

Strategic Systems Engineering at Texaco

By Mark J. Hogan

Senior Architect | Infrastructure Strategist | Technical Mentor

Table of Contents

1.	Restoring Cross-System Telemetry Visibility in TDACS.....	1
2.	Engineering Resilience in a Fragile TCP/IP Stack	2
3.	GUI-Driven DNS Management Before Windows DNS	2
4.	Automating Domain Identity Management and Hygiene with Perl	2
5.	Architecting Secure Desktop Environments from Bare Metal	3
6.	Creating Mission Control: A Live Management Console for Domain Visibility and Workflow Automation	3
7.	Repurposing SCADA Interfaces for Unified Telemetry	4
8.	Final Reflection	Error! Bookmark not defined.

1. Restoring Cross-System Telemetry Visibility in TDACS

At Texaco’s Los Angeles refinery, cost-cutting measures led to the decommissioning of a Process Supervisory Computer (PSC)—a Data General MV system that had served as the central telemetry hub within the TDACS ecosystem. This PSC aggregated and redistributed flow values between Process Control Computers (PCC1, PCC2, PCC3), enabling downstream units to make informed operational decisions based on upstream data.

Its removal created a critical visibility gap. Without the PSC, units operated in isolation—unable to access real-time or historical telemetry from adjacent systems. This disruption threatened both operational efficiency and safety.

To solve this, I engineered a solution using TDLink—a custom TCP/IP service commissioned by Texaco to expose Get/Put APIs into the TDACS Commons. Originally intended for testing, the interface supported reading and writing “Other Variables” across systems. Verdi Wahjosoedibjo developed a utility that parsed command files to exercise these functions.

I repurposed that utility into a production-grade telemetry restoration engine: TDXFER. Running on a UNIX system, TDXFER orchestrated value transfers between PCC1, PCC2, and PCC3, restoring inter-unit visibility without reintroducing legacy infrastructure. It leveraged TDLink’s API to read from one PCC and write to another, using structured command files and scheduled execution.

This solution not only bridged the telemetry gap—it demonstrated how deep system knowledge, creative reuse of existing tools, and strategic engineering can unlock hidden potential in constrained environments.

2. Engineering Resilience in a Fragile TCP/IP Stack

When TDXFER came online to restore cross-unit telemetry, its success hinged on the reliability of the underlying communications layer. Unfortunately, Texaco's cost-saving choice—Claflin & Clayton's third-party TCP/IP stack for Data General systems—proved brittle. The software would silently degrade over time, severing telemetry links without warning and leaving operators blind to critical refinery data.

To mitigate this risk, I architected and built a Unix-based monitoring and recovery framework. It operated in two parts:

- A heartbeat generator continuously sent sinusoidal test values to each Process Control Computer (PCC1–3).
- A companion monitor watched for “flatline” behavior—an unmistakable sign of TCP/IP failure.

Upon detection, the system triggered an automated restart of the affected services and issued text alerts before and after recovery. This self-healing loop ran 24x7, restoring telemetry within seconds and eliminating the need for manual intervention.

The result: a fault-tolerant communications layer that quietly protected refinery operations from silent failure—proving that resilience isn't just about uptime, but about foresight.

3. GUI-Driven DNS Management Before Windows DNS

Before Microsoft introduced native DNS management tools, Texaco's infrastructure depended on Linux-based DNS services and NT 4.0 domains. Managing zone files required shell access and manual edits—an error-prone, opaque process that excluded many administrators from safely participating in domain hygiene.

To democratize DNS operations and reduce risk, I engineered a hybrid solution using Excel VBA and Unix/Windows scripting:

- The Excel workbook served as a visual interface, housing all A-records and IP mappings.
- Embedded VBA logic generated valid DNS configuration files directly from spreadsheet data.
- Shell scripts deployed the files to target servers and remotely restarted DNS services.

Administrators could initiate updates with intuitive button controls—bringing GUI simplicity to a traditionally command-line domain. This system enabled safe, repeatable DNS management across distributed NT environments, years ahead of native tooling.

It wasn't just a workaround—it was a strategic leap toward accessible infrastructure automation.

4. Automating Domain Identity Management and Hygiene with Perl

In the mid-1990s, Texaco's NT 4.0 domain infrastructure faced growing pains in identity management. Manual provisioning led to inconsistent naming conventions, misaligned group memberships, and unreliable home drive and roaming profile configurations—introducing risk across secure desktop environments and SCADA-integrated systems.

To address this, I architected a Perl-based automation suite that:

- Validated user attributes against standardized naming and role schemas
- Enforced correct group assignments and access controls

- Audited and verified existence of home/profile directories
- Auto-generated missing shares with appropriate permissions and mappings

This solution transformed domain hygiene from a reactive chore into a proactive system. It reduced administrative overhead, eliminated configuration drift, and enabled scalable, predictable identity management across distributed NT domains.

Years ahead of native tooling, this Perl framework laid the groundwork for resilient infrastructure and operational confidence—turning identity management into a strategic asset.

5. Architecting Secure Desktop Environments from Bare Metal

In the mid-1990s, I was tasked with migrating Texaco’s Los Angeles refinery desktops from Windows 3.11 to a 32-bit operating system. The options were limited: **Windows 95**, with its bootable FAT32 recovery flexibility but fragile architecture, or **Windows NT**, with its robust NTFS security model but unforgiving crash behavior. I chose NT—not for ease, but for long-term strategic control.

