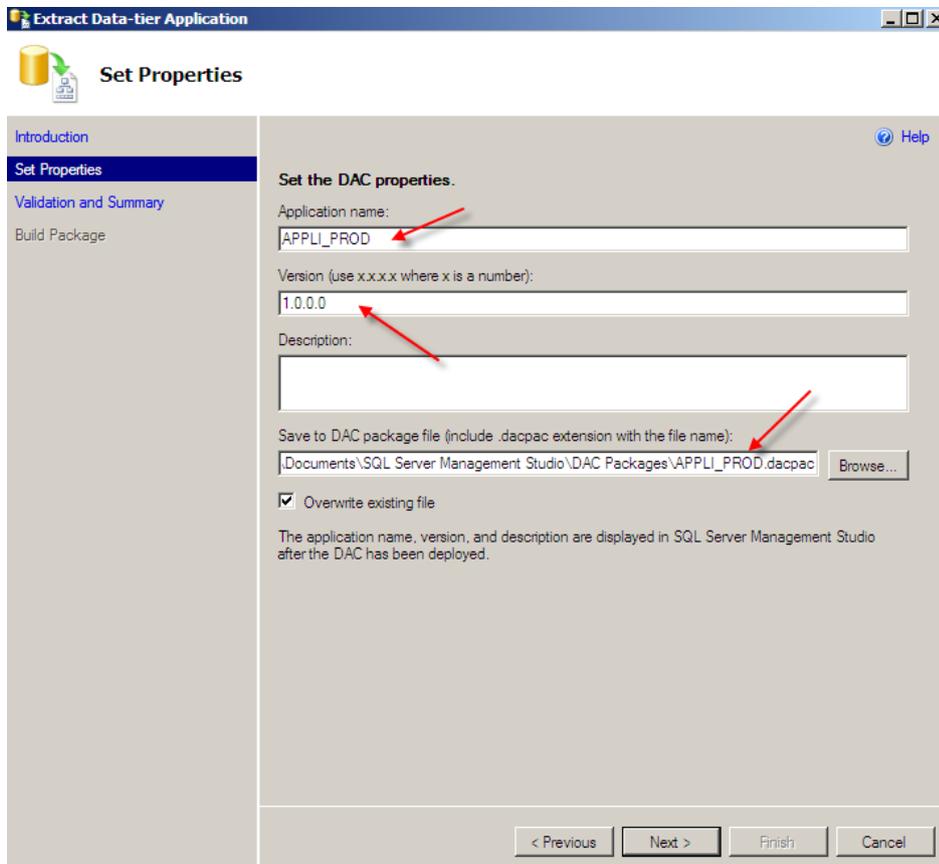
EXTRACTION D'UNE APPLICATION DAC

L'extraction à partir d'une base de données existante via SQL Server Management Studio est une autre possibilité de création d'une nouvelle application DAC. L'assistant graphique permet de générer directement un package DAC. SQL Server vérifie dans ce cas le contenu de la base de données source avec une tâche de validation des objets supportés par DAC et les comptes de connexion. Un package DAC est alors créé. Un des avantages de cette méthode est qu'elle est applicable pour différentes versions de SQL Server qui vont de 2008 à 2008 R2 pour le moment.

