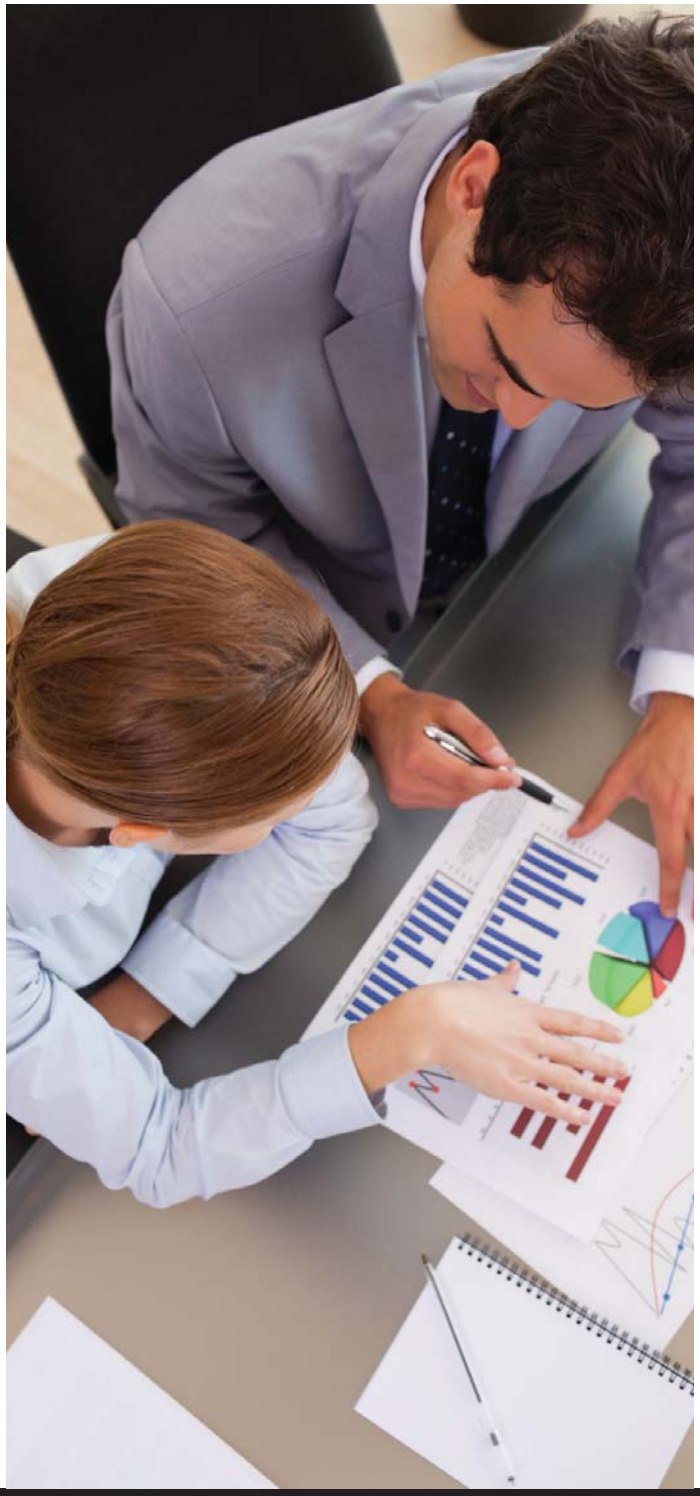
Creating a Master Data Environment



An Incremental Roadmap for Public Sector Organizations

Master Data Management (MDM) is the processes and technologies used to create and maintain consistent and accurate representations of master data.

Movements toward modularity, service orientation, and SaaS make Master Data Management a critical issue. Master Data Management leverages both tools and processes that enable the linkage of critical data through a unified platform that provides a common point of reference.

When properly done, MDM streamlines data management and sharing across the enterprise among different areas to provide more effective service delivery.

CMA possesses more than 15 years of experience in Master Data Management and more than 20 in data management and integration. CMA has also focused our efforts on public sector since our inception, more than 30 years ago.

This document describes the fundamental keys to success for developing and maintaining a long term master data program within a public service organization.

