

# DataMigratePro — Getting Started with Prerequisites

Version: 2026-04 | Language: English & Deutsch

---

## 1. What is DataMigratePro?

DataMigratePro is a **direct migration and synchronization tool** that moves data from legacy Microsoft Dynamics NAV, Navision, or Business Central environments into **Dynamics 365 Business Central** (Cloud or On-Premise) **without intermediate jump versions**.

### Key Capabilities

- ✓ **Direct Migration:** Skip expensive "jump" versions; migrate directly to BC.
- ✓ **Parallel Operation:** Source and target systems remain operational during migration—no hard cutoff.
- ✓ **Live Synchronization:** Real-time bi-directional data sync (GL entries, customers, vendors, assets, custom tables, files).
- ✓ **Repeatable Deltas:** Run full migration once; then sync only changed records in iterative catch-up cycles.
- ✓ **Flexible Deployment:** Run as CLI tool, automated service daemon, or integrate via BC app configuration.

**German:** DataMigrate Pro ist die schnellste und zuverlässigste Lösung für NAV/Navision-zu-BC-Migrationen. Keine Komplexität, keine versteckten Kosten—nur hochperformante, bidirektionale Synchronisierung.

---

## 2. Prerequisites Checklist

Before starting installation and first migration, confirm you have all prerequisites in place.

### 2.1 System & Network Prerequisites

- Windows machine or server** (local or VM) with DataMigratePro client executable
- Network connectivity** to SQL Server (source database)
- Network connectivity** to Business Central OData endpoint (either cloud or on-premise)
- TLS 1.2+** support in Windows (verify: `[System.Net.ServicePointManager]::SecurityProtocol`)
- Disk space:** Minimum 2 GB for logs, temp files, and JSON chunk staging
- RAM:** Minimum 4 GB available (depends on batch size; larger batches need 8–16 GB)
- (Optional) **Azure Service Bus** connectivity if using continuous sync service mode

**Deutsch:** Stellen Sie sicher, dass Ihr Windows-System auf aktuellem Patchstand ist und alle Netzwerkverbindungen zu SQL und BC verfügbar sind.

---

### 2.2 Database & SQL Prerequisites

- SQL Server access** (SQL Auth or Integrated Windows Auth) to source database(s)
- SQL Server version:** SQL Server 2016 or newer (or Azure SQL)
- Read permissions** on source NAV/BC database (`[Company$Table]` naming convention)
- Source database name** known (e.g., NAV2017, BC\_PROD)
- (Optional) **Separate migration database** for helper tables (can be same as source during pilot)
- SQL query execution** allowed; no blocked stored procedures or disabled jobs

### Typical Source Databases:

- Dynamics NAV 2009, 2009 R2, 2013, 2013 R2, 2015, 2016, 2017, 2018
  - Business Central v14–v27
  - Azure SQL (if already cloud-hosted)
-

## 2.3 Business Central Access Prerequisites

### For SaaS (Cloud) BC Environment:

- Tenant ID** (Azure AD directory ID) — found in BC → Application Insights → Tenant ID, or from Azure Portal
- Client ID** (Azure AD app registration ID) — created in your tenant's App Registrations
- Client Secret** — generated in Azure AD app (store securely; expires annually)
- BC user** with migration/admin roles in target company
- OData endpoint URL** — pattern: `https://api.businesscentral.dynamics.com/v2.0/<tenant-id>/Production/ODataV4/Upload_LoadData?company=<company-name>`
- BC has DataMigratePro app installed** from AppSource (v2026.02+)
- BC migration configuration** exists (migration tables, field mappings, table groups)

### For On-Premise BC Environment:

- BC instance URL** and port (e.g., `https://bc-server:7048`)
- BC service account username & password** or trusted domain account
- BC user with admin role** in target company
- OData endpoint accessible** (check firewall rules)
- DataMigratePro app installed** in BC (via PowerShell or extension management)
- BC migration configuration** deployed

---

## 2.4 Credentials & Secrets Prerequisites

- SQL connection string** (or username/password if using SQL Auth) — stored securely
- Azure AD tenant credentials** (if SaaS) — stored in secure config file or environment
- BC user credentials** — with appropriate roles (migration roles, or admin for first setup)
- DataMigratePro registration key** (license) — provided by IO Integrated
- No credentials in command-line arguments** (use `settings.json` instead, with restricted file permissions)

**Security Note:** Store all credentials in `settings.json` with file permissions set to Read for system service account only. Never commit credentials to Git.