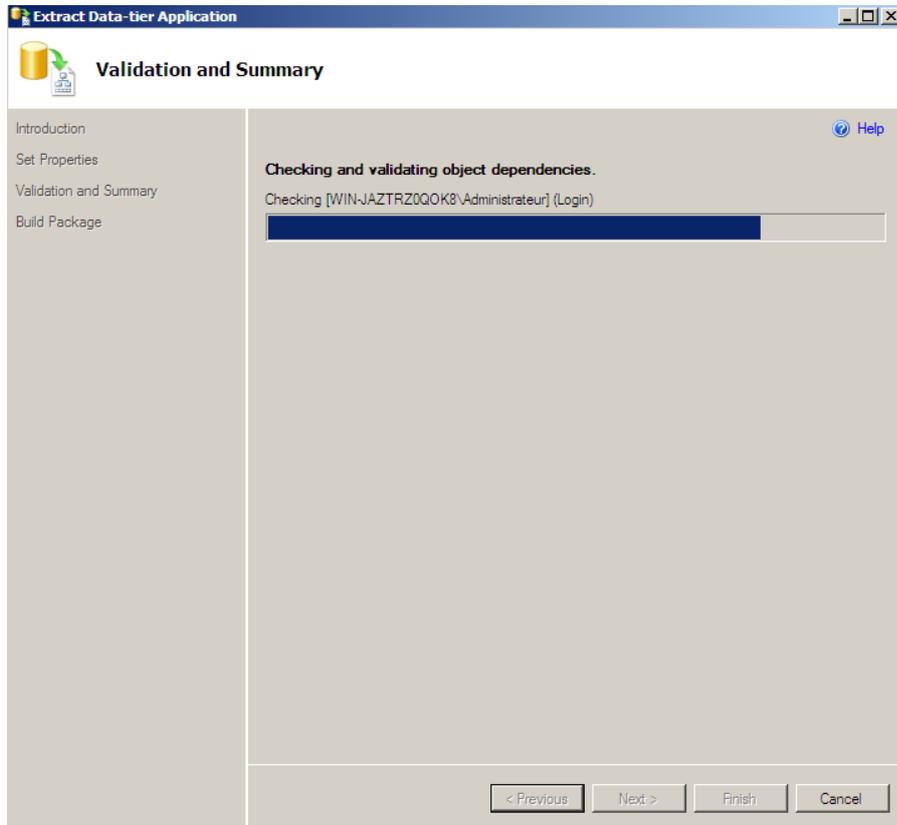




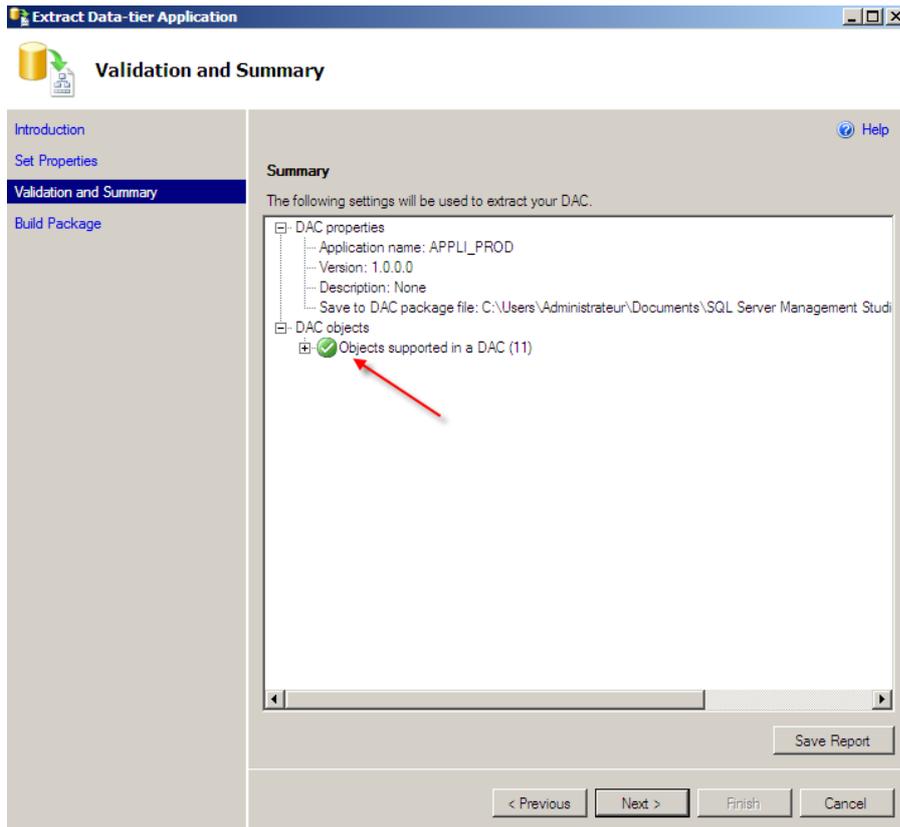
L'assistant requiert un nom pour la nouvelle application DAC, un numéro de version, une description (optionnelle) et le répertoire où sera généré le package DAC.



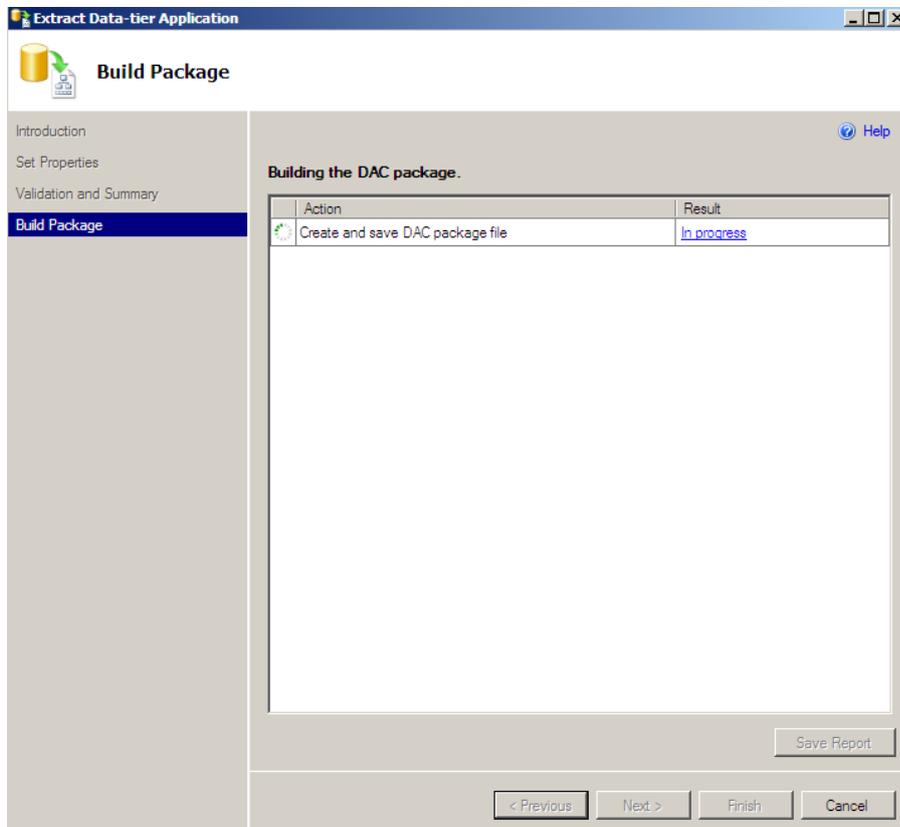
Phase de vérification du contenu de la base de données :

