NTi
DATA PROVIDER

SOLUTION OVERVIEW AND TECHNICAL DETAILS

www.aumerial.com

IBM Silver Partner

The native, standalone ADO.NET provider for your IBM i / AS/400 data, programs, and services.

1 TABLE OF CONTENT

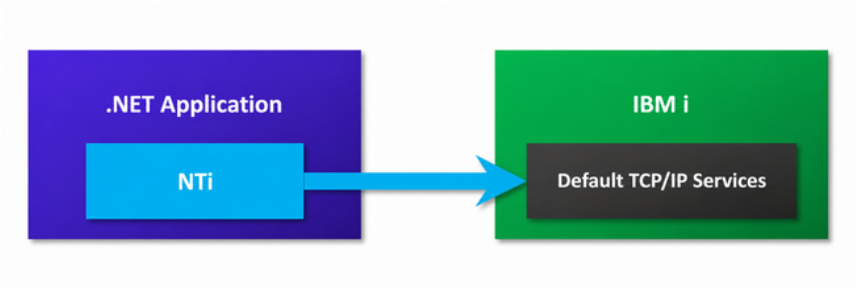
2 INTRODUCTION

2.1 .NET and the Common Language Runtime

.NET, introduced by Microsoft, has become a cornerstone of business application development. Driven by the popularity of C#, .NET benefits from a large, active, and thriving community.

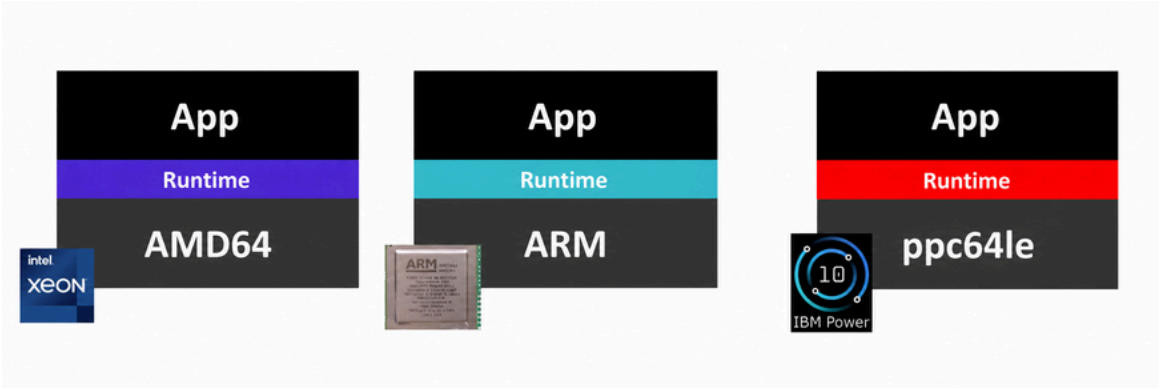
Accessing all IBM i resources from modern .NET applications is now a significant challenge for many organizations. Modern .NET versions are cloud-oriented and cross-platform by design: no architecture or OS constraints, applications are portable and runnable in any environment. Whether containerized on Linux on Power or deployed as a Windows x86 thick client, applications are fully platform-agnostic.

IBM i access must itself be platform-agnostic, leaving behind native-compiled drivers and legacy technologies from the 1990s such as ODBC and OleDB. When working in .NET, you want pure .NET, and that is exactly what AUMERIAL delivers with NTi.



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

To run across all platforms, .NET applications and modules rely on components provided by a library called the CLR (Common Language Runtime). The CLR contains all the fundamentals needed to perform operations and interact with hardware. Each architecture has its own CLR implementation, which guarantees that all applications built on these components run natively on that architecture.

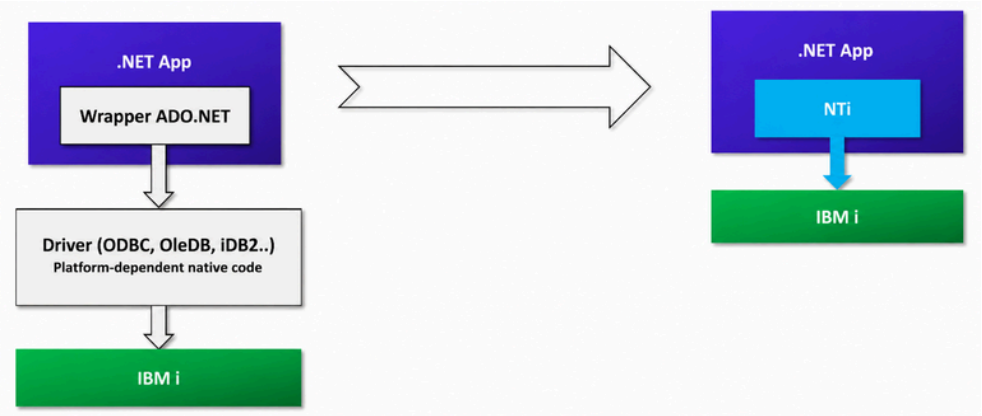


2 The .NET CLR model

2.2 NTi Data Provider, AUMERIAL's answer to aging drivers

To connect to IBM i systems from .NET, developers traditionally rely on drivers such as ODBC or OleDB. These drivers are written and compiled in machine code for a specific architecture, and fall entirely outside the scope of the CLR provided by .NET. Access is not handled by .NET itself, and the resulting application is not cross-platform.