---

## 2.5 Planning & Documentation Prerequisites

- Migration scope document:** Which tables migrate? Which are excluded?
- Table mapping definition:** Source table ID ↔ target BC table ID
- Company mapping:** Source company names → target company names
- Migration timeline:** Pilot window, catch-up cycles, go-live date
- Rollback plan:** Backup of BC before go-live, restore procedure documented
- Sign-off from stakeholders:** Finance, IT, end users aware of timeline and changes
- Test data available:** Known good source data for pilot run validation

---

# 3. Installation & Setup

## 3.1 Download & Prepare Folder

1. **Download DataMigratePro package** from your distribution channel (AppSource, partner portal, or email).
2. **Create folder** for local installation:

```
mkdir "C:\Program Files\DataMigratePro"
```

3. **Extract package contents** to the folder:
  - DataMigratePro.exe
  - DataMigratePro.dll and supporting libraries
  - scripts/ folder (PowerShell helpers, argument completer)

- o docs/ (optional, for reference)

#### 4. Verify executable exists and runs:

```
cd "C:\Program Files\DataMigratePro"
.\DataMigratePro.exe --help
```

Expected: Shows usage and available tasks.

---

## 3.2 Create settings.json for Your Environment

settings.json is the **primary configuration file** that persists all connection details, timeouts, and BC configuration.

### Example for SaaS (Azure AD Authentication)

```
{
  "SqlConnectionString": "Data Source=SQLSERVER01;Initial Catalog=NAV2017_DB;Integrated
    Security=SSPI;",
  "SourceDatabase": "NAV2017_DB",
  "SourceCompany": "CRONUS AG",
  "DestinationCompany": "CRONUS AG",
  "MappingDatabase": "NAV2017_DB",
  "MigrationDatabase": "NAV2017_DB",
  "EndpointUrl": "https://api.businesscentral.dynamics.com/v2.0/12345678-1234-5678-1234-
    567890123456/Production/ODataV4/Upload_LoadData?company=CRONUS%20AG",
  "IsSaaS": true,
  "Environment": "Production",
  "HttpClientTimeoutSeconds": 100,
  "TenantInfo": {
    "TenantId": "12345678-1234-5678-1234-567890123456",
    "ClientId": "aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee",
    "ClientSecret": "YOUR_SECRET_HERE"
  },
  "BlobMagicSignature": [ 2, 69, 125, 91 ],
  "Verbose": false
}
```

### Key fields:

- SqlConnectionString: SQL Server connection to **source database**
- TenantInfo: Azure AD credentials for SaaS BC (leave empty for On-Prem)
- EndpointUrl: BC OData Upload\_LoadData endpoint
- HttpClientTimeoutSeconds: Increase for large payloads (default 100s; try 300s+ for production)

### Example for On-Premise BC

```
{
  "SqlConnectionString": "Data Source=SQLSERVER01;Initial Catalog=NAV2017;Integrated Security=SSPI;",
  "SourceDatabase": "NAV2017",
  "SourceCompany": "CRONUS AG",
  "DestinationCompany": "CRONUS AG",
  "MappingDatabase": "NAV2017",
  "MigrationDatabase": "NAV2017",
  "EndpointUrl": "https://bc-server.company.local:7048/BC/ODataV4/Upload_LoadData?company=CRONUS%20AG",
  "IsSaaS": false,
  "Environment": "Production",
  "HttpClientTimeoutSeconds": 100,
  "BcUserName": "domain\\user",
  "BcUserPassword": "password_or_PAT",
  "BlobMagicSignature": [ 2, 69, 125, 91 ],
  "Verbose": false
}
```

**Deutsch Hinweis:** Speichern Sie settings.json in einem sicheren Verzeichnis mit eingeschränkten Berechtigungen (nur System-Service-Konto oder Administrator).

---

### 3.3 Install DataMigratePro App in Business Central

#### For SaaS BC:

1. Go to **Microsoft AppSource** and search "DataMigratePro"
2. Click **Get it now** → authenticate with your BC tenant
3. Choose **Production** environment → **Install**
4. Wait for installation to complete (5–10 minutes)
5. In BC, go to **Business Central Admin Center** → verify app is listed as **DataMigrate Pro**

#### For On-Premise BC:

1. Download DataMigratePro app .app file from partner portal
2. Connect to your BC PowerShell admin:

```
$adminModulePath = "C:\Program Files\Microsoft Dynamics 365 Business
Central\220\Service\NavAdminTool.ps1"
. $adminModulePath
Publish-NAVApp -ServerInstance BC -Path "C:\path\to\DataMigratePro.app" `
-SkipValidation
```