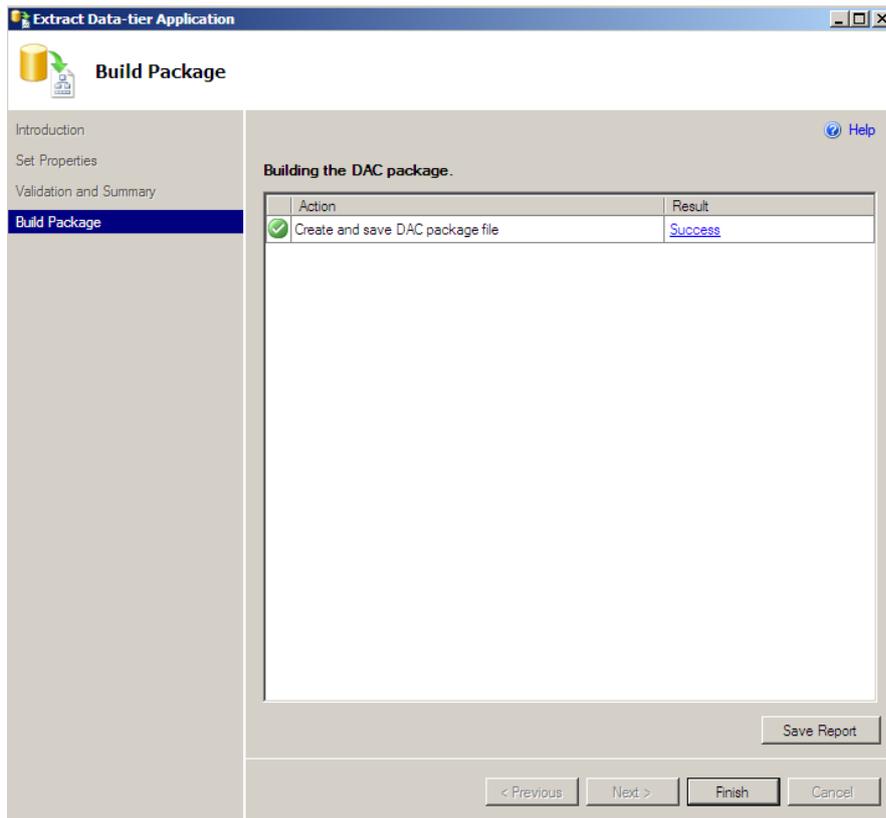


Phase de vérification réussie

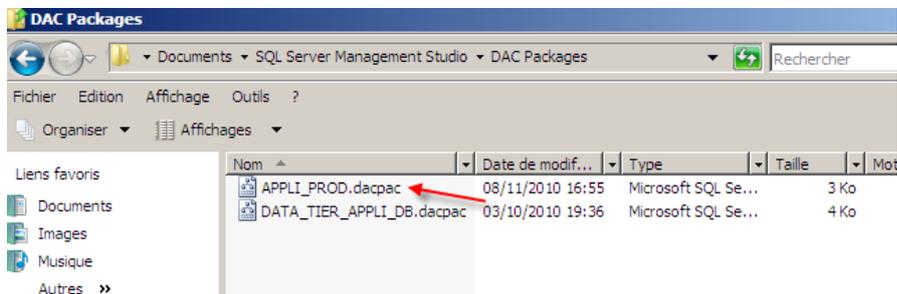


La création du package DAC peut maintenant s'effectuer





Le package DAC a bien été généré

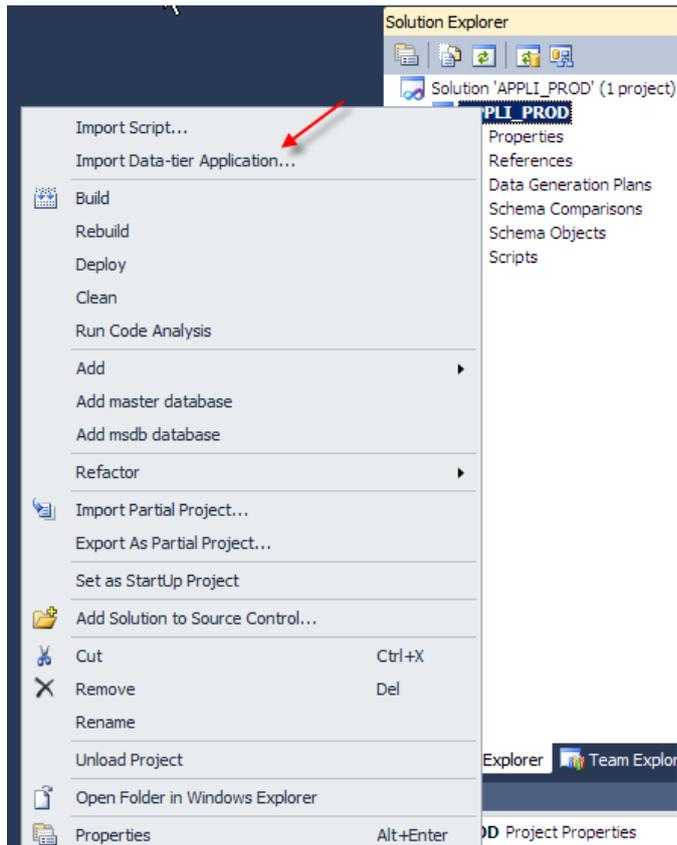


IMPORTATION D'UNE APPLICATION DAC DANS VISUAL STUDIO

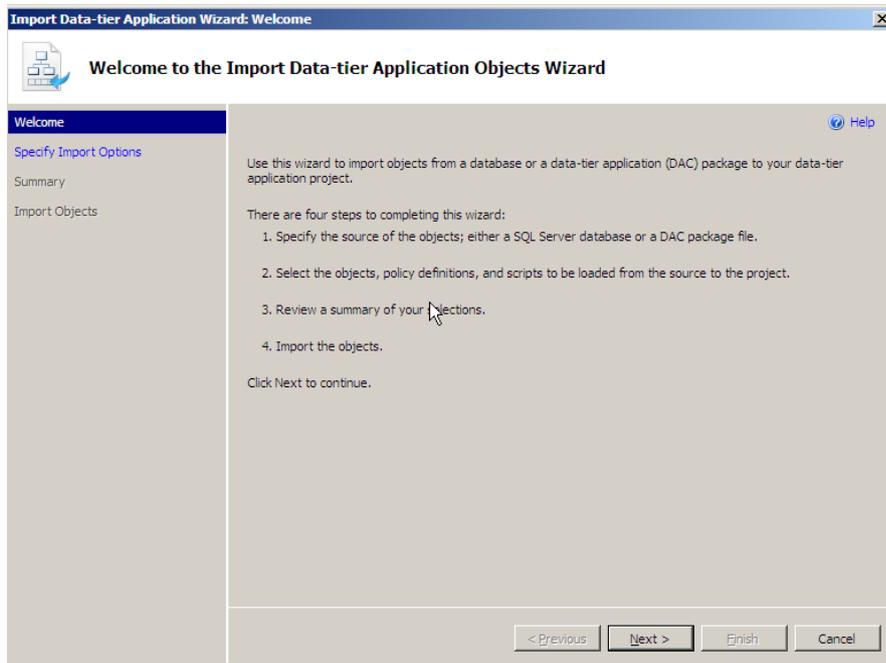
Il existe deux méthodes d'importation d'un package DAC dans Visual Studio :

