AUMERIAL

NTi
DATA PROVIDER

DETAILS TECHNIQUES DE LA SOLUTION

IBM
Silver Partner

Le connecteur ADO.NET natif et autonome pour accéder aux données, programmes, et services de votre IBM i, AS/400.

TABLE DES MATIÈRES

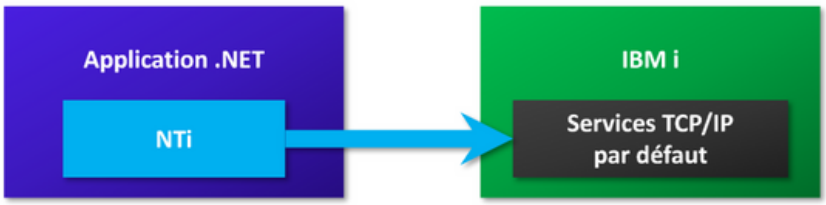
1 INTRODUCTION

1.1 .NET et le Common Language Runtime

La technologie .NET introduite par Microsoft est un incontournable du développement d'applications métiers. Grâce à la popularité de son langage C#, .NET jouit d'une grande communauté active et dynamique.

Bénéficier de l'intégralité des ressources IBM i dans des applications .NET Core modernes est aujourd'hui un enjeu de taille pour de nombreux clients. En effet, les nouvelles versions .NET 6 et 8 sont orientées cloud et multiplateformes. Plus de notion d'architecture ou de système d'exploitation, les applis deviennent portables et exécutables dans tous les environnements. Conteneurisées sur linux on Power ou déployées en client lourd Windows x86, les applications sont agnostiques à la plateforme

L'accès aux ressources IBM i se doit lui aussi d'être agnostique à la plateforme et il faut oublier les drivers compilés en code natif et autres héritages de technologies des années 90 telles que l'ODBC ou OleDB. Lorsque l'on fait du .NET, on ne veut que du .NET et c'est exactement ce qu'AUMERIAL propose avec NTi.



1 NTi, le lien direct entre .NET et l'IBM i

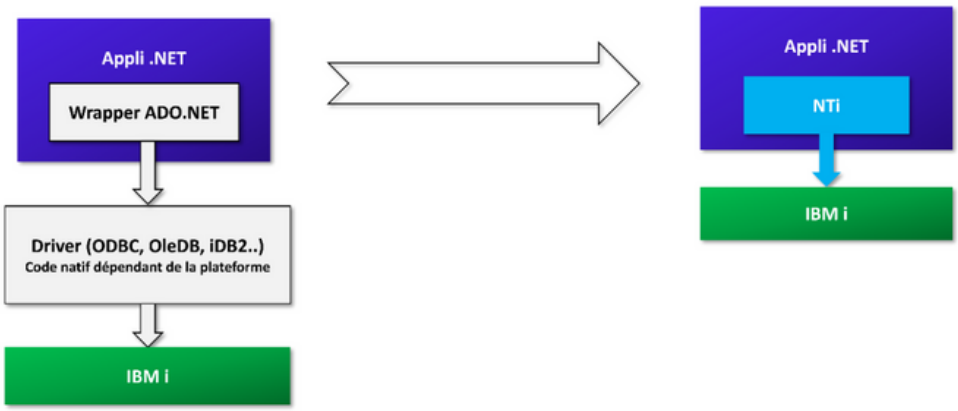
Afin de s'exécuter sur toutes les plateformes, les applications et modules développés en .NET utilisent des composants disponibles dans une bibliothèque appelées CLR (Common Language Runtime). La CLR contient tous les fondamentaux pour réaliser les différentes opérations et interagir avec le hardware. Chaque architecture dispose de son implémentation de la CLR, ce qui garantit que toutes les applications construites sur ces composants fonctionnent nativement dans ces architectures.