3. Restart BC service and verify app appears in **Extensions**

---

### 3.4 Configure Migration Tables in BC App

After app installation, configure the migration target:

1. In BC, open **DataMigratePro Setup** (search role center)
2. Fill in:
  - o **Source Database:** NAV2017
  - o **Source Company:** CRONUS AG
  - o **Destination Company:** CRONUS AG
  - o **Migration Tool User:** System user for this migration
  - o **Update Mode:** (leave default unless instructed otherwise)
3. Click **Load Configuration from Local DB** if running local instance
4. Save and close

**German:** Diese Konfiguration verbindet die BC-App mit Ihrer lokalen Migration-DB und dem DataMigratePro-Client.

---

### 3.5 Optional: Install as Windows Service for Continuous Sync

If you plan **unattended / daemon-mode operation** (e.g., nightly sync runs):

1. **Test the listener first (foreground mode):**

```
cd "C:\Program Files\DataMigratePro"
.\DataMigratePro.exe -t listen `
--registration "YOUR_REGISTRATION_KEY" `
--tenantid "YOUR_TENANT_ID"
```

Press Ctrl+C after ~10 seconds to verify it starts without errors.

2. **Install as Windows Service:**

```
.\DataMigratePro.exe -t listen --install "DataMigrate PRO Sync Service" `
--registration "YOUR_REGISTRATION_KEY" `
--tenantid "YOUR_TENANT_ID"
```

3. **Verify service is installed:**

```
Get-Service "DataMigrate PRO Sync Service"
```

Expected: Status = **Stopped** (starts manually or scheduled)

#### 4. Start service:

```
Start-Service "DataMigrate PRO Sync Service"
```

#### 5. Verify in Event Viewer:

- Open **Event Viewer** → **Windows Logs** → **Application**
- Filter for **DataMigratePro** → Check for startup successes or errors

---

## 4. First Run: Single Table Test

Before migrating all data, run a **small test** to validate mapping and connectivity.

### Step 1: Register License (One-Time)

Register your DataMigratePro license key (provided by IO Integrated):

```
cd "C:\Program Files\DataMigratePro"

$RegistrationKey = "YOUR_REGISTRATION_KEY_HERE"
$TenantId = "12345678-1234-5678-1234-567890123456"

.\DataMigratePro.exe `
  --registration $RegistrationKey `
  --tenantid $TenantId `
  --sqlconnection "Data Source=SQLSERVER01;Initial Catalog=NAV2017;Integrated Security=SSPI;" `
  -w "https://api.businesscentral.dynamics.com/v2.0/$TenantId/Production/ODataV4/Upload_LoadData?
      company=CRONUS%20AG"
```

**Expected:** Message confirms registration validity. Your settings are saved to `settings.json`.

---

### Step 2: Load Configuration from BC

Pull the migration configuration from your BC instance (table mappings, field definitions, etc.):

```
.\DataMigratePro.exe -t loadconfiguration
```

**Expected:** Output shows `Configuration loaded successfully` → local copy of BC migration config is ready.

---

### Step 3: Run Single-Table Test with Validation

Test with the **Customer table** (table ID 18 in most NAV/BC systems):

```
.\DataMigratePro.exe -t putdata `
  -s 18 `
  -i "SELECT TOP 10 * FROM [CRONUS AG$Customer]" `
  --automapping 18 `
  --testrun
```

#### Parameters:

- `-s 18` = Migrate to BC Customer table
- `-i "SELECT ..."` = SQL query to fetch up to 10 test records
- `--automapping 18` = Auto-detect field mappings based on source table structure
- `--testrun` = Send only 1 record per table to validate mapping without full upload

**Expected outcome:**

- No errors in console output
- BC logs show 1 test record received
- No validation errors logged

**If errors occur:** See Troubleshooting section below.

---

**Step 4: Inspect Test Results in BC**

1. In BC, open **Posted Entries** or **General Journal** (depending on source table)
  2. Filter for today's date
  3. Verify test record exists with expected field values
  4. If fields look wrong, check mapping file and rerun Step 3 with corrected mapping
- 

**5. First Full Migration: Master Data**

After successful test run, migrate all master data tables (customers, vendors, dimensions, etc.).