- En se connectant directement sur le serveur de bases de données
- En important un package DAC préalablement généré depuis l'assistant prévu à cet effet

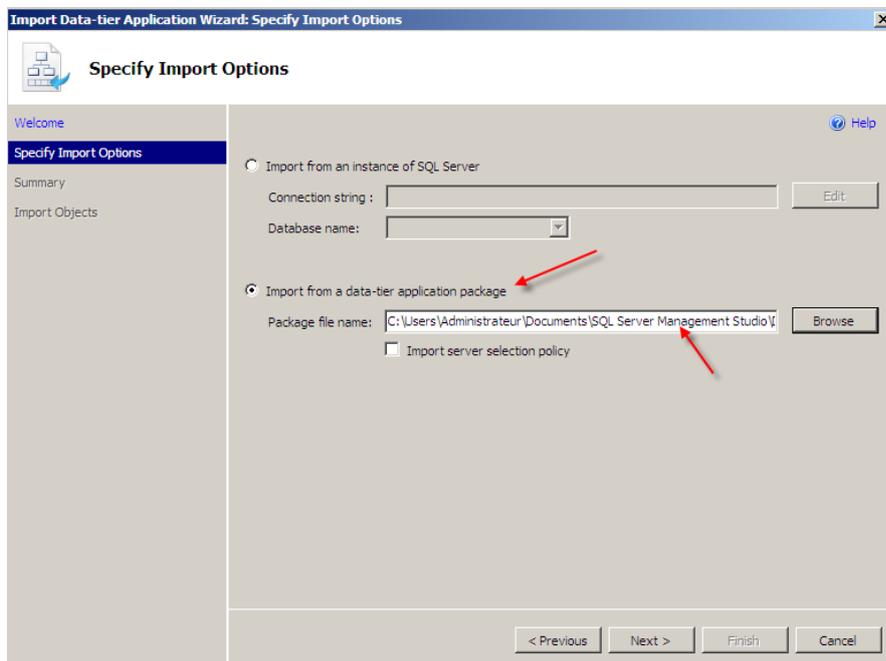
Il s'agit tout d'abord de créer un nouveau projet *Data Tier Application*. Une fois le projet créé il suffit d'importer l'application DAC avec une des méthodes citées ci-dessus. Nous utiliserons la méthode d'importation du package APPLI_PROD.dacpac généré auparavant.