Faced with this situation and the lack of any satisfactory answer, we developed NTi: a fully managed .NET provider for IBM i resources, built entirely on CLR components.



3 On the left with a driver, on the right with NTi

As a result, every task related to IBM i access, from establishing the connection to code page conversion, is handled by code that runs portably across all platforms. With NTi, there are no longer any deployment constraints. Business applications can be deployed anywhere, without limitation.

3 HOW IT WORKS

3.1 Connection overview

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

- **QZDASOINIT** for the database
- **QZRCSRVS** for programs and commands
- **QZSOSIGN** for sign-on

These connections are established through standard .NET CLR components. NTi then submits requests to IBM i by sending the appropriate DataStreams. All data conversions and text encoding/decoding are handled by NTi.

Due to its strongly object-oriented nature, .NET code is instance-based. Unused instances are regularly disposed of by .NET as part of memory optimization. With NTi, the TCP/IP connection to IBM i is tied to the current NTi connection instance.

When the connection instance is disposed, the TCP socket is closed and destroyed, and the corresponding IBM i job terminates. This creates a strong coupling between NTi instances and IBM i jobs, eliminating the issue of lingering active jobs (phantom QZDASOINIT jobs).

> **Note: exchanges between NTi and IBM i can be traced using the built-in IBM i TCP trace functions (STRTRCTCP).**

3.2 Connection configuration

By default, the user provides the IP address or hostname of the target IBM i LPAR, and the default ports are used:

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

The user can also choose to specify the port numbers to use when connecting to the various services.

If the configuration is non-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 it has been changed.

The user also specifies whether the connection should use SSL/TLS. SSL configuration must obviously be set up beforehand to establish this type of connection.

If needed, trust in the IBM i certificate can be forced, allowing SSL to be used regardless.

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

3.3 Features

3.3.1 Database access

Through its implementation of the ADO.NET model, NTi provides full access to the IBM i database via methods and syntax familiar to every .NET developer:

- Direct SQL query execution
- Prepared and parameterized SQL query execution
- SQL procedure execution with or without input/output parameters
- Opening and reading cursors returned by SQL queries or procedures
- LOB field access (BLOB, CLOB, XML, Geospatial)
- Commit control (transactions)
- Support for all data types

3.3.2 CL command execution and program calls

Alongside database access, NTi provides methods to call CL commands and IBM i programs without going through SQL:

- CL command execution
- Program/API calls with or without input/output parameters

CL commands and IBM i program calls are handled through the dedicated AS-RMTCMD server. SQL is therefore completely bypassed in these use cases.

3.3.3 Available extensions

NTi Data Provider includes two dedicated extensions, available as separate NuGet packages.

- **Aumerial.EntityFrameworkCore:** the extension that brings Entity Framework Core support for DB2 for i. Compatible with EF Core 8, 9 and 10, it lets you work with your IBM i database directly through C# objects, no SQL required. Define your tables as C# classes and let EF Core generate the schema (Code First), or scaffold your classes from an existing database (DB First). Queries are written in LINQ and automatically translated to SQL. Schema changes are handled through automatic migrations.

- **Aumerial.Toolbox:** the extension that brings a set of ready-to-use methods for advanced IBM i interactions: spool file management, system value retrieval, user profile information, service program calls, and more.

> **For more information on these extensions, refer to our documentation:**
> **documentation.aumerial.com**

4 GETTING STARTED WITH NTi

4.1 Client side (.NET)

4.1.1 Initial configuration

On the client side, no configuration is required. All that is needed is network access to the target IBM i partition. The .NET runtime must also be installed on the machine. ACS is not required either, as everything is included in the .NET package.

4.1.2 Download

NTi is available directly from 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 is referenced and downloaded (approximately 400KB), NTi is operational and ready to use.

4.2 IBM i side

4.2.1 Prerequisites

On the IBM i side, valid credentials are required to connect, and the necessary TCP services must be active (see Connection overview). Otherwise, no specific configuration or additional installation is required on IBM i, beyond setting up a license key (see below).

4.2.2 License key

A license key is required to use NTi and is stored by default in a dedicated library named KNTI.

The license is obtained and activated through the Aumerial portal (portal.aumerial.com) in just a few minutes:

1. Create your account at **portal.aumerial.com**
2. Enter the serial number and LPAR partition ID of your IBM i
3. Download the generated SQL script

This script, which holds the license key and associated values, can be run directly from Run SQL Scripts (ACS). Each key is tied to a single IBM i partition and is valid for a limited period. A **free 30-day trial license** is available at portal.aumerial.com.

5 CONCLUSION

By providing direct, high-performance access to IBM i, NTi establishes itself as the go-to solution for IBM i access from .NET. All IBM i resources become accessible from any environment through a single package: Linux on Power, Mono on IBM i, containerized in Docker or OpenShift, as well as cloud services such as Azure or AWS.

Finally, getting started with NTi is straightforward and fast. Within minutes, an application built for ODBC or iDB2 can be migrated and fully operational with NTi.

A TEAM AT YOUR SERVICE

NTi Data Provider is published by Aumerial, an IBM Silver partner.
Our sales team is available to answer your questions, arrange a demonstration,
or guide you through the license setup process.

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

12 Rue Albert Einstein, 48000 Mende
www.aumerial.com

The future of IBM i is written in .NET