**Single Command: Full Master Data Setup**

```
.\DataMigratePro.exe -t putalldata `
--tablegroup "1,2" `
--automapping `
--changesonly 0
```

**Explanation:**

- `--tablegroup "1,2"` = Migrate table groups 1 (customers, vendors) and 2 (GL accounts, dimensions)
- `--automapping` = Use auto-detected field mappings for each table
- `--changesonly 0` = Full export (all records, no filtering by change date)

**Expected duration:** 5–60 minutes (depends on data volume and network latency)

**Expected output:**

- Progress messages showing tables being migrated
  - Final summary: X records transferred, Y errors, Z warnings
- 

**Validation After Full Run****1. Check BC company:**

- Dashboard shows GL account balances, customer count, vendor count
- Verify balances match source NAV/BC

**2. Check logs:**

```
Get-ChildItem "C:\Program Files\DataMigratePro\logs\*" |
Sort-Object LastWriteTime -Descending |
Select-Object -First 1 |
Get-Content -Tail 50
```

**3. Count records in BC vs. source:**

- Open BC → Table Browser → select each table → count records
  - Cross-check against source database record counts
-

## 6. Validation Checklist Before Go-Live

Before switching to production, validate these items:

### Data Integrity

- GL Balances:** YTD, MTD, and period balances match source system
- Open Documents:** All unpaid invoices, purchase orders, sales orders migrated
- Posted Transactions:** All GL entries, item ledger entries, customer ledger entries populated
- Dimension Sets:** Dimension IDs, values, and assignments transferred correctly
- Custom Tables:** Any custom NAV tables migrated to corresponding BC custom tables
- Files/Media:** Document references, images uploaded successfully

### System Readiness

- BC Users:** All users have licenses and are assigned roles
- Data Entry:** Test users can create invoice/PO and post without errors
- Reporting:** Standard BC reports display correct data (AR aging, AP aging, GL trial balance)
- Integrations:** Any 3rd-party apps or API integrations tested with migrated data
- Backup:** Full backup taken before switching off NAV/Navision

### Performance & Capacity

- BC capacity plan:** Storage, database size, and licensing match data load
- Licensing:** All required modules licensed in BC (Finance, Inventory, HR, etc.)
- Query performance:** AD-hoc lookups (customers, items) complete in <2 seconds
- User load test:** 10+ concurrent users worked without timeouts

---

## 7. Troubleshooting: Common Issues

### Issue: "Registration invalid" or "License expired"

**Symptom:** Error message: Registration key not valid for this tenant or License expired.

**Causes:**

- Registration key is for different tenant
- License has expired (annual renewal required)
- Registration key copied incorrectly (extra spaces, wrong format)

**Fix:**

1. Double-check registration key from email/license document (remove leading/trailing spaces)
2. Confirm `--tenantid` matches your BC tenant (ask Microsoft or check BC → **Admin Center**)
3. Contact IO Integrated support if license has expired

---

### Issue: "Connection string invalid" or "SQL Server not found"

**Symptom:** Error: Named Pipes Provider, error: 40 or Cannot open database 'X' or Login failed.

**Causes:**

- SQL Server hostname/IP wrong
- SQL Server firewall blocked connection
- Database name typo
- SQL user has no permission to source database

**Fix:**

1. **Test SQL connection manually** from Windows machine:

```
$conn = New-Object System.Data.SqlClient.SqlConnection
$conn.ConnectionString = "Data Source=SQLSERVER01;Initial Catalog=NAV2017;Integrated Security=SSPI;"
$conn.Open()
$conn.State # Should print "Open"
```

2. **Check SQL credentials in settings.json** — verify spelling and permissions
3. **Check SQL Server firewall** — allow port 1433 (default) from DataMigratePro client machine
4. **Verify database exists** — run SQL Server Management Studio query: `SELECT @@SERVERNAME; SELECT DB_NAME();`

---

## Issue: "BC Endpoint unreachable" or "401 Unauthorized"

**Symptom:** Error: Unable to connect to the remote server or The remote server returned an error: (401) Unauthorized.

### Causes:

- -w endpoint URL typo or wrong environment
- Azure AD credentials (tenant/client/secret) incorrect
- BC user doesn't have migration permissions
- TLS certificate issue (on-prem BC)

### Fix (SaaS):

#### 1. Verify endpoint URL format:

```
https://api.businesscentral.dynamics.com/v2.0/TENANT-ID/Production/ODataV4/Upload_LoadData?
company=COMPANY_NAME
```

- Replace TENANT-ID with your actual tenant ID
- Replace COMPANY\_NAME with your BC company name
- URL-encode company name (spaces → %20)