Understanding Master Data

Master Data Behavior

Master data can be described by the way that it interacts with other data. For example, in transaction systems, master data is almost always involved with transactional data. This relationship between master data and transactional data may be fundamentally viewed as a noun/verb relationship. Transactional data captures the verbs, such as open, approve, close, send, and submit; master data are the nouns. This is the same relationship between data warehouse facts and dimensions.

Master Data Life Cycle

Master data can be described by the way that it is created, read, updated, deleted, and searched. This life cycle is called the CRUD cycle and is different for different master-data element types. As cardinality (the number of elements in a set) decreases, the likelihood of an element being treated as a master-data element—even a commonly accepted subject area, such as case, decreases. Master data tends to be less volatile than transactional data. As it becomes more volatile, it typically is considered more transactional. Simple entities are rarely a challenge to manage and are rarely considered master-data elements. The less complex an element, the less likely the need to manage change for that element. The more valuable the data element is to the organization, the more likely it will be considered a master data element. Value and complexity work together. While master data is typically less volatile than transactional data, entities with attributes that do not change at all typically do not require a master data solution.

One of the primary drivers of master data management is reuse. In today's complex environments, a person's information needs to be shared across multiple applications. Because access to a master database is not always available, people start storing master data in various locations, such as spreadsheets and application private stores. If a master data entity is reused in multiple systems, it should be managed within a master data management system.

Master Data Defined: The following primary data classifications will be used throughout this document.

Unstructured Data. Data found in e-mails, white papers, magazine articles, etc.

Transactional Data. Data related to cases, claims, tickets, workflows, etc.

Metadata. Simply "data about other data". Resides in file and database formats.

Hierarchical Data. Data on relationships between other data elements.

Master Data. Master data are the critical nouns of a business and fall generally into four groupings: people, things, places, and concepts. Further categorizations within those groupings are called subject areas, domain areas, or entity types. The granularity of domains is essentially determined by the magnitude of differences between the attributes of the entities within them.

Other Reasons to Establish a Master Data Management Program

Beyond the complexities involved in reusing data, there are other smart reasons to establish a Master Data Management program.

Because it is used by multiple applications, an error in master data can cause errors in all the applications that use it. Even if the master data has no errors, few organizations have just one set of master data. Merging master lists together can be very difficult. The same person may have different names, numbers, addresses, and phone numbers in different databases. Normal database joins and searches will not be able to resolve these differences. A very sophisticated tool that understands nicknames, alternate spellings, and typing errors will be required. The tool will probably also have to recognize that different name variations can be resolved, if they all live at the same address or have the same phone number.

Movements toward modularity, service orientation, and SaaS also make Master Data Management a critical issue. For example, if you create a single service that communicates through well-defined XML messages, you may think you have defined a single view of your constituents. But if the same constituent is stored in five databases with three different addresses and four different phone numbers, what will your constituent service return?

Maintaining a high-quality, consistent set of master data for any organization is a necessity. The processes (and systems) required to maintain this data are collectively known as Master Data Management.

What is Master Data Management

MDM is not just a technological problem. Fundamental changes to business process will be required to maintain clean master data, and some of the most difficult MDM issues are more political than technical. MDM includes both creating and maintaining master data. Investing a lot of time, money, and effort in creating a clean, consistent set of master data is a wasted effort unless the solution includes tools and processes to keep the master data clean and consistent as it is updated and expanded.

While MDM is most effective when applied to all the master data in an organization, in many cases the risk and expense of an enterprise-wide effort are difficult to justify. It may be easier to start with a few key sources of master data and expand the effort, once success has been demonstrated and lessons have been learned. Most MDM projects include the following key phases:

- Identify sources of master data.
- Identify the producers and consumers of the master data.
- Collect and analyze metadata about for your master data.
- Appoint data stewards.
- Implement a data-governance program and data-governance council.
- Develop the master-data model.
- Choose a toolset.
- Design the infrastructure.
- Generate and test the master data.
- Modify the producing and consuming systems.
- Implement the maintenance processes.

