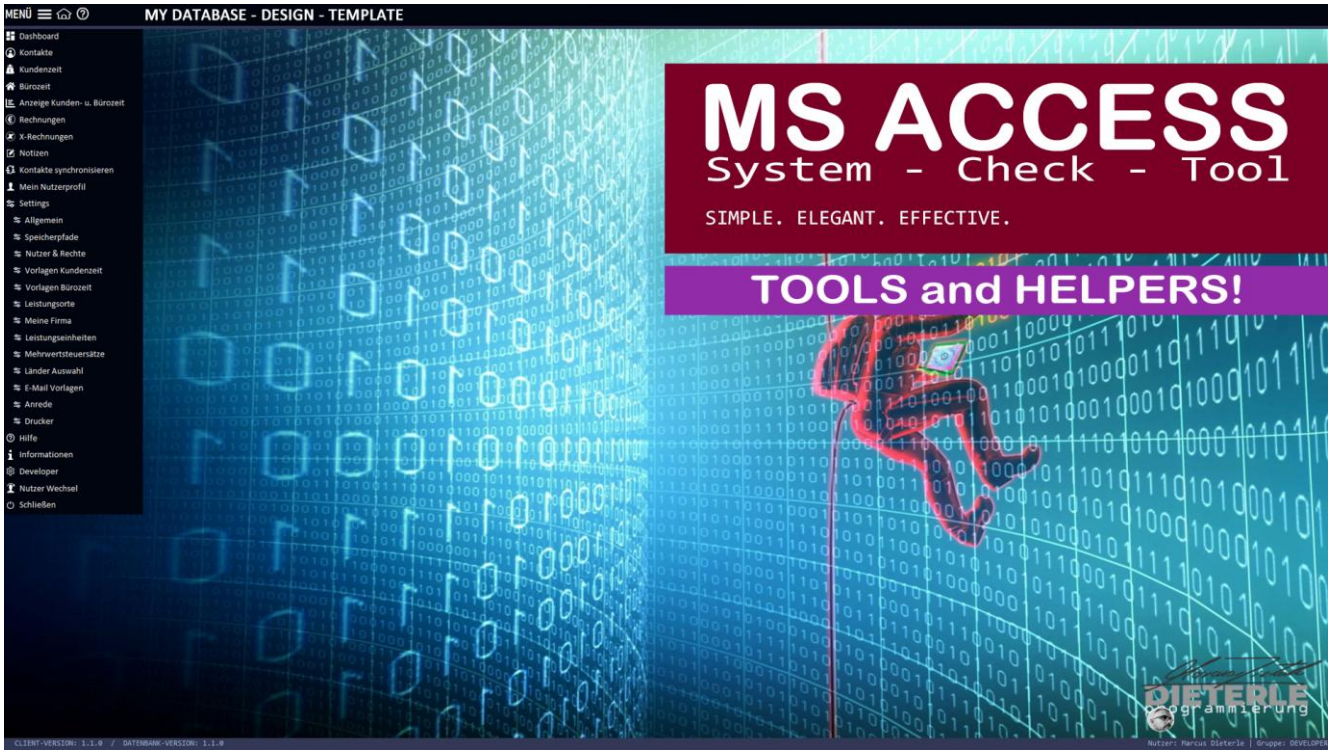

🚀 Developers System Check

Know the environment before you develop.

Developing an **MS Access** solution without knowing the target environment is like driving with your eyes closed.

You might reach your goal.
But chances are high you'll hit obstacles along the way.

We've all experienced it:

✔ Everything works perfectly on the developer machine
❌ Errors occur on the client's PC
❌ Features behave differently
❌ Performance drops
❌ API declarations fail
❌ 32/64-bit incompatibilities appear

And suddenly, troubleshooting begins.

💡 Why Environment Transparency Is Critical

Before building or deploying an Access database, you should know:

- Which Windows version is running?
- Which Office version and build?
- 32-bit or 64-bit?
- How much RAM is available?

- How many monitors are connected?
- What screen resolutions are used?
- Which references are loaded?
- What system architecture is present?

These factors directly influence:

✅ Compatibility
✅ Stability
✅ Performance
✅ Feature availability
✅ API declarations
✅ Long-term maintainability

Without this information, development becomes guesswork.

🔍 Introducing: Developers System Check

Developers System Check is a 100% VBA-based tool designed specifically for Microsoft Access environments.

It analyzes the complete system environment and provides structured, exportable data that allows developers to make informed decisions.

🧰 What Does It Analyze?

Below is an overview of key checks, what they mean, and why they matter.

🖥️ Operating System

- Windows version
- Build number
- System architecture (32/64 bit)

Why this matters:
Operating system differences can impact API compatibility, security restrictions, and available system features.

📦 Microsoft Office & Access

- Exact Office version
- Access version
- Bitness (32/64 bit)
- Installation type (e.g., Click-to-Run)

Why this matters:
Certain features and libraries are only available in specific Office versions.
32-bit vs 64-bit directly affects API declarations and compatibility.

🧠 System Architecture & Hardware

- Processor information
- Installed RAM
- System type

Why this matters:
Large datasets and complex operations require sufficient system resources.
Performance planning starts with hardware transparency.

🖥️ Monitor Configuration

- Number of monitors
- Resolution per monitor
- Screen dimensions

Why this matters:
Form layouts, popups, and UI scaling depend heavily on screen configuration.
A solution optimized for 4K may not behave well on Full HD systems.

🔐 User & Network Environment

- Username
- Computer name
- Domain information
- Network environment

Why this matters:
Many issues are user- or domain-specific.
This data enables precise identification and comparison.

📚 References & Libraries

- Loaded VBA references
- Version details

Why this matters:
Missing or incompatible references are among the most common causes of Access failures.

📤 Flexible Data Storage

Collected system data can be saved as:

📁 Access table
📄 TXT file
📄 YAML file

YAML files can also be imported, allowing aggregation of data from multiple systems.

🌍 Typical Use Case

Imagine developing a database for a company with 25 workstations.

Instead of asking every user about their setup:

1. Place the System Check tool on the company network.
2. Each user runs it once.
3. The system data is collected automatically.
4. Export the data as YAML files.
5. Import all YAML files into a central evaluation database.

The result:

- Complete overview of all environments
- Identification of inconsistencies
- Early detection of risks
- Clear foundation for technical decisions

You immediately see if:

- Some PCs are still running 32-bit Office
- Office builds differ significantly
- One workstation is underpowered
- Versions are mixed across departments

🛠️ Troubleshooting Made Efficient

One of the most common support scenarios:

“It works everywhere — except on this one PC.”

With Developers System Check, you can directly compare:

✔️ A working environment
❌ A problematic environment

Differences become visible instead of speculative. This **dramatically reduces troubleshooting time**.

🎯 Strategic Development Advantage

Before development even begins, you can decide:

- Should the environment be standardized?
- Is a version upgrade necessary?
- Should certain features be avoided?
- Is 64-bit required?

You develop based on facts — not assumptions.

⏳ Save Time. Reduce Risk. Increase Reliability.

Most issues in Access projects are not caused by the code itself.

They are caused by environmental differences.

Developers System Check makes those differences transparent.

And transparency is the foundation of professional software development.

If you develop serious Microsoft Access solutions,
understanding the target environment is not optional — it is essential.

🧪 Try it out

Using my little testing form you can try the functions directly. This makes it easy to learn the functions step by step. Do you develop a lot with Access? Then give it a try — I'm looking forward to your feedback or ideas for additional functions! 🚀