Runtime Execution Profiles Reference

Manage multiple execution profiles.

Reference **Solution** **Runtime** **UI** Startup | Build and Publish | Execution Profiles | Monitor

---

Runtime Execution Profiles (Reference)

Manage database connections and device nodes for different environments, automating configuration changes between development, validation, and production stages.

Access: Runtime Execution Profiles

Profile Overview
Configuration Interface
Connection Replacement
Using Profiles
Windows Service Configuration
Typical Configurations
Best Practices
Verification
Troubleshooting

Profile Overview

Execution Profiles eliminate manual configuration changes when moving solutions between environments:

Profile	Purpose	Typical Use
Production	No replacements	Live system
Development	Dev connections	Testing/debugging
Validation	Test environment	QA/staging
Custom	User-defined	Lab/customer site

Production profile uses solution's main configuration. Other profiles apply replacements at runtime.

Configuration Interface

Profile Settings

For each profile (Development/Validation/Custom):

Setting	Function
Enable Profile Settings	Activate this profile
Set ReadOnly on Modules	Prevent modifications
On Startup Replace Connections	Define replacements

Read-Only Modules

Select modules to protect from changes:

- Tags
- Devices
- Alarms
- Scripts
- Displays

AI Runtime-Write Gate

Each profile carries a per-profile gate that controls whether the DesignerMCP **set_object_value** tool is allowed to drive runtime values (Tag.* and other Object Model paths) under that profile. The tool is the AI authoring counterpart to a Script writing into a tag — same write semantics (eRunEvent. Changed: equal-write is silent), same kernel coercion, same downstream effects on subscribers (Scripts on Tag.Changed, Alarms watching the value, Historian on-change samplers, Device output handlers).

When a profile blocks AI runtime writes, the tool refuses with errorCode `AI_OBJECT_WRITE_DISABLED` and a structured payload pointing to this column — the AI surfaces the exact "uncheck the AIWriteReadOnly flag on this profile" guidance to the user.

Profile	Default	Rationale
Production	Blocked	Customer install never exposes AI-driven runtime writes by default. Production profile is the safe baseline.
Development	Allowed	Engineer-sandbox default. DesignerMCP sessions are profile-locked to Development; the AI authoring loop sees the tool working out of the box.
Validation	Blocked	QA/staging is treated like Production for AI-write safety. Engineer opts in per solution by unchecking the flag.
Custom	Blocked	Lab / customer / training profile defaults to safe. Engineer opts in per solution as needed.

The flag has the same polarity as the Read-Only Modules family above — *checked* means blocked, *unchecked* means allowed. Partner integrations that write tags via the WebAPI use a separate JWT-scoped gate model (see the Runtime REST API Reference) and are unaffected by this column.

Connection Replacement

Replacement Table Columns

Column	Description	Example
ReplaceObject	Object to replace	DB_Production
Replace	Enable replacement	True/False
TableType	Configuration table	DatasetsDBs
GatewayIP	New gateway address	192.168.1.100
ReplaceConnection	New connection string	Full connection
DateCreated	Entry creation	Timestamp
DateModified	Last change	Timestamp

Supported Replacements

Type	Table	Use Case
Database	DatasetsDBs	Different SQL servers
Device Node	DevicesNodes	Different PLCs
Historian	HistorianStorageLocation	Different storage
TagProvider	UnsTagProviders	Different sources

Using Profiles

In Designer

1. Go to Runtime Startup
2. Select Execution Profile
3. Run solution
4. Profile replacements apply automatically

Command Line

bash

```
# Development
TStartup.exe /solution:"solution.dbsln" /profile:1

# Validation
TStartup.exe /solution:"solution.dbsln" /profile:2

# Custom
TStartup.exe /solution:"solution.dbsln" /profile:3
```

Windows Service Configuration

Running Profiles as Service

AutoStart in Solution Center and Designer installs the Production profile. To run another profile as the service, install it from an elevated shell with `/profile:<n>`, instead of editing the registry:

```
TManageServices.exe /installtstartup /solution:"C:\Solutions\Plant1.dbsln" /profile:1 /silent
```

The service is named `TStartup-<SolutionName>`. It takes its port and modules from the selected profile, so no port parameter is needed. For start, stop, status and remove, see TManageServices CLI.

Profile Index Values

Profile	Command Parameter
Production	(none)
Development	/profile:1
Validation	/profile:2
Custom	/profile:3

The value is the profile number. A profile name such as `/profile:Development` is not accepted and is read as Production.

Typical Configurations

Development Profile

Common replacements:

- Local test database
- Simulation devices
- Debug file paths
- Test email server

Validation Profile

Common replacements:

- Staging database
- Test PLCs
- QA file shares
- Test notifications

Custom Profile

Use cases:

- Laboratory setup
- Customer demo
- Training environment
- Pilot installation

Best Practices

Profile Design

DO:

- Keep Production unchanged
- Document all replacements
- Test profile switching
- Use meaningful names

DON'T:

- Modify Production directly

- Mix environment data
- Skip validation testing
- Use production credentials

Security Considerations

- Never store production passwords in profiles
- Use environment-specific credentials
- Limit profile access
- Audit profile changes
- Leave AI Runtime-Write blocked on Production / Validation / Custom unless an engineer has explicitly opted in for an authoring session

---

Verification

Check Active Profile

Monitor shows current profile:

- Info.ProfileName property
- Runtime System Monitor page
- Client status bar

Verify Replacements

Confirm replacements applied:

1. Check connection strings
2. Test device communication
3. Verify database access
4. Monitor data flow

Troubleshooting

Issue	Cause	Solution
Profile not switching	Not enabled	Enable profile settings
Connections unchanged	Replace unchecked	Check replacement flag
Service wrong profile	Service installed without `/profile`	Reinstall with `TManageServices.exe /installtstartup /solution:"<path>" /profile:<n>`
Port conflicts	Same port used	Change profile ports
AI `set_object_value` refuses with AI_OBJECT_WRITE_DISABLED	AI Runtime-Write Gate is blocked on the active profile	Open Runtime Execution Profiles, select the active profile row, uncheck the AI runtime-write flag (Development is unchecked by default; Production / Validation / Custom ship checked).

In this section...
