

Appendix 18 - The Grumpy Old IT Guy Master Data Services Framework (GITG-MDSF™)

A Practical Framework for Implementing and Governing Enterprise Master Data with Microsoft Master Data Services

Copyright © 2026 Grumpy Old IT Guy Publishers / Angelo R. Bobak

Introduction

Master Data Services (MDS) has long served as Microsoft's centralized platform for managing trusted enterprise master data. Although Microsoft has shifted its strategic focus toward Microsoft Fabric, Purview, and cloud-native governance solutions, thousands of organizations continue to operate successful MDS environments that remain critical to daily business operations.

The Grumpy Old IT Guy Master Data Services Framework (GITG-MDSF™) provides a structured methodology for designing, implementing, governing, and

maintaining enterprise master data using SQL Server Master Data Services.

Rather than viewing MDS simply as another database application, the framework positions MDS as the authoritative repository that governs enterprise business entities, hierarchies, business rules, workflows, and stewardship activities.

The framework integrates:

- SQL Server
- Master Data Services (MDS)
- SQL Server Integration Services (SSIS)
- Data Quality Services (DQS)
- SQL Server Agent
- Power BI
- SQL Server Reporting Services (SSRS)
- Microsoft Purview (where applicable)
- Enterprise Data Governance

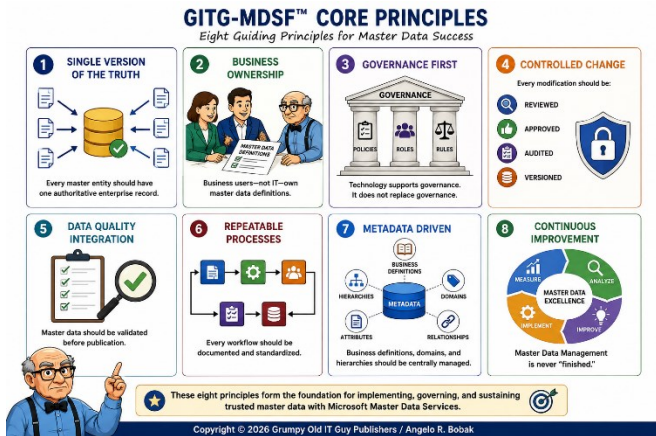
Objectives

The framework helps organizations:

- Create trusted Golden Records
- Eliminate duplicate master data
- Standardize enterprise business entities
- Improve data quality
- Support regulatory compliance
- Simplify downstream integration
- Enable enterprise reporting
- Establish long-term governance

Core Principles

The GITG-MDSF™ Framework is built upon eight guiding principles.



Principle 1 — Single Version of the Truth

Every master entity should have one authoritative enterprise record.

Principle 2 — Business Ownership

Business users—not IT—own master data definitions.

Principle 3 — Governance First

Technology supports governance.

It does not replace governance.

Principle 4 — Controlled Change

Every modification should be:

- reviewed
 - approved
 - audited
 - versioned
-

Principle 5 — Data Quality Integration

Master data should be validated before publication.

Principle 6 — Repeatable Processes

Every workflow should be documented and standardized.

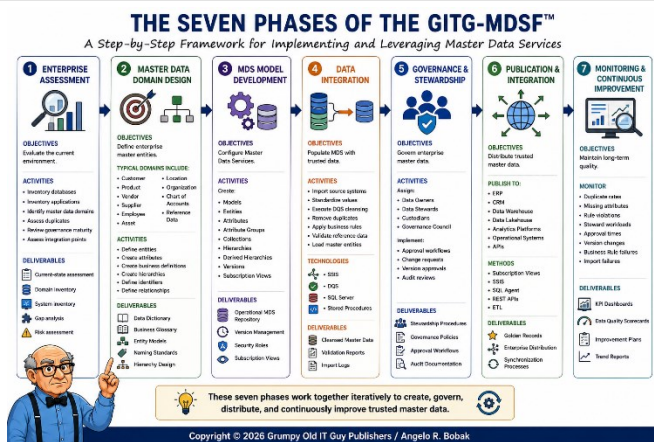
Principle 7 — Metadata Driven

Business definitions, domains, and hierarchies should be centrally managed.

