TATSOFT

GUIDE

Build Your First SCADA Solution with AI

A practical walkthrough using FrameworX AI Designer and Claude Desktop

18
MCP tools built in

5,000+
Deployments worldwide

2 hrs
ProveIT Conference

PREREQUISITES

- FrameworX installed
- Claude Desktop connected
- MCP configured

TATSOFT

FrameworX AI Designer

What's inside

This guide walks you through building a complete water treatment monitoring solution with AI Designer — pillar by pillar.

SECTION 01

How AI Designer Works

AI Designer ships with three distinct MCP services. Understanding which one to use — and why they are separate — sets you up for every step in this guide.

LIVE IDE CO-PILOT

DesignerMCP

Connects to: Designer.exe

Tools: 18 tools

Connects to the running Designer IDE. Claude works directly inside your project — every change appears in real time with an orange border and AI Designer badge. Covers the full solution lifecycle: tags, alarms, historian, devices, protocols, displays, symbols, scripts, UNS nodes, security, and documentation.

Use for active project development.

OFFLINE ENGINEERING

ConsoleMCP

Connects to: File system

Tools: File-based

File-based engineering via Claude Code. No running Designer needed. AI generates JSON configuration files in the FrameworX export format. Import into Designer for validation and deployment. Useful for batch builds from a specification document or for projects without a running environment.

Use for batch generation from specs.

LIVE DATA QUERIES

AI Runtime

Connects to: TServer.exe

Tools: 7+ tools + custom

Connects to the running solution server for operations-side work: tag values, alarm status, historian trends. Engineers create custom MCP tools in C# that expose domain-specific queries with validation and access control — 'get tank levels' instead of reading a raw tag path.

Use for operations, dashboards, reporting.

💡 These three services connect to different processes for a reason — mixing design-time configuration, offline engineering, and runtime data access would compromise all three.

SECTION 01 CONTINUED

The Ownership Model

The most common question when seeing AI work inside a live engineering tool: who controls what it creates? The MCP Category system is the answer.

The AI is a collaborator. The engineer is the authority.

FIRST TOOL CALL TRIGGERS AN APPROVAL PROMPT

Authorization

Nothing happens until you confirm the connection. The AI does not execute anything in the background before you approve the session. The Designer shows a visual indicator — orange border and AI Designer badge — whenever AI is actively working.

TAGGED WITH THE MCP CATEGORY

AI-created objects

Objects the AI creates are tagged as AI-owned and fully modifiable by the AI. Every change is logged with the same audit trail that captures human changes. You always know what the AI built.

HUMAN-CONTROLLED BY DEFAULT

Engineer-created objects

For objects you created manually, the AI is limited to annotating the Description field only. To give the AI full access to a human-created object, manually re-add the MCP Category. This is a collaboration model, not a lockout.

⚠️ Ask the AI to add 12 tags — it adds 12 and stops. Ask it to review alarm config against ISA-18.2 — it reviews and reports back. The AI responds to explicit requests only.

SECTION 02

Before You Build

Two steps before your first AI session: structure your starter prompt and define what you know about your project.

THE STARTER PROMPT

Paste this at the start of every new Claude session.

```
You are an automation engineer working in
FrameworX by Tatsoft, specializing in SCADA
and HMI development.

About me: I am a [engineer / SI / student]
with [X] years of experience.

Context: I am working on [solution name].
[Describe the solution in 1-2 sentences.]

Work with me like this:
- Build one pillar at a time. Confirm before
  moving to the next.
- After each step, list objects and paths.
- If uncertain about any detail, ask me.
  Do not guess.
```

WHY THIS STRUCTURE WORKS

ROLE	Sets Claude's expertise and platform context
CONTEXT	Your specific solution, goals, and constraints
INSTRUCTIONS	How Claude should work with you step by step

DEFINE BEFORE YOUR FIRST SESSION

Protocol:	Vendor name is enough — AI finds the driver
Tag paths / folders:	Or tell Claude to follow ISA-95
Data types:	FLOAT, INT, BOOL per instrument group
Alarm setpoints:	Hi, Hi-Hi, Lo, Lo-Lo per measurement
What you don't know:	List unknowns — ask Claude to suggest

SECTION 03

Build Pillar by Pillar

Complete and verify each pillar before moving to the next. The order matters — alarms need tags, scripts need devices.

01 Tags | 02 Devices | 03 Alarms | 04 Historian | 05 Displays | 06 Scripts

01 Tags

Your tag hierarchy is the foundation. Get it right and everything else builds on it. The AI reads your ISA-95 structure, checks existing naming patterns, and creates tags with correct types, paths, and scale ranges.

SAMPLE PROMPT

> Create a tag hierarchy for a water treatment plant using ISA-95 naming: [Site].[Area].[Unit].[Equipment].[TagName]. Include 12 instruments: 3 flow meters (Modbus FLOAT), 2 pH sensors (Modbus FLOAT), 3 turbidity monitors (Modbus FLOAT), 2 chemical level sensors (Modbus FLOAT), 2 pump run/fault status (Modbus BOOL). Create all tags in /Tags/WaterTreatment.

