

# Tight as a Drum<sup>®</sup> Software Change & Configuration Management



## Introducing the product

Tight as a Drum's suite of Object Management Software has matured into a robust Software Lifecycle Tool from a strong tradition of software change management practices on the iSeries platform. The brand name became Tight as a Drum in North America several years ago. Tight as a Drum is a registered trademark of Unbeaten Path International.

The software has a strong usage heritage in Europe as demonstrated by these **customer logos**. It was invented in The Netherlands and has a reputation for continuous improvement so as to leverage new technology in the marketplace. A recent manifestation of this investment is another **new release** which delivers a complete set of plug-ins for the Eclipse/WDSC/RAD environment.

## Foundational principles

You are invited to click [here](#) to read a white paper on Software Configuration & Change Management (SCCM) solutions. It describes important concepts worth considering before proceeding with the selection/implementation of a SCCM solution.

The Tight as a Drum product rests upon these three principles:

### *Intimate knowledge of the software configuration database*

The first irrefutable law of software change management is: "if a developer makes a change, he/she must have precise knowledge about the consequences for any other system components." Tight as a Drum's software configuration database automatically records and sustains a complete record of object relationships and version history for managed software components.

### *A defined software lifecycle*

The second irrefutable law is that process measurement and enhancement must necessarily begin with a precise definition of how the software lifecycle will be managed. Tight as a Drum functionality comfortably embraces the entire software change workflow: incident registration, requirements management, development, testing, ratification, distribution and implementation of software changes within production environments.

### *Reliable performance from people*

It doesn't matter how fancy the development tools are, process integrity rests upon reliable performance from people committed to follow the established workflow.



## Application of Tight as a Drum software

Using Tight as a Drum, you can map the dependencies between software components in a Software Configuration Database and then define a software life-cycle for each change episode. Typically the following five steps are used for every software change:

1. *Incident Management* ---> monitoring, when does a problem occur?
2. *Change Request Management* ---> problem analysis, what are the demands?
3. *Task Management* ---> who is doing what, when?
4. *Test Management* ---> who carries out which tests, and when?
5. *Deployment* ---> taking the software into productive use again

Above and beyond this basic functionality, customers can elect to apply a broad range of additional functionality: a fully integrated help desk which registers incidents and change requests, robust impact analysis, plug-ins that support development work on platforms like LANSa, and support for software deployment from the System I into UNIX and Windows environments.

Click [here](#) to see an eight page presentation about Tight as a Drum features.

## “Tight as a Drum” means strong internal controls

SOX auditors know all about software change vulnerabilities and Murphy’s Law. So, they’ll be asking questions about how that process is controlled at your enterprise. The “AS#2” interpretation of SOX authored by **PCAOB** demands that auditors test information technology controls because other business process controls are dependent upon them.

It doesn’t take much imagination to see how a poorly controlled software change process can introduce internal control problems. Consider this scenario:

*In January ‘09, contractors deliver a modification to your order processing system which includes new pricing functionality which is both subtle and sophisticated. Later that same year, the Customer Service Manager asks for a revision to the format of an order processing screen display.*

*A well-meaning programmer accidentally grabs a version of the order processing source code from 2008. The screen display change is executed in that out-of-date object. When the modified object is compiled, the screen display looks great but the contractors’ work has been wiped out. The subtle pricing errors aren’t looked for nor discovered until early 2010 when the CFO spots erosion in some gross profit margins.*

That genre of internal control issue demands attention whether or not your enterprise has compliance requirements under Sarbanes-Oxley/PCAOB, FDA 21 CFR part 11, HIPAA, or GLBA. (Easy-to-read executive summaries describing the alphabet soup of compliance standards are available [here](#).)

## Information technology: fertile ground for Murphy’s Law

Battle-hardened CIOs testify that each software change project invents at least one fresh, non-humorous corollary. Seasoned project managers know that the number of new corollaries tends to grow geometrically with project team size.

## Fertile ground for Murphy's Law, *continued ...*

Veteran technical managers at Unbeaten Path recall with clarity how challenging it was to manage project teams for large Y2K remediation projects. One project with +/- 4000 objects hit a snag when we couldn't quite deduce which library 16 repaired AS/SET objects originally came from. So, we had to do some of the remediation work a second time to get it perfect.

Even one-man software change projects tangle with Murphy's Law because there are so many interdependent steps: definition of user needs, conceptual design, detailed design, coding, testing, debugging, fixing, testing again, introduction to users, design refinement, more testing/debugging/fixing/testing again, user acceptance, documentation, installation, cut-over, and going live.

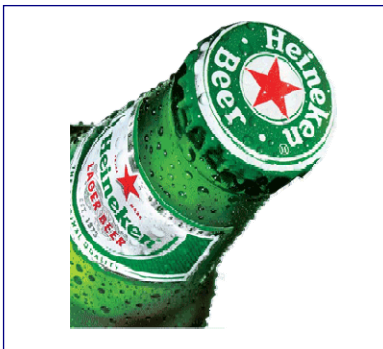
*Edward A. Murphy, Jr. was one of the engineers on the rocket-sled experiments that were done by the US Air Force in 1949 to test human acceleration tolerances. One experiment involved a set of 16 accelerometers mounted to different parts of the subject's body. There were two ways each sensor could be glued to its mount, and somebody methodically installed all 16 the wrong way around. Murphy then made the original form of his pronouncement, which the test subject (Major John Paul Stapp) quoted at a news conference a few days later:*

**"If there are two or more ways to do something,  
and one of those ways can result in a catastrophe,  
then someone will do it."**

Some humorous Murphy's Law corollaries for information technology are available [here](#).

## Immunity from Murphy's Law: how to maintain control

Click on these two customer case studies; they provide testimony that Tight as a Drum has helped them maintain strong software change & configuration management control:



## Questions ?

It would be a privilege to answer any questions about **Tight as a Drum** software. Here's Unbeaten Path International's contact information:

**Toll free North America: (888) 874-8008**

**International: (+USA) 262-681-3151**

Send us an email ( click [here](#) )