Principle 8 — Continuous Improvement

Master Data Management is never "finished."

The Seven Phases of the GITG-MDSF™



The 7 Phases of Master Data Management with SQL Server MDS

Phase 1 — Enterprise Assessment

Objectives

Evaluate the current environment.

Activities

- Inventory databases
- Inventory applications
- Identify master data domains

- Assess duplicates
- Review governance maturity
- Assess integration points

Deliverables

- Current-state assessment
 - Domain inventory
 - System inventory
 - Gap analysis
 - Risk assessment
-

Phase 2 — Master Data Domain Design

Objectives

Define enterprise master entities.

Typical domains include:

- Customer
- Product
- Vendor
- Supplier
- Employee
- Asset
- Location
- Organization
- Chart of Accounts
- Reference Data

Activities

- Define entities
- Create attributes
- Create business definitions
- Create hierarchies
- Define identifiers
- Define relationships

Deliverables

- Data Dictionary
- Business Glossary
- Entity Models
- Naming Standards
- Hierarchy Design

Phase 3 — MDS Model Development

Objectives

Configure Master Data Services.

Activities

Create:

- Models
- Entities
- Attributes
- Attribute Groups
- Collections
- Hierarchies
- Derived Hierarchies
- Versions

- Subscription Views

Deliverables

- Operational MDS Repository
 - Version Management
 - Security Roles
 - Subscription Views
-

Phase 4 — Data Integration

Objectives

Populate MDS with trusted data.

Activities

- Import source systems
- Standardize values
- Execute DQS cleansing
- Remove duplicates
- Apply business rules
- Validate reference data
- Load master entities

Technologies

- SSIS
- DQS
- SQL Server
- Stored Procedures

Deliverables

- Cleansed Master Data
 - Validation Reports
 - Import Logs
-

Phase 5 — Governance and Stewardship

Objectives

Govern enterprise master data.

Activities

Assign:

- Data Owners
- Data Stewards
- Custodians
- Governance Council

Implement:

- Approval workflows
- Change requests
- Version approvals
- Audit reviews

Deliverables

- Stewardship Procedures
- Governance Policies
- Approval Workflows
- Audit Documentation

Phase 6 — Publication and Integration

Objectives

Distribute trusted master data.

Activities

Publish to:

- ERP
- CRM
- Data Warehouse
- Data Lakehouse
- Analytics Platforms
- Operational Systems
- APIs

Methods

- Subscription Views
- SSIS
- SQL Agent
- REST APIs
- ETL

Deliverables

- Golden Records
- Enterprise Distribution
- Synchronization Processes

Phase 7 — Monitoring and Continuous Improvement

Objectives

Maintain long-term quality.

Monitor

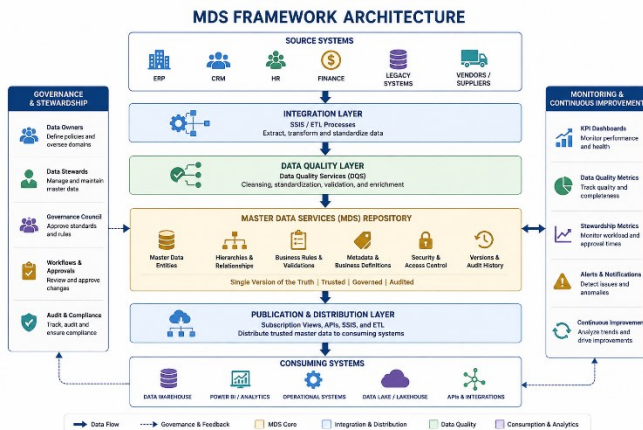
- Duplicate rates
- Missing attributes
- Rule violations
- Steward workloads
- Approval times
- Version changes
- Business Rule failures
- Import failures

Deliverables

- KPI Dashboards
 - Data Quality Scorecards
 - Improvement Plans
 - Trend Reports
-

Framework Architecture

The following architecture illustrates how enterprise master data flows through the **Grumpy Old IT Guy Master Data Services Framework (GITG-MDSF™)** to create trusted, governed, and reusable master data across the organization.



