

A U M E R I A L

IBMi  WINDOWS - LINUX - iOS - ANDROID



[Solution overview and technical details](#)

1 TABLE OF CONTENTS

2	Introduction	3
2.1	.NET And The Common Language RUNTIME	3
2.2	NTi Data Provider, AUMERIAL'S Answer To Aging Drivers	3
3	NTi Operating Details	5
3.1	General Information About The Connection	5
3.2	Connection Configuration	5
3.3	Features	6
3.3.1	DataBase Access	6
3.3.2	Executing CL Commands and Calling Programs	6
4	NTi Implementation	6
4.1	Client Side (.NET)	6
4.1.1	Initial Configuration	6
4.1.2	Download	6
4.2	IBM i Side	6
4.2.1	Prerequisites	6
4.2.2	License Key	7
5	Conclusion	7

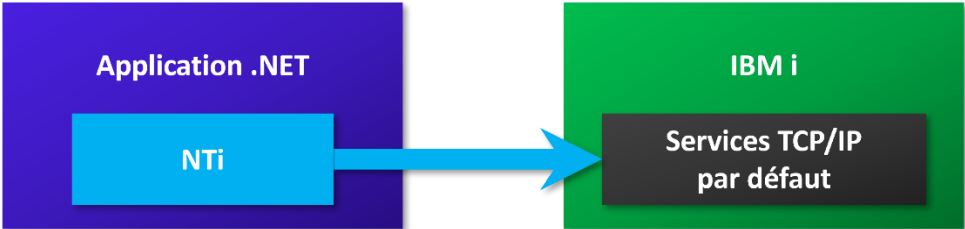
2 INTRODUCTION

2.1 .NET AND THE COMMON LANGUAGE RUNTIME

The .NET technology introduced by Microsoft is a must-have for business application development. Thanks to the popularity of its C# language, .NET enjoys a large, active and dynamic community.

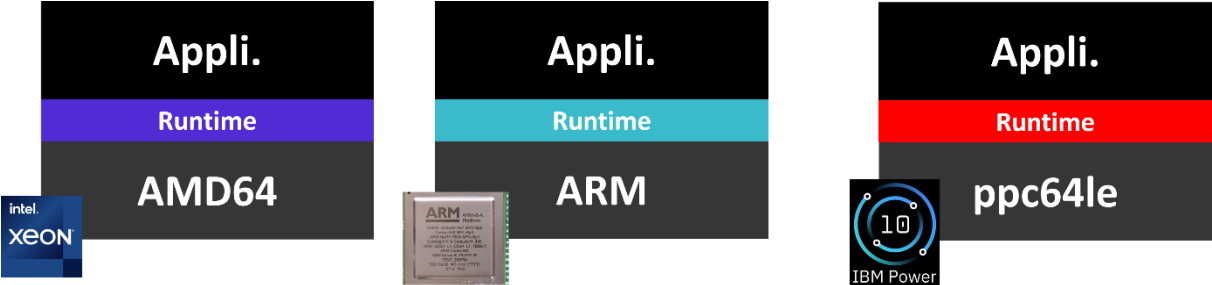
Taking full advantage of IBM i resources in modern .NET Core applications is a major challenge for many customers today. The new versions of .NET 6 and 8 are cloud-oriented and cross-platform. No longer restricted by architecture or operating system, applications become portable and executable in all environments. Containerized on Linux on Power or deployed as a Windows x86 fat client, applications are platform agnostic.

Access to IBM i resources must also be platform agnostic, so forget about drivers compiled in native code and other legacies of 90s technologies such as ODBC or OleDb. When you're doing .NET, you only want .NET, and that's exactly what AUMERIAL offers with NTi.



1 NTi, the direct link between .NET and IBM i

In order to run on all platforms, applications and modules developed in .NET use components available in a library called CLR (Common Language Runtime). The CLR contains all the fundamentals for carrying out the various operations and interacting with the hardware. Each architecture has its own CLR implementation, which guarantees that all applications built on these components run natively in these architectures.



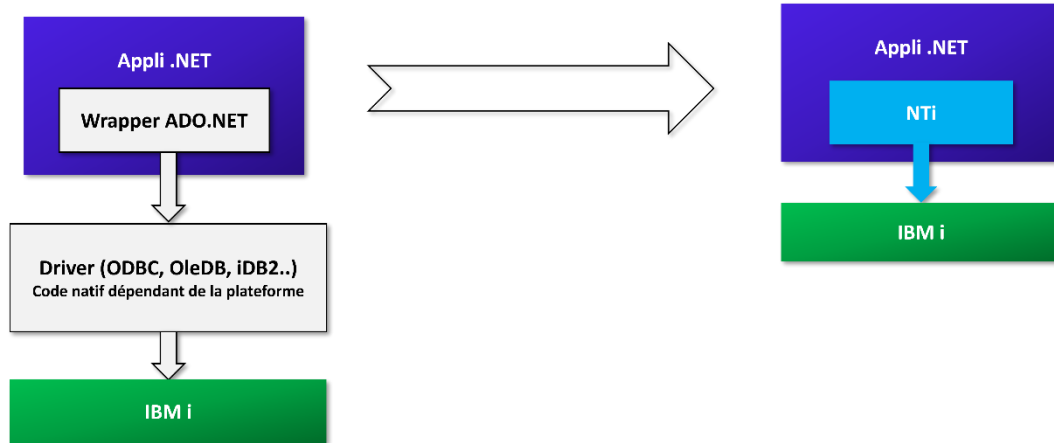
2 The .NET CLR Model

2.2 NTi DATA PROVIDER, AUMERIAL'S ANSWER TO AGING DRIVERS

To connect to IBM i systems from .NET, drivers such as ODBC or Ole-DB are required. These drivers are written and compiled in machine code according to the architecture, and are

totally outside the scope of the CLR provided by .NET. Access is not provided by .NET and the resulting application is not multiplatform.