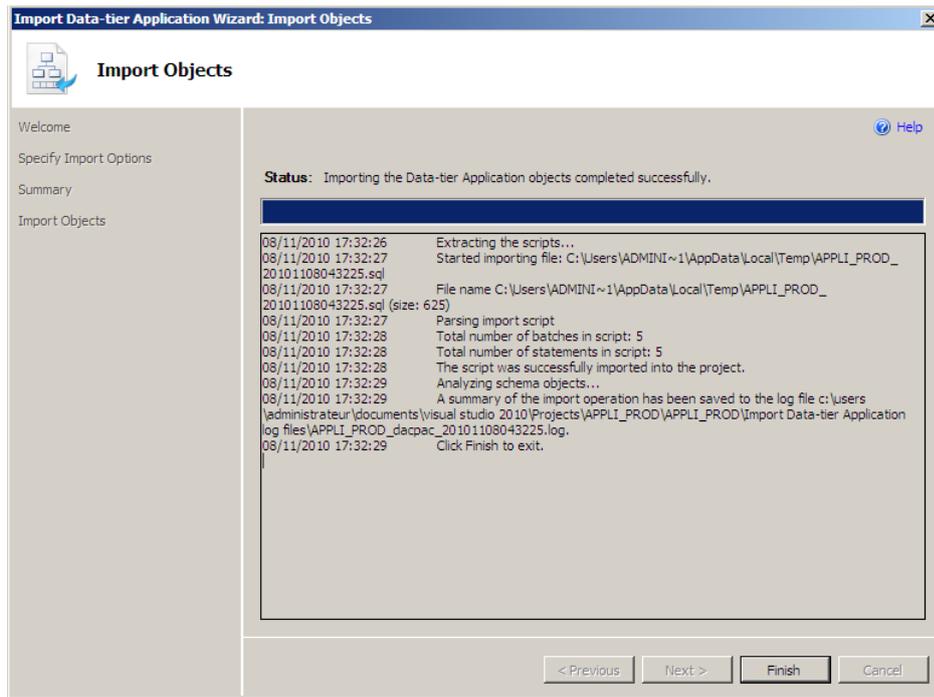


Un nouvel assistant démarre

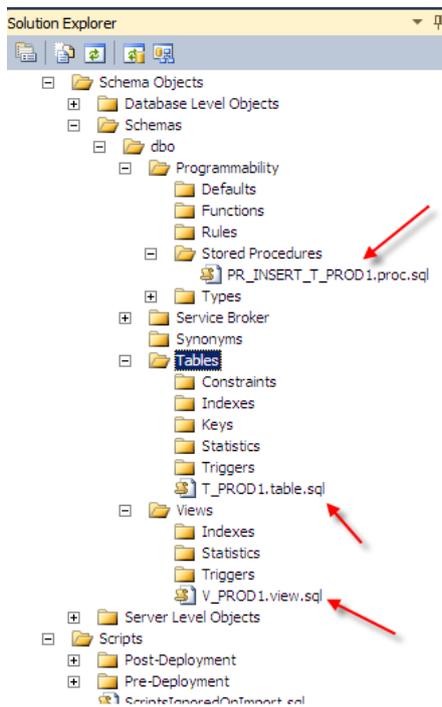


Nous devons alors choisir la méthode d'importation de l'application DAC. Nous choisirons la méthode d'importation d'un package DAC





Une fois l'application DAC importée on peut visualiser les scripts SQL générés dans l'explorateur de solutions.

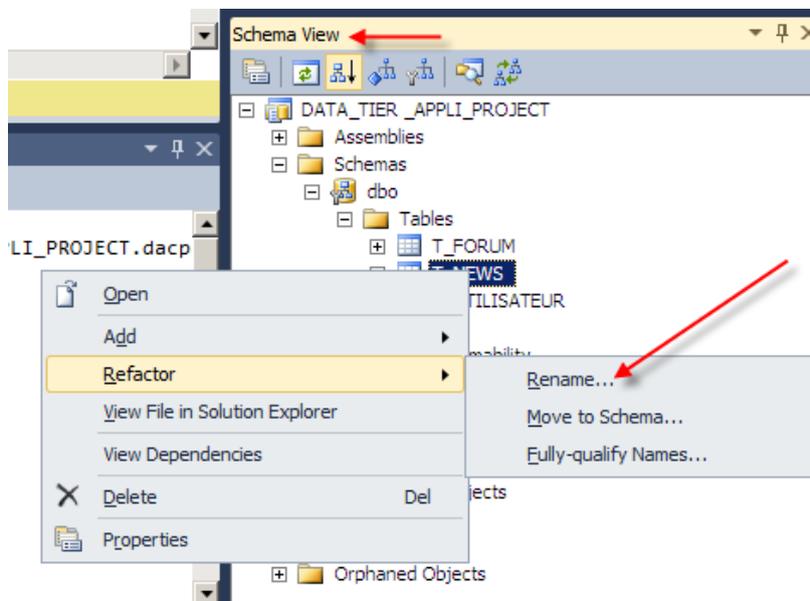


POSSIBILITES DE REFACTORING DANS VISUAL STUDIO 2010

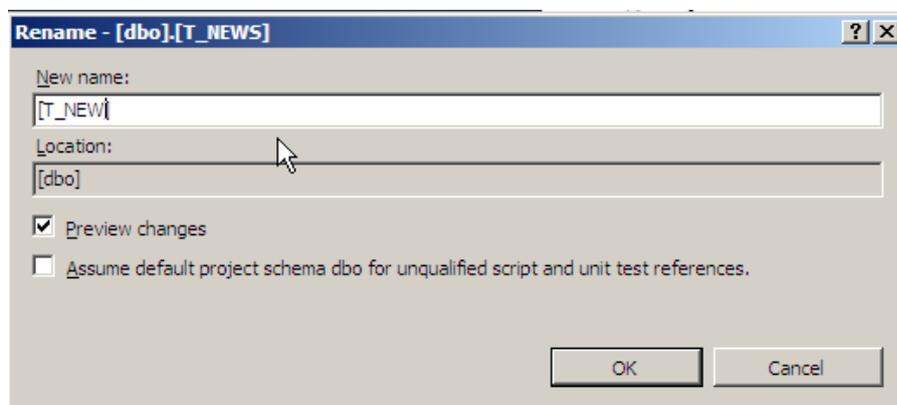
Le refactoring est une autre fonctionnalité intéressante à utiliser avec Visual Studio 2010. Celle-ci n'est pas nouvelle mais il me semble important de rappeler son existence ici. Par exemple, changer le nom d'une table qui possède de multiples références vers d'autres objets peut devenir rapidement fastidieux et source d'erreurs. Dans un projet de type SQL Server (et DAC), cette action est relativement facile à réaliser.