✓ Open Tags in Designer. Confirm 12 tags with correct types, paths, and ISA-95 names. Search for any tags with missing scale ranges.

02 Devices

The AI searches protocols by vendor name and configures channels, nodes, and data points. Give it the IP address and protocol and it handles the register addressing. No manual lookup of the communication driver reference.

SAMPLE PROMPT

> Add a Modbus TCP device at 192.168.1.10 port 502. Channel name: WTP_PLC. Map all 12 instrument tags to Modbus registers: flow meters at 40001-40006 (FLOAT, holding), pH at 40007-40010, turbidity at 40011-40016, chemical levels at 40017-40020, pump status at 10001-10004 (BOOL, discrete inputs).

✓ Check Device Channels in Designer. Confirm the channel shows Connected. Force a scan and confirm tag values updating. Any Bad Quality tags need register address review.

03 Alarms

Describe alarm requirements in plain language. AI generates ISA-18.2 compliant configuration with correct deadbands per measurement type. Review and adjust site-specific setpoints before accepting.

SAMPLE PROMPT

> Configure alarms for all 12 instruments following ISA-18.2. Flow meters: High at 150 L/min, deadband 5%. pH: Low 6.5, High 8.5, deadband 0.1 pH. Turbidity: High 4 NTU, High-High 10 NTU, deadband 0.2 NTU. Chemical levels: Low 20%, Low-Low 10%, deadband 2%. Pump fault: Critical priority.

✓ Open Alarm Configuration. Confirm limits, priorities, and deadbands per tag. Force a test by writing a value above the high limit and confirm the alarm appears in the active list.

Complete and verify each pillar before moving to the next.

04 Historian

AI recommends logging rates and deadbands based on equipment type, not generic defaults. Process values, alarm state changes, and pump cycles each log at rates appropriate for how the data will be used.

SAMPLE PROMPT

> Set up historian logging for all 12 tags. Process measurements (flow, pH, turbidity, chemical levels): log on change, deadband 0.5% of span, minimum interval 5 seconds. Pump status: log every state change. Create trend groups: ProcessValues and EquipmentStatus. Set retention to 90 days.

✓ Open Historian Configuration. Confirm all 12 tags appear with correct deadbands and intervals. Force a value change and confirm a historian record is written.

05 Displays

AI builds the structure and binds all tags correctly. Plan for 30 to 60 minutes of visual polish after generation — element sizing, alignment, and color consistency need manual refinement. Image placement on canvas displays is always manual.

SAMPLE PROMPT

> Create a main operator display called WTP_Overview. Include: current values for all 12 instruments, an active alarms summary table (last 10), pump status indicators (green/red from run status), and a trend chart for both flow meters. Use a dark theme consistent with High Performance HMI principles. Bind all values to existing tags.

✓ Open the display in Designer. Confirm tag bindings are correct and showing live values. Check alarm table updates when you trigger a test alarm. Review layout and adjust element sizing as needed.

06 Scripts

Describe the behavior you want and AI writes parametric, documented scripts. Always review the generated logic and confirm variable names match your actual tag paths before enabling the script in the runtime.

SAMPLE PROMPT

> Write a script that runs every 60 seconds and checks if either pump has been in fault status for more than 5 minutes. If so, send an operator notification with the pump name, fault duration, and the current flow value from its associated flow meter. Name the script PumpFaultMonitor. Add a comment block explaining the logic.

✓ Open the script in Designer. Read the logic and confirm tag paths match your actual tag names. Force a pump fault tag to True and confirm the script fires and produces the notification.

SECTION 09

Working With Existing Solutions

You do not need a new project to get value from AI Designer. Connect to any existing solution and use these three workflows.

AUDIT

Audit

Open a solution built by someone else. Ask the AI to audit the entire application: all tags, alarm configuration status, display bindings, and known issues. Get a full picture in 10 minutes — not days of reading undocumented code.

SAMPLE PROMPT

"Review the entire solution. List all tags, identify any with missing alarm configuration, report any display objects bound to tags that do not exist, and flag any scripts referencing undefined tag paths."

→ A complete audit report in the chat. Act on the items or ask the AI to fix specific issues.

DOC

Document

Generate documentation that typically gets deferred under project pressure. AI reads the live solution state and writes descriptions, narratives, and summaries directly from what is actually configured — not from memory.

SAMPLE PROMPT

"Generate full documentation for this solution: tag descriptions for all tags, alarm narratives for each alarm group, a device channel summary, and a one-page system overview describing the solution's purpose and architecture."

→ Complete documentation generated from the actual solution state. Review and adjust operator-facing language.

EXTEND

Extend

Add a new system or subsystem to an existing application without breaking existing configuration. The AI reads the current project structure, follows established naming conventions, and builds the new section consistently.