Faced with this situation and the lack of a satisfactory solution to the problem, we developed NTi, a provider of access to IBM i resources for .NET, based entirely on .NET CLR resources.



3 Left with driver, Right with NTi

This means that all the tasks involved in accessing the IBM i, from establishing a connection to converting code pages, are carried out by code that is portable to all platforms. As a result, with NTi, there are no constraints on application deployment. Business applications can be deployed anywhere, with no limits.

3 NTi OPERATING DETAILS

3.1 GENERAL INFORMATION ABOUT THE CONNECTION

NTi connects to the IBM i by establishing TCP/IP connections with the following jobs (or their SSL counterparts if applicable):

- QZDASOINIT for the database
- QZRCSRVS for programs and command
- QZSOSIGN for Signon

These connections are established using standard components available in the .NET CLR. NTi then submits the various requests to the IBM i, sending the appropriate DataStreams. All data conversion and text encoding/decoding is performed by NTi.

Due to its highly object-oriented nature, .NET code is instance-based. On a regular basis, unnecessary instances are destroyed by .NET in the interests of memory optimization. With NTi, the TCP/IP connection to the IBM i is associated with the current NTi connection instance. When this connection is destroyed, the TCP socket is closed and then destroyed, and the corresponding IBM i job stops. This principle creates a strong coupling between NTi instances and IBM i jobs, eliminating the problem of IBM i jobs remaining active (QZDASOINIT “ghosts”).

Note : Exchanges between NTi and the IBM i can be traced using the IBM i's built-in TCP trace functions (STRTRCTCP).

3.2 CONNECTION CONFIGURATION

By default, the user provides the IP address or host name of the IBM i LPAR to be reached, and the default ports are used:

SERVICE	NON-SSL PORT	SSL PORT
SIGNON	8476	9476
DATABASE	8471	9471
COMMAND/PROGRAMS	8475	9475

Users can also choose to specify the port numbers to be used to connect to the various services.

If the configuration is not standard and the ports are unknown, the user can choose to use the port mapper, which defaults to port 449. This port can also be specified if modified.

The user also chooses whether or not the connection should use SSL/TLS. SSL must of course be configured upstream in order to establish this type of connection.

If required, you can force trust in the IBM i certificate and still use SSL.

Important : NTi connection is only possible if the corresponding TCP servers on the IBM i (*DATABASE, *RMTCMD, *SIGNON and *SVRMAP) are active.

3.3 FEATURES

3.3.1 DataBase Access

Through the implementation of the ADO.NET model, NTi offers total access to the IBM i database via methods and syntax known to all :

- Instant execution of SQL queries
- Execution of prepared and parameterized SQL queries
- Execute SQL procedures with or without input/output parameters
- Opening and reading cursors returned by SQL queries or procedures
- Data retrieval from LOB fields (BLOB, CLOB, XML, Geospatial)
- Validation check (transactions)
- Support for all data types

3.3.2 Executing CL Commands and Calling Programs

Besides to the database part, NTi offers methods for calling CL commands and IBM i programs without using SQL. :

- CL command exécution
- Program/api call with or without input/output parameters

CL commands and IBM i program calls are made to the dedicated AS-RMTCMD server. SQL is thus completely dispensed with in these use cases.

4 NTI IMPLEMENTATION

4.1 CLIENT SIDE (.NET)

4.1.1 Initial Configuration

On the client side, no configuration is required. All you need is access to the targeted IBM i partition to use NTI. You also need to install the .NET runtime on your machine. There's no need for ACS either, as everything is embedded in the .NET package..

4.1.2 Download

NTi can be downloaded directly via the NuGet.org platform integrated into Visual Studio, Visual Studio code and all other .NET development environments under the name Aumerial.Data.NTi.

Once the package has been referenced and downloaded (approx. 400Kb), NTi is up and running and ready to use.

4.2 IBM I SIDE

4.2.1 Prerequisites

On the IBM i side, it is obviously necessary to have valid login credentials and to have the required TCP services enabled (see General information on login). Otherwise, no special configuration or installation other than setting up a license key (see. [General information on connection](#)) is required on the IBM i.

4.2.2 License Key

A license key is required to use NTi, and is added by default to a dedicated NTi library called KNTI. Creating this library and adding the license key takes just a few minutes, preferably via the built-in NTi command-line interface. This license key is supplied to the user when starting a 30-day trial version or at each renewal of a 1-year period. Each license key gives access to a single IBM i partition (identified by serial number and partition ID) for a limited time.

5 CONCLUSION

By offering direct, high-performance access to the IBM i, NTi has positioned itself as the essential solution for accessing the IBM i from .NET. All IBM i resources become accessible from all environments via a single package: Linux on Power, Mono on IBM i, containerized in Docker or Openshift... but also cloud services such as Azure or AWS.

Finally, NTi is quick and easy to implement. In just a few minutes, you can migrate an application developed for ODBC or iDB2 and make full use of it with NTi.

To find out more, get a demonstration and try out NTi, please contact us:

Rémi ROUILLOT - CTO

remi.rouillot@auerial.com

+33(0) 6 86 83 91 09

AUMERIAL SAS

12 rue Albert Einstein
48000 MENDE



Silver Partner