#### 2. Verify Azure AD credentials:

- Tenant ID: Open **Azure Portal** → **tenant info** or in BC admin center
- Client ID: **Azure Portal** → **App Registrations** → your app → **Application (client) ID**
- Client Secret: **Azure Portal** → **App Registrations** → your app → **Certificates & secrets** → **Client secrets**

#### 3. Check BC user permissions:

- In BC, user must be assigned to BC role with migration permissions (or admin role temporarily)

### Fix (On-Prem):

#### 1. Verify endpoint URL and port:

```
https://bc-server.company.local:7048/BC/ODataV4/Upload_LoadData
```

#### 2. Verify BC service account credentials and SSL certificate:

```
Get-NAVServerInstance -ServerInstance BC |
Format-List ServiceAccount, SSLEnabled, AcsUri
```

#### 3. Check firewall on BC server:

- Allow inbound HTTPS on port 7048 from DataMigratePro client IP
-

## Issue: "No records transferred" or "0 records"

**Symptom:** Migration completes with no errors, but 0 records appear in BC.

### Causes:

- SQL query returned empty result set
- Field mapping incomplete or wrong
- BC table has validation errors (rejects records silently)

### Fix:

#### 1. Validate SQL query manually:

```
SELECT TOP 10 * FROM [CRONUS AG$Customer]
```

If query returns rows, problem is downstream.

#### 2. Check mapping file:

- Verify column names in `-i` query match mapping file field names
- Run `--testrun` to get detailed validation errors

#### 3. Check BC logs:

- BC → **Diagnostics** → **Error Logs** (search for today's date)
- Look for validation errors like `Required field empty` or type mismatches

#### 4. Increase verbosity:

```
.\DataMigratePro.exe -t putdata ... --verbose 1
```

Review detailed logs in `logs/` folder for field-level errors.

---

## Issue: "Timeout" or "Request took too long"

**Symptom:** Error: The operation timed out or `HttpRequestException` after 100 seconds.

### Causes:

- BC endpoint busy or slow
- Large batch of records (> 100K rows)
- Network latency to BC endpoint
- SQL query inefficient

### Fix:

#### 1. Increase timeout in settings.json:

```
"HttpClientTimeoutSeconds": 300
```

Then re-run command.

#### 2. Reduce batch size: Use `-r <range>` to split large table migrations into smaller chunks:

```
.\DataMigratePro.exe -t putdata -s 18 -i "SELECT * FROM ..." \  
-r 1..50000
```

#### 3. Run during off-peak hours if BC is under high user load

#### 4. Optimize SQL query — add indexes, use `WITH(NOLOCK)`, avoid full table scans

---

## 8. Next Steps After Successful Setup

1. **Repeat full migration for other table groups** (transactions, dimensions, etc.)
2. **Set up delta sync cycles** using `--changesonly 1` for catch-up runs
3. **Configure Windows Service** (`-t listen`) for unattended nightly syncs
4. **Train users** on new BC interface and business processes
5. **Decommission NAV/Navision** after go-live cutover and successful validation
6. **Archive old NAV database** for compliance/audit trail retention

---

## 9. Getting More Help

Need	Resource
<b>CLI parameters</b>	See <code>DataMigratePro-Parameters-Quick-Reference.pdf</code> or <code>docs/DataMigratePro-CLI-Reference.md</code>
<b>Scenario examples</b>	See <code>docs/DataMigratePro-CLI-Szenarien.md</code> (German; comprehensive workflows)
<b>Configuration details</b>	See <code>docs/settings.json.md</code>
<b>Performance tuning</b>	See <code>docs/Performance-Logging-Guide.md</code>
<b>Support tickets</b>	Contact IO Integrated: <a href="mailto:kontakt@ioi.gmbh">kontakt@ioi.gmbh</a> or +49 (0)214 8402 3000

---

### Checklist Summary

- All prerequisites confirmed (§2)
- Installation complete (§3)
- First test run successful (§4)
- Full master data migrated (§5)
- Validation checklist passed (§6)
- Ready for go-live to production BC

---

Version 2026-04 | Bilingual Setup Guide | For DataMigratePro ≥ v2026.02 | Approved for Distribution

---

### IO Integrated GmbH & Co. KG

Experts in Business Central Migration & Data Integration

#### Contact:

- Email: [kontakt@ioi.gmbh](mailto:kontakt@ioi.gmbh)
- Phone: +49 (0)214 8402 3000
- Web: <https://ioi.gmbh/>

*This document is the property of IO Integrated GmbH & Co. KG and intended for authorized use only.*