2 Le modèle de la CLR .NET

1.2 NTi Data Provider, la réponse d'AUMERIAL aux drivers vieillissants

Pour se connecter aux systèmes IBM i depuis .NET, on a recours à des pilotes (ou drivers) tels que l'ODBC ou Ole-DB. Ces pilotes sont écrits et compilés en code machine selon l'architecture et sont totalement en dehors du champ d'action de la CLR fournie par .NET. L'accès n'est pas réalisé par .NET et l'application qui en résulte n'est pas multiplateforme.

Face à ce constat et à l'absence de réponse satisfaisante à ce problème, nous avons développé NTi, un fournisseur d'accès aux ressources de l'IBM i à destination de .NET intégralement basé sur des ressources de la CLR .NET.



3 À gauche avec un driver, À droite avec NTi

Ainsi la totalité des tâches relatives à l'accès à l'IBM i, de l'établissement de la connexion à la conversion des pages de code, sont réalisées par du code portable sur toutes les plateformes. Par conséquent, avec NTi, il n'y a plus de contrainte quant au déploiement d'une application. On déploie ses applications métier partout sans aucune limite.

2 DÉTAILS DU FONCTIONNEMENT DE NTi

2.1 Généralités sur la connexion

NTi se connecte à l'IBM i en établissant des connexions TCP/IP avec les travaux suivants (ou leurs homologues SSL le cas échéant) :

- **QZDASOINIT** pour la base de données
- **QZRCSRVS** pour les programmes et les commandes
- **QZSOSIGN** pour le Signon

L'établissement des ces connexions est réalisé de façon standard via des composants disponibles dans le CLR .NET. Ensuite NTi soumet les différentes requêtes à l'IBM i en envoyant les DataStreams adaptés. Toutes les conversions de données, encodage/décodage de texte sont réalisées par NTi.

De par son côté fortement orienté objet, le code .NET est basé sur des instances. Très régulièrement, les instances inutiles sont détruites par .NET dans un souci d'optimisation de la mémoire. Avec NTi, la connexion TCP/IP à l'IBM i est associée à l'instance de la connexion NTi en cours.

Ainsi lors de la destruction de cette connexion, le socket TCP est fermé puis détruit et le job IBM i correspondant s'arrête. Ce principe crée un couplage fort entre les instances de NTi et les jobs IBM i, ce qui supprime les problèmes de jobs IBM i qui restent actif (QZDASOINIT « fantômes »).

> **Remarque : Il est possible de tracer les échanges entre NTi et l'IBM i à l'aide des fonctions de trace TCP intégrées à l'IBM i (STRTRCTCP).**

2.2 Configuration de la connexion

Par défaut, l'utilisateur fournit l'adresse IP ou le nom d'hôte de la LPAR IBM i à atteindre et les ports par défauts sont utilisés :

SERVICE	PORT NON SSL	PORT SSL
SIGNON	8476	9476
BASE DE DONNEES	8471	9471
COMMANDES/PROGRAMMES	8475	9475

L'utilisateur peut également choisir de spécifier lui-même les numéros de ports à utiliser pour se connecter aux différents services.

Dans le cas où la configuration n'est pas standard et que les ports sont inconnus, l'utilisateur peut choisir d'utiliser le mappeur de ports, par défaut sur le port 449. Ce port peut lui aussi être spécifié s'il est modifié.

L'utilisateur choisit également si la connexion doit ou non utiliser SSL/TLS. La configuration de SSL doit évidemment être réalisée en amont pour pouvoir établir ce type de connexion.

Si besoin il est possible de forcer la confiance dans le certificat de l'IBM i et d'utiliser SSL malgré tout.