To avoid the chaos of unauthorized installs and the infamous Blue Screen of Death, I studied Microsoft’s guidance on **Zero Administration**, **C2-level security**, and **automated Windows installations**. A sister refinery had experimented with disk cloning via duplicators, but the process was brittle and hard to scale.

Together with Tom Hall, I pioneered an **unattended installation framework** for Windows NT. We introduced auto-logon, scripted build recipes, and network-based provisioning. The system would:

- Auto-install Windows NT
- Authenticate and pull build instructions from network shares
- Install assigned software packages
- Revert system drive permissions from “Admin Full / Everyone None”
- Grant precise access to folders, registry keys, DCOM objects, and system files
- Enable applications requiring full permissions to run under “User” roles via group-based ACLs

This architecture enforced **least privilege access** across the enterprise. No user operated with administrative rights, yet all critical applications ran seamlessly. Achieving this required deep registry analysis, file system redirection, and relentless trial and error.

The solution was deployed across multiple refineries—including Texaco Los Angeles, Bakersfield, Eldorado, Pembroke, Shell Wood River, Shell Norco, and others preparing for the Shell–Saudi Aramco merger.

This secure desktop model became the foundation for Texaco’s Managed Desktop Solution and later evolved into **IT Works**, a commercial platform under System Management Technologies, Inc. It proved that **security and usability can coexist—when engineered with precision and foresight**.

6. Creating Mission Control: A Live Management Console for Domain Visibility and Workflow Automation

To address the lack of real-time visibility into refinery users, computers, and domain groups, I architected Mission Control—a comprehensive management console powered by SQL Server that unified domain oversight, ticketing, and administrative workflows.

Mission Control featured:

- **Live object views** of Users, Computers, and Groups with contextual right-click actions
- **Custom menus** for internal and user-defined operations, enabling targeted actions across domain objects
- **Integrated ticketing system** with auto-generation and optional completion for routine tasks (e.g., Reset Password, Unlock Account)
- **Work Request tracking** for full auditability and accountability
- **Real-time updates** across all user and device records, ensuring synchronized visibility and control

This tool transformed administrative operations by embedding automation into everyday tasks while enforcing traceability. Even the most mundane actions—like unlocking an account—were logged, ticketed, and auditable.

Mission Control laid the groundwork for future management platforms by combining:

- **Visibility** into live domain activity
- **Automation** of repetitive and critical workflows
- **Governance** through structured ticketing and permission-aware actions





It empowered admins to act decisively without sacrificing control, clarity, or compliance—proving that strategic tooling can elevate even routine IT operations into scalable, supportable infrastructure.

7. Repurposing SCADA Interfaces for Unified Telemetry

At Texaco's Los Angeles refinery, I initially commissioned an IBM RS/6000 to monitor the Sulfur Recovery Unit (SRU) via Honeywell UxS stations. But I quickly recognized that AspenTech's CIM/21 SCADA system, originally scoped for a single unit, had untapped potential.

CIM/21 was designed to scan any CIMIO-compatible device—so I worked directly with AspenTech Support to configure CIMIO-TDLINK, enabling it to scan far beyond its original boundaries.

I rearchitected the system to unify telemetry across the entire refinery:

-  SRU Unit via **CIMIO-UxS**
-  Two Windows-based weather stations via CIMIO-DDE
-  Waste Water Treatment Unit via CIMIO-OPC via WonderWare OPC
-  Three TDACS PCCs via a custom CIMIO-TDLINK interface

This configuration allowed CIM/21 to scan another SCADA system (TDACS)—a capability never envisioned by its designers.

The result was a virtual Process Supervisory Computer (PSC) that aggregated telemetry from every unit into a single, accessible platform. For the first time, operators could view all values from all systems in one place, dramatically improving situational awareness, operational efficiency, and cross-unit coordination.

By repurposing existing tools and pushing their limits, we redefined what SCADA integration could look like—without replacing a single system.

8. Final Reflection: Engineering with Empathy, Strategy, and Precision

None of these solutions came from a manual. They were born from necessity, curiosity, and a refusal to accept the constraints of the moment. Whether restoring telemetry across decommissioned SCADA systems, automating fault recovery in brittle TCP/IP stacks, or reimagining desktop security from bare metal, each initiative reflects a mindset that blends strategic clarity with technical empathy.

What ties them together isn't just ingenuity—it's intention.

- **Deep System Understanding:** Every breakthrough began with a forensic-level grasp of the tools at hand. I didn't just use systems—I studied them, challenged their assumptions, and uncovered capabilities even their creators hadn't envisioned.
- **Strategic Reuse:** From repurposing TDLINK APIs to transforming Excel into a DNS management console, I consistently turned existing tools into unexpected solutions—proving that innovation doesn't always require new tech, just new thinking.
- **Operational Resilience:** Whether through self-healing telemetry loops or automated identity hygiene, I built systems that didn't just work—they endured. These weren't patches; they were frameworks for long-term stability.
- **Human-Centered Design:** Even in deeply technical domains, I prioritized accessibility. GUI-driven DNS, permission-aware desktop builds, and intuitive management consoles empowered teams to act confidently without sacrificing control.
- **Legacy as a Teaching Tool:** These stories aren't just war stories—they're blueprints. Each one is a lesson in how to think, how to build, and how to lead with both rigor and empathy.

This isn't just a retrospective—it's a call to preserve and share the engineering mindset that made these wins possible. Because the real legacy isn't the code or the config files—it's the clarity, the courage, and the commitment to building systems that serve people, not just machines.