Dans le projet DATA_TIER_APPLI_PROJECT nous devons renommer la table T_NEWS. Celle-ci est référencée dans la procédure stockée PS_POST_NEWS et la vue V_UTILISATEUR_NEWS. Le nom de la table T_NEWS doit être, par conséquent, renommé dans ces 2 objets.

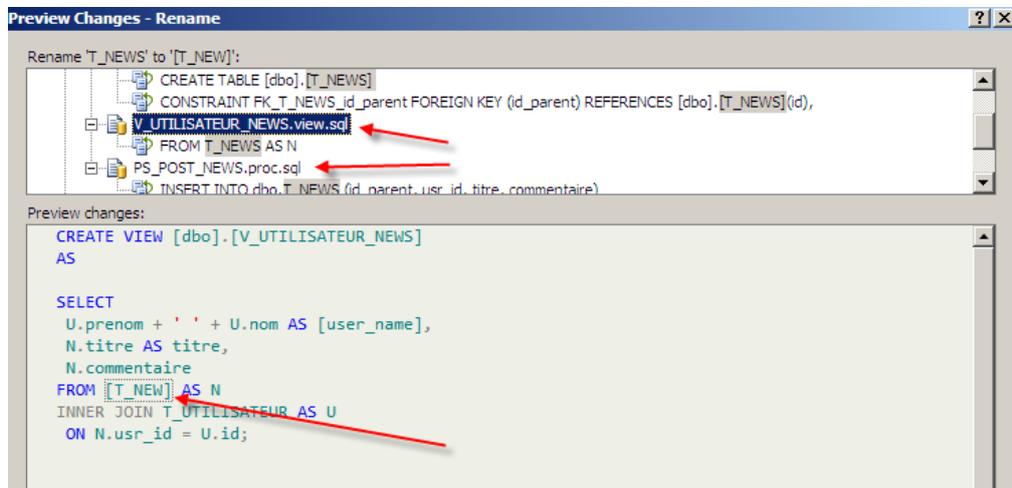
Il faut dans un premier temps se rendre dans la fenêtre de vue de schéma du projet DAC et utiliser ensuite le menu *Refactor* pour la table T_NEWS.



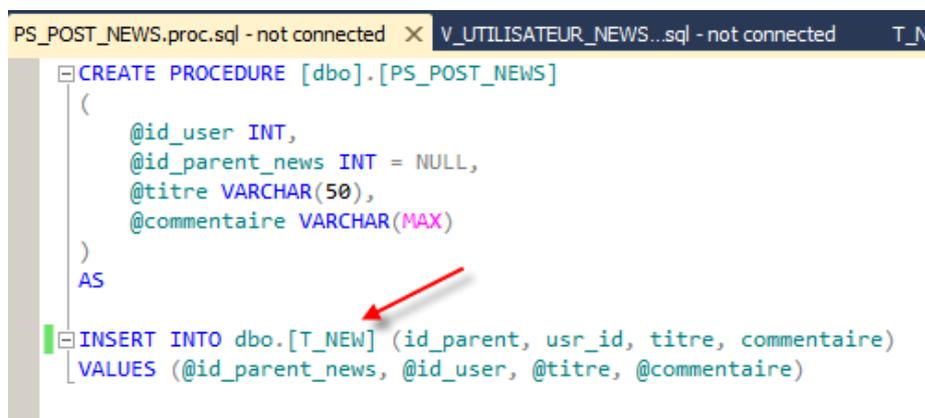
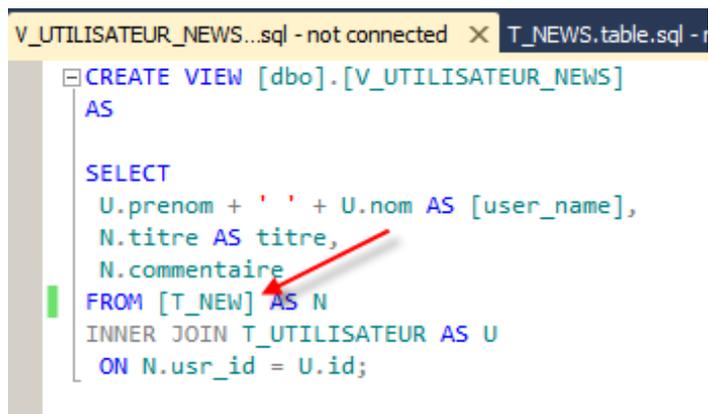
On renomme ensuite la table T_NEWS en T_NEW



Il est possible de pré-visualiser les changements que vont être opérés par le changement de nom de la table. On peut remarquer ici que les scripts SQL relatifs à la vue V_UTILISATEUR_NEWS et à la procédure stockée PS_POST_NEWS sont concernés. Il est possible de valider le changement ou de l'annuler à ce stade.