> **Important : La connexion NTi n'est possible que si les serveurs TCP correspondant sur l'IBM i (*DATABASE, *RMTCMD, *SIGNON et *SVRMAP) sont actifs.**

2.3 Fonctionnalités

2.3.1 Accès à la base de données

À travers l'implémentation du modèle ADO.NET, NTi offre un accès total à la base de données de l'IBM i via des méthodes et une syntaxe connue de tous :

- Exécution immédiate de requêtes SQL
- Exécution de requêtes SQL préparées et paramétrées
- Exécution de procédures SQL avec ou sans paramètres d'entrée/sortie
- Ouverture et lecture des curseurs renvoyés par des requêtes ou procédures SQL
- Récupération de données dans des champs LOB (BLOB, CLOB, XML, Geospatial)
- Contrôle de validation (transactions)
- Support de tous les types de données

2.3.2 Exécution de commandes CL et appel de programmes

En plus de la partie base de données, NTi offre des méthodes pour appeler des commandes CL et des programmes IBM i sans passer par du SQL :

- Exécution de commandes CL
- Appel de programme/api avec ou sans paramètres d'entrée/sortie

Les commandes CL et les appels de programmes IBM i sont réalisés auprès du serveur dédié AS-RMTCMD. On se passe ainsi totalement du SQL dans ces cas d'utilisation.

3 MISE EN OEUVRE DE NTi

3.1 Côté Client (.NET)

3.1.1 Configuration initiale

Côté client, il n'y a aucune configuration à réaliser. Il suffit de disposer d'un accès à la partition IBM i ciblée pour pouvoir utiliser NTI. Il faut également installer la runtime .NET sur sa machine. Pas besoin d'ACS non plus puisque tout est embarqué dans le package .NET.

3.1.2 Téléchargement

NTi se télécharge directement via la plateforme NuGet.org intégrée à Visual Studio, Visual Studio code et tous les autres environnements de développement .NET sous le nom **Aumerial.Data.NTi**.

Une fois le package référencé et téléchargé (environ 400Ko), NTi est opérationnel et prêt à être utilisé.

3.2 Côté IBM i

3.2.1 Prérequis

Côté IBM i, il est évidemment nécessaire de disposer d'identifiants valides pour se connecter et que les services TCP requis soient activés (cf. Généralités sur la connexion). Autrement, aucune configuration particulière ou installation autre que la mise en place d'une clé de licence (cf. ci-dessous) n'est requise sur l'IBM i.

3.2.2 Clé de licence

Une clé de licence est nécessaire pour utiliser NTi et s'ajoute par défaut dans une bibliothèque dédiée à NTi nommée KNTI.

La création de cette bibliothèque et l'ajout de la clé s'effectuent en quelques minutes à l'aide d'un script SQL fourni par AUMERIAL. Ce script, contenant la clé et les valeurs associées, peut être exécuté directement depuis Run SQL Scripts (ACS) ou depuis tout autre client SQL connecté à la partition IBM i.

Cette clé de licence est fournie à l'utilisateur lors du démarrage d'une version d'essai de 30 jours ou à chaque renouvellement d'une période de 1an. Chaque clé de licence permet d'accéder à une seule partition d'un IBM i (identifiés par le numéro de série et l'identifiant de la partition) pendant une durée limitée.

3 CONCLUSION

En offrant un accès direct et performant à l'IBM i, NTi se positionne en tant que solution incontournable pour accéder à l'IBM i depuis .NET. Toutes les ressources de l'IBM i deviennent accessibles depuis tous les environnements via un package unique : Linux sur Power, Mono sur IBM i, conteneurisé dans Docker ou Openshift… mais aussi les services cloud tels qu'Azure ou AWS.

Enfin la mise en œuvre de NTi est simple et rapide. En quelques minutes, il est possible de migrer une application développée pour ODBC ou iDB2 et l'exploiter pleinement avec NTi.

CONTACT

Une équipe à votre service.
Demandez votre licence d'essai gratuitement dès aujourd'hui.

France - WW

Rémi ROUILLOT

+33 6 86 83 91 09

remi.rouillot@aumerial.com

DACH - PL

Tomasz AFELTOWICZ-SCHULTZ

+49 179 421 6006

tafs@aumerial.com

www.linkedin.com/company/aumerial

www.youtube.com/@aumerial

hub.docker.com/orgs/aumerial/repositories

github.com/Aumerial

Le futur de l'IBM i s'écrit en .NET

AUMERIAL

www.aumerial.com

www.documentation.aumerial.com