SAMPLE PROMPT

"Add a new chemical dosing system to the existing solution. Follow the same ISA-95 naming convention and folder structure already in use. Include 4 new instruments: one flow meter, one pH sensor, one chemical level sensor, one dosing pump."

→ New configuration consistent with existing project standards. Verify against existing patterns before accepting.

💡 Start every legacy session with the Audit prompt before making any changes. It gives the AI the context it needs to work consistently with existing patterns.

BEST PRACTICES

Writing Prompts That Work

These patterns reduce iteration and get you to a working result faster. They apply across all six pillars.

WHAT WORKS

Be specific about scope

Tell the AI exactly how many tags, which instruments, which folder. Vague scope produces incomplete results that need rework.

Reference existing patterns

If your project already has tags, say 'follow the naming convention already in use.' The AI reads the project and matches it.

Build on context

In the same session, the AI retains context from earlier steps. 'Use the tags you just created' works — you don't need to repeat the tag list.

State what you don't know

Write 'I'm not sure of the correct Modbus register format for this driver — suggest the standard approach.' The AI will ask or explain rather than guess.

Use the verify step

After every AI build step, check the result in Designer before continuing. Errors caught at the tag stage cost 5 minutes. Errors caught at the display stage cost an hour.

WHAT DOESN'T

Building everything from one prompt

One-prompt full builds are possible but produce more iteration. Pillar by pillar gives you checkpoints and reduces rework.

Skipping the verify step

The AI does not know if a tag value makes sense for your process. A flow meter reading 9999.9 at startup is a register addressing error, not a software bug.

Modifying AI-placed images

AI does not reliably place images on canvas displays. Place images manually in Designer first, then ask Claude to build around them.

Running sessions until degraded

If responses become less precise, start a new session, paste the starter prompt, and briefly re-state where you are in the build. Most issues resolve immediately.

SECTION 10

Known Limitations

AI Designer handles tags, devices, alarms, historian, scripts, and display structure reliably. These are the two areas that currently require manual work after AI generation.

Display visual polish

AI-generated displays are structurally correct — elements bind to the right tags, navigation works, alarms display. The visual layout, element sizing, alignment, and color consistency typically need manual refinement in Designer. Plan for 30 to 60 minutes of polish on a typical display after AI generation.

WORKAROUND

PLAN FOR IT

Build the structure with AI, then open the display in Designer and adjust sizing and alignment manually. The AI handles the binding; you handle the aesthetics.

Image placement on canvas displays

AI does not reliably place images on canvas displays. Images placed by AI often end up in incorrect positions or at unexpected scales, and asking the AI to correct a manually placed image may corrupt the placement.

WORKAROUND

MANUAL ONLY

Place all images manually in Designer first. Then ask Claude to build around them. Make all image changes manually.

Session length and context degradation

If a session runs long — typically after 60 to 90 minutes of work — responses may become less precise. The AI may start forgetting earlier context, producing results that don't match patterns established earlier.

WORKAROUND

EASY TO FIX

Start a new chat, paste the starter prompt, and briefly describe where you are: 'Tags and devices are complete. I am now configuring alarms.' Most issues resolve in one or two messages.

✓ These limitations are documented on docs.tatsoft.com and will be updated as capabilities improve. Check the AI Designer In Action page for the latest known issues.

CONTEXT

What to Expect From AI Designer

Engineers using FrameworX AI Designer report productivity improvements of 2x to 10x for configuration tasks. The range is wide because three variables matter.

2x – 10x

Productivity improvement for configuration tasks

40 – 70%

of project hours spent on mechanical config (before AI)

2 hours

Full pizza restaurant solution at ProveIT Conference, Dallas

THREE VARIABLES THAT DETERMINE WHERE YOU LAND IN THE 2X–10X RANGE

Project type

Lower end: Complex, non-repetitive control problems with lots of custom logic: 2x

Higher end: Configuration-heavy projects with repeating patterns (same device type × many instances): 8–10x

Platform familiarity

Lower end: Senior engineer who already knows the platform deeply: 2x — the AI fills gaps rather than replacing expertise

Higher end: Mid-level engineer learning the platform: much more — the AI carries platform knowledge while the engineer focuses on the process

AI collaboration skill

Lower end: First session, learning prompt structure, working through the verification loop: 2x

Higher end: After 2–3 projects: prompts are tighter, verification is faster, the AI's output needs less review

TATSOFT

Ready to build?

Download FrameworX AI Designer and run your first session today.

Free Download

tatsoft.com/fx-101/#download

Community Edition — no license required

AI Integration Docs

docs.tatsoft.com/display/FX/AI+Integration

Full technical documentation for all three services

MCP Setup Guide

docs.tatsoft.com/display/FX/MCP+and+Claude+Setup

Step-by-step Claude Desktop configuration

Tutorial Library

training.tatsoft.com/course/tutorials

Video walkthroughs including Module 8 AI tutorials

Questions? Join the community on Discord: https://discord.com/invite/BYhbTfyRyh