Une fois le changement effectué on peut vérifier que les deux scripts concernés.

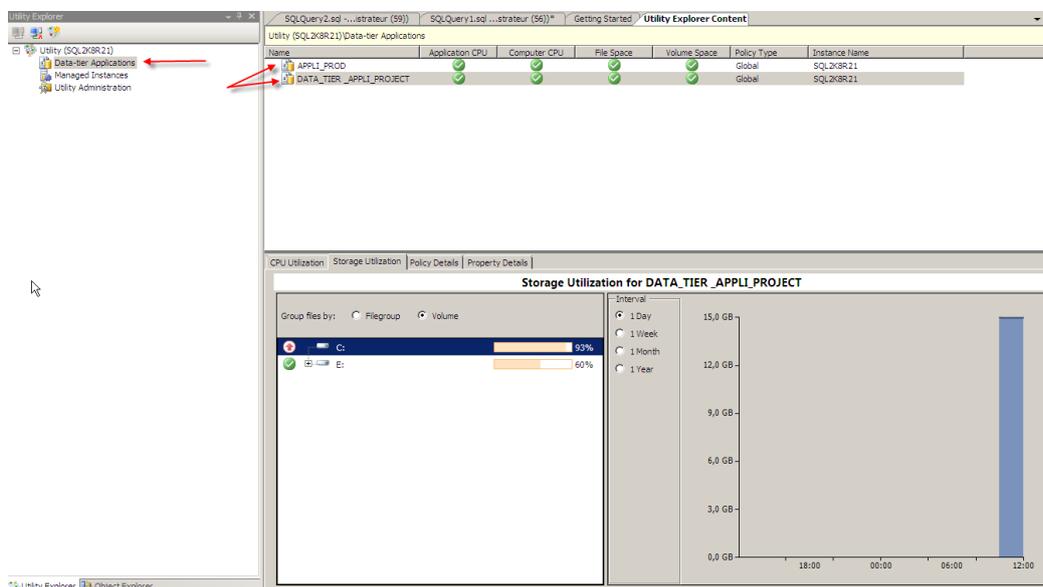


Le changement a bien été propagé dans les différents scripts concernés par la nouvelle table T_NEW.

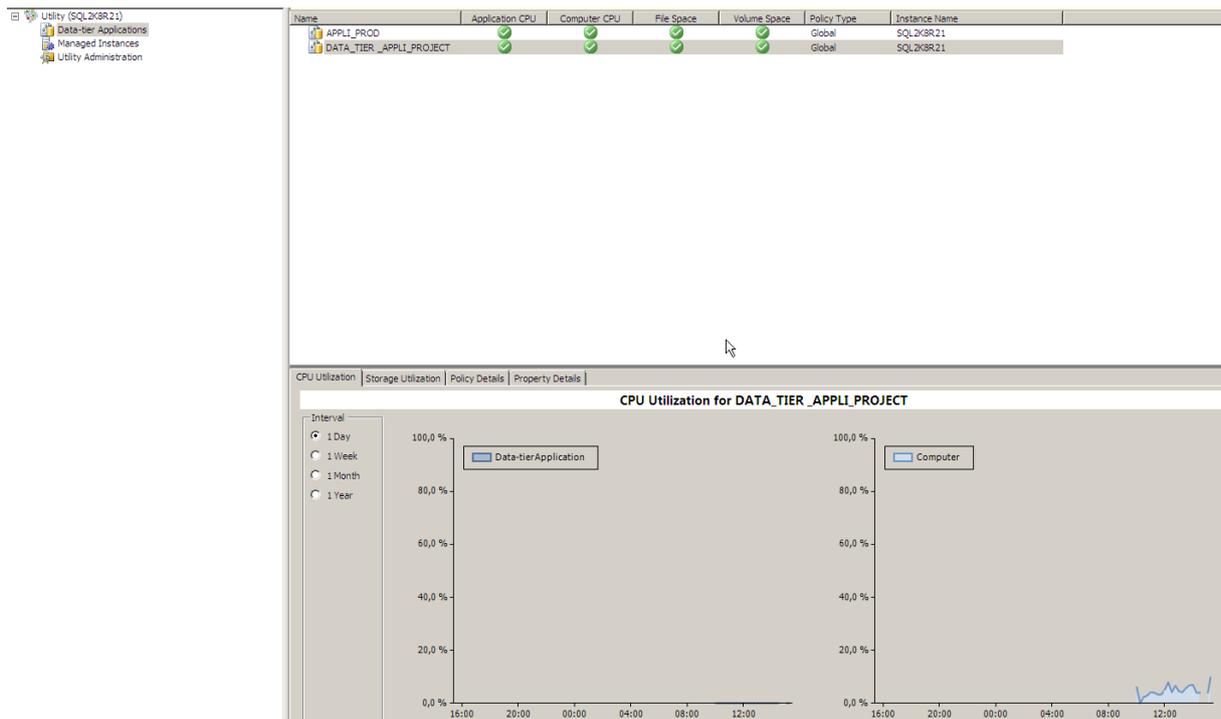
MONITORING D'UNE APPLICATION DAC

SQL Server 2008 R2 propose également une fonctionnalité de monitoring appelée *SQL Server Utility* qui permet de surveiller une ou plusieurs instances DAC. *SQL Server Utility* fournit des informations sur l'utilisation des ressources par une instance DAC qui permettent de savoir si une instance utilise ou non de façon optimale les ressources du serveur qu'il héberge. Dans le cadre d'une mutualisation des ressources ces informations peuvent être très utiles.