SQL Server Master Data Services Architecture

Rather than allowing every application to maintain its own version of critical business information, the architecture establishes a centralized Master Data Services (MDS) repository that serves as the authoritative source for enterprise master data.

The process begins with the organization's **enterprise source systems**, which typically include Enterprise Resource Planning (ERP), Customer Relationship Management (CRM), Human Resources (HR), Finance, legacy applications, and external vendor systems. Each of these systems often stores overlapping information about customers, suppliers, products, employees, locations, and

other master data entities. Because these systems are developed independently over many years, inconsistencies, duplicate records, missing values, and conflicting business definitions frequently arise.

Data from these operational systems is extracted through the **SQL Server Integration Services (SSIS) Integration Layer**, which provides the mechanisms for importing, transforming, and staging information before it enters the enterprise master data environment. SSIS orchestrates scheduled data loads, applies transformation logic, performs data mapping, and prepares incoming records for quality validation.

Once data has been integrated, it passes through **SQL Server Data Quality Services (DQS)**. This layer improves the reliability and consistency of enterprise information by applying standardized business rules, reference data, validation routines, domain knowledge, and data cleansing processes. DQS identifies duplicate records, standardizes attribute values, corrects formatting inconsistencies, validates reference values, and enriches information before it becomes part of the enterprise master data repository.

The cleansed and standardized information is then loaded into the **Master Data Services (MDS) Repository**, which serves as the organization's centralized system of record for master data. MDS manages enterprise entities such as customers, products, suppliers, employees, locations, and reference data while maintaining complete control over versions, hierarchies, collections, and metadata. By centralizing these business entities, MDS establishes a **single version of the truth** that can be trusted throughout the enterprise.

Within the MDS repository, several governance capabilities ensure that master data remains accurate and well controlled. **Business rules and validation** enforce organizational standards and prevent invalid or incomplete information from being published. **Hierarchies and version management** organize business entities into meaningful structures while preserving historical versions for auditing and change management. **Security and stewardship** provide role-based access, allowing authorized data stewards to review, approve, and manage changes while maintaining complete auditability of all modifications.

After master data has been reviewed and approved, it is distributed through **Subscription Views, APIs, and ETL processes**. These mechanisms enable downstream applications to consume consistent, trusted master data without maintaining independent copies or manually synchronizing information. Whether data is delivered through SQL Server subscription views, integration services, REST APIs, or scheduled ETL processes, every consuming system receives standardized and governed enterprise data.

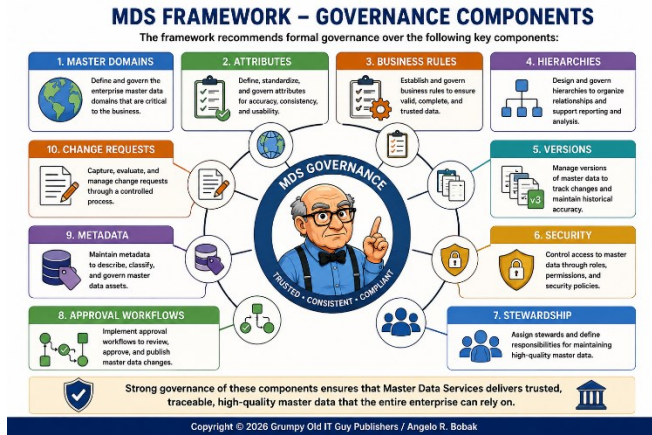
The final stage of the architecture supports a wide variety of business consumers. **Enterprise Data Warehouses** leverage trusted master data to improve reporting consistency and dimensional modeling. **Power BI** dashboards and analytics platforms use standardized information to generate accurate business intelligence, scorecards, and executive reports. **Operational systems** receive synchronized master records that improve transaction processing, reduce duplicate data entry, and ensure consistent business operations across the enterprise.

This layered architecture transforms isolated operational data into trusted enterprise information through a disciplined process of integration, standardization, validation, governance, and controlled distribution. By combining SSIS, **Data Quality Services**, and **Master Data Services** within a unified SQL Server ecosystem, organizations can significantly improve data quality, strengthen governance, simplify integration, and establish a sustainable foundation for enterprise master data management.

