



WHITEPAPER

Compliance: what it means for databases



Introduction

Compliance is the general term used to describe the efforts made by many (typically larger) organizations to meet regulatory standards.

In the US, the most important compliance regulations are the Sarbanes–Oxley Act of 2002 (SOX) for public companies, and the Statement on Auditing Standards No. 70: Service Organizations (SAS70) for private organizations.

Other compliance regulations include:

- The Health Insurance Portability and Accountability Act (HIPAA)
- The PCI Data Security Standard (PCI DSS)
- The Gramm–Leach–Bliley Act (GLB)
- The Federal Information Security Management Act of 2002 (FISMA)
- The International Organization for Standardization (ISO) standard ISO17799
- In the UK, the Financial Services Authority (FSA) also maintains regulatory requirements

The majority of this legislation deals with increasing accountability in the wake of some highly visible and damaging breaches of public trust. It gives rise to a need for more reliable paper trails, security and access controls, detailed and reliable monitoring, and change histories.

The impact on database professionals

Regulatory compliance has an impact throughout an organization, from finance departments and CIOs to individual developers and database administrators (DBAs). For example, compliance auditors will require DBAs – the custodians of a company's critical data – to account for all changes to a database, and detail all those with access to it.

This means they must be able to demonstrate they have a robust database change management process in place to:

- Manage how schema and data changes are made
- Document the changes that are made
- Maintain a detailed history of who made which changes
- Document database schema and access permissions
- Revert to previous versions if problems occur

Unfortunately, these are areas where database development has long lagged behind wider application development.

The nature of database code has historically been seen as a barrier to the implementation of change management, and therefore the maintenance of an audit trail. Many databases even go into production without any clear and meaningful history of changes, and auditors may regard this as an unacceptable avenue of risk.

An audit trail for SQL development

Being able to demonstrate a clear audit trail of changes made to your database is the first step in getting databases ready for compliance.

Introducing version control to the development process lets you know who changed what, when, and why. You get complete oversight of the changes that go into production and can more readily demonstrate that you comply with standards.

"Many organizations have to comply with legal requirements for change auditing, such as those mandated by Sarbanes-Oxley. Implementing a version control system could be the quickest and easiest way to provide the required level of historical tracking of all changes so that for every change to the database, you know who did it and when."

Grant Fritchey, Product Evangelist at Redgate Software & Microsoft MVP

With **SQL Source Control** you can look through the full revision history of a database or database object.

It integrates with existing version control systems so that you can check your database changes into the system you are already using for your application code. And because it plugs into SQL Server Management Studio, there is no need to change the way you and your team work.

This eases adoption, giving you a database development audit trail with minimal disruption.

Preparing for compliance with the SQL Toolbelt

Compliance requires strict process enforcement and information management. Although no single tool can solve an organization's compliance problems in their entirety, the SQL Toolbelt presents a package of solutions to some of the most painful and costly database development and deployment problems.

SQL Source Control allows you to preserve an incremental history of these changes, who made them, why, and when.

SQL Compare and **SQL Data Compare** can not only deploy these changes, but produce detailed reports of the differences between databases, and the final changes you deploy.

SQL Doc produces thorough schema documentation.

SQL Data Generator lets you test databases with realistic data in cases where sensitive production data cannot be used.

DLM Dashboard visualizes database schema changes as they move through different environments, detects unauthorized changes, and alerts you by email.

DLM Automation allows you to automate the build and deployment of the database code you check in with SQL Source Control. With database automation, you can put in place a reliable, repeatable, and predictable change management process.

You can download the **SQL Toolbelt**, with all the tools described in this paper, for a free 14 day trial here: www.red-gate.com/sql-toolbelt