Utilisation du stockage :



Utilisation des ressources processeurs :



SQL Server Utility permet également de définir des seuils d'utilisation de ressources qui permettent de déclencher des alertes visuelles pour à un administrateur de bases de données afin de l'informer d'une application DAC qui respecterait pas les limites d'utilisation des ressources en adéquation avec les besoins métiers de l'entreprise.

La table système *msdb.dbo.sysdac_history_internal* peut également être utilisée comme source de monitoring 'une application DAC.

SQLQuery2.sql -...istrateur (59) | SQLQuery1.sql ...strateur (56)* | Getting Started | Utility Explorer Content

```

SELECT
  action_id,
  sequence_id,
  action_type_name,
  dac_object_type_name,
  action_status_name,
  sqlscript
FROM msdb.dbo.sysdac_history_internal

```

Results	Messages					
action_id	sequence_id	action_type_name	dac_object_type_name	action_status_name	sqlscript	
23	3	1	create	database	rollback	CREATE DATABASE [DATA_TIER_APPLI_PROJECT_1757126...
24	3	2	create objects	database	fail	SET QUOTED_IDENTIFIER ON GO CREATE TABLE [dbo].[T...
25	3	3	rename	database	not started	NULL
26	3	4	register	dacpac	not started	NULL
27	4	0	deploy	dacpac	rollback	NULL
28	4	1	create	database	rollback	CREATE DATABASE [DATA_TIER_APPLI_PROJECT_90d9660...
29	4	2	create objects	database	fail	SET QUOTED_IDENTIFIER ON GO CREATE TABLE [dbo].[T...
30	4	3	rename	database	not started	NULL
31	4	4	register	dacpac	not started	NULL
32	5	0	deploy	dacpac	success	NULL
33	5	1	create	database	success	CREATE DATABASE [DATA_TIER_APPLI_PROJECT_c810cd1...
34	5	2	create objects	database	success	SET QUOTED_IDENTIFIER ON GO CREATE TABLE [dbo].[T...

OBJETS SUPPORTES PAR DAC

Comme nous l'avons vu un peu plus haut dans l'article, la conversion d'une base de données existante en application DAC requiert à priori une phase de validation des objets supportés par DAC. La liste des objets supportés est la suivante (extrait de la BOL Microsoft) :

DATABASE ROLE	FUNCTION: Inline Table-valued
FUNCTION: Multistatement Table-valued	FUNCTION: Scalar
INDEX: Clustered	INDEX: Non-clustered
INDEX: Unique	LOGIN
SCHEMA	STORED PROCEDURE: Transact-SQL
TABLE: Check Constraint	TABLE: Collation
TABLE: Column, including computed columns	TABLE: Constraint, Default
TABLE: Constraint, Foreign Key	TABLE: Constraint, Index
TABLE: Constraint, Primary Key	TABLE: Constraint, Unique
TRIGGER: DML	TYPE: User-defined Data Type
TYPE: User-defined Table Type	USER
VIEW	

VERSIONS DE SQL SERVER SUPPORTANT DAC

Chaque version de SQL Server possède un nombre plus ou moins restreint d'opérations DAC. Il faut maintenant ajouter aux versions traditionnelles SQL Azure.

DAC Operation	SQL Server 2008 R2	SQL Azure	SQL Server 2008	SQL Server 2005	SQL Server 2000
Delete DAC	Yes	Yes	Service Pack 2 (SP2) or later	No	No
Deploy DAC	Yes	Yes	SP2 or later	No	No
Extract DAC	Yes	Yes	Yes	Yes	Yes
Register DAC	Yes	Yes	SP2 or later	No	No
Upgrade DAC	Yes	No	SP2 or later	No	No
Client Tools Support DAC Operations	Yes	No	No	No	No

CONTRAINTES D'UTILISATION

Même si DAC est une fonctionnalité intéressante à utiliser, il faudra compter avec certaines contraintes d'utilisations :

- Le changement de nom d'une base de données associée à une application DAC ne peut pas s'effectuer dans un schéma classique d'administration car les métadonnées de l'application DAC sont stockées dans la base système *msdb*.
- La structure physique de la base de données est figée. Il est impossible d'ajouter des groupes de fichiers supplémentaires ou de partitionner la base de données par exemple.
- La liste des objets supportés par DAC réduit considérablement la liste des bases de données candidates à une éventuelle conversion et les possibilités de développement.
- Les changements ne peuvent être effectués directement dans la base comme l'ajout d'un index, d'une table ou encore la modification d'une colonne. Bien que cela puisse être perçu comme une contrainte, il est tout à fait logique que ces changements soient effectués en amont dans la phase de développement du cycle de vie de l'application.
- Le processus de mise à jour d'une application DAC fait une copie de l'ancienne version de la base de données associée à l'instance DAC. L'espace disque nécessaire et le temps de mise à jour peuvent être importants en fonction de la volumétrie des données. Les bases de données de petite taille sont plus adaptées à DAC.