At the heart of the architecture is a simple but powerful principle: **enterprise systems should not compete to define master data—they should consume it from a single, governed, and trusted source.** The GITG-MDSF™ provides the practical framework for achieving that objective, enabling organizations to improve operational efficiency, enhance reporting accuracy, support regulatory compliance, and build lasting confidence in their enterprise information assets.

Governance Components

Here are the objects that require governance:



MDS Framework Governance components

The framework recommends formal governance over:

- Master Domains
- Attributes
- Business Rules
- Hierarchies
- Versions
- Security
- Stewardship
- Approval Workflows
- Metadata
- Change Requests

Recommended Roles

Role	Responsibilities
Executive Sponsor	Strategic oversight
Data Governance Council	Policy approval
Data Owner	Business ownership
Data Steward	Data maintenance
MDS Administrator	Platform administration
SQL Server DBA	Infrastructure
SSIS Developer	Data integration
Data Quality Analyst	Validation
Business SME	Domain expertise
Auditor	Compliance verification

This is the minimal set of roles that are required for a successful master data governance framework. Executive sponsorship, funding, and ongoing support are required, or the implementation will fail.

Key Performance Indicators

The GITG-MDSF™ recommends tracking:

- Duplicate Master Records
- Golden Record Accuracy
- Steward Approval Time
- Business Compliance Rule
- Import Success Rate
- Hierarchy Completeness
- Metadata Completeness
- Attribute Completeness
- Version Publication Success
- Data Synchronization Latency

Success Factors

Successful implementations typically include:

- Executive sponsorship
- Strong governance
- Business stewardship
- Well-defined business rules
- Reliable integration architecture
- Comprehensive metadata
- User training
- Continuous monitoring

Common Pitfalls

Organizations frequently encounter:

- Treating MDS as merely another database
 - Lack of business ownership
 - Poorly defined master domains
 - Weak governance processes
 - Excessive customization
 - Inadequate stewardship
 - Ignoring data quality before loading
 - Publishing unapproved master records
 - Failing to monitor KPIs
 - Poor change management
-

Relationship to Other GITG™ Frameworks

The GITG-MDSF™ is designed to work alongside the other frameworks in the Grumpy Old IT Guy ecosystem:

- Appendix 8 - GITG-APF Grumpy Old IT Guy AI Profiling Framework
- Appendix 9 - GITG-DCF Grumpy Old IT Guy Data Cleansing Framework
- Appendix 10 - GITG-CDSF Grumpy Old IT Guy Customer Data Standardization Framework
- Appendix 11 - GITG-RMF Grumpy Old IT Guy Record Matching Framework
- Appendix 12 - GITG-FMF Grumpy Old IT Guy Fuzzy Matching Framework
- Appendix 13 - GITG-SMAF- SQL Server Matching Architecture Framework
- Appendix 14 - GITG-MMF- SQL Server Matching Monitoring Framework
- Appendix 15 - GITG-MDF- Master Data Framework
- Appendix 16 - GITG-MDF- Master Data Foundation Framework
- Appendix 17 - GITG-RDF- Reference Data Framework

Together, these frameworks provide a comprehensive methodology for designing, implementing, governing, and sustaining trusted enterprise data assets.

The Grumpy Old IT Guy Rule

"Master Data Services doesn't create trusted master data—people, governance, and disciplined processes do. MDS simply gives them a place to live."

Successful Master Data Management is not achieved by installing software. It is achieved by establishing governance, enforcing standards, empowering data stewards, and continuously improving the quality of enterprise information. The **GITG-MDSF™** provides the practical roadmap for turning SQL Server Master Data Services into a governed, trusted, and sustainable enterprise master data platform.

© 2026 Angelo R. Bobak
All rights reserved.

First Edition: February 2026

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means—electronic, mechanical, photocopying, recording, or otherwise—without the prior written permission of the publisher.

Grumpy Old IT Guy Publishers
Westchester County, NY, USA
www.grumpyolditguy.com