MDM is a complex and time consuming process. The key to success is to implement MDM incrementally, so that the organization realizes a series of short-term benefits while the complete project is a long-term process. No MDM project can be successful without the support and participation of the business users. IT professionals do not have the domain knowledge to create and maintain high-quality master data. Any MDM project that does not include changes to the processes that create, maintain, and validate master data is likely to fail.

Establishing a Master Data Environment

There are two basic steps to creating master data: clean and standardize the data, and match data from all the sources to consolidate duplicates. Before you can start cleaning and normalizing your data, you must understand the data model for the master data. As part of the modeling process, the contents of each attribute were defined, and a mapping was defined from each source system to the master data model. This information is used to define the transformations necessary to clean your source data.

Cleaning the data and transforming it into the master data model is basically an Extract, Transform, and Load (ETL) processes used to populate a data warehouse. If you already have ETL tools and transformation defined, it might be easier just to modify these as required for the master data, instead of learning a new tool. Typical ETL data-cleansing functions include:

- Normalize data formats. Make all the phone numbers look the same, transform addresses, etc.
- Replace missing values. Insert defaults, look up ZIP codes from the address, etc.
- Standardize values. Convert all date and time stamps, convert mixed monetary forms, etc.
- Map attributes. Parse the first name and last name out of a combined field, merge notes fields, etc.

Most tools will cleanse the data that they can, and put the rest into an error table for hand processing. Depending on how the matching tool works, the cleansed data will be put into a master table or a series of staging tables. As each source is cleansed, the output should be examined to ensure the cleansing process is working correctly.

Matching master-data records to eliminate duplicates is both the hardest and most important step in creating master data. False matches can actually lose data and missed matches reduce the value of maintaining a common list. The matching accuracy of MDM tools is one of the most important selection criteria.

Matching algorithms are normally very complex and sophisticated. The more attribute matches and the closer the match, the higher degree of confidence the MDM system has in the match. This confidence factor is computed for each match, and if it surpasses a threshold, the records match. The threshold is normally adjusted depending on the consequences of a false match. For example, you might specify that if the confidence level is over 95 percent, the records are merged automatically, and if the confidence is between 80 percent and 95 percent, a data steward should approve the match before they are merged.

Most merge tools merge one set of input into the master list, so the best procedure is to start the list with the data in which you have the most confidence, and then merge the other sources in one at a time. If you have a lot of data and a lot of problems with it, this process can take a long time. You might want to start

with the data from which you expect to get the most benefit having consolidated; run a pilot project with that data, to ensure your processes work and you are seeing the business benefits you expect; and then start adding other sources, as time and resources permit. This approach means your project will take longer and possibly cost more, but the risk is lower. This approach also lets you start with a few organizations and add more as the project demonstrates success, instead of trying to get everybody on board from the start.

Another factor to consider when merging your source data into the master list is privacy. When constituents become part of the constituent master, their information might be visible to any of the applications that have access to the constituent master. If the constituent data was obtained under a privacy policy that limited its use to a particular application, you might not be able to merge it into the constituent master.

Maintaining a Master Data Environment

Single Copy

In this approach, there is only one master copy of the master data. All additions and changes are made directly to the master data. All applications that use master data are rewritten to use the new data instead of their current data. While this approach guarantees consistency of the master data, in many cases it's not practical. Modifying all your applications to use a new data source with a different schema and different data is expensive. If any of your applications have proprietary data bases, it might even be impossible.

Multiple Copies, Single Maintenance

In this approach, master data is added or changed in the single master copy of the data, but changes are sent out to the source systems in which copies are stored locally. Each application can update the parts of the data that are not part of the master data, but they cannot change or add master data. This reduces the number of application changes that will be required, but the applications will minimally have to disable functions that add or update master data. Users will have to learn new applications to add or modify master data, and some of the things they normally do will not work anymore.

Continuous Merge

In this approach, applications are allowed to change their copy of the master data. Changes made to the source data are sent to the master, where they are merged into the master list. The changes to the master are then sent to the source systems and applied to the local copies. This approach requires few changes to the source systems; if necessary, the change propagation can be handled in the database, so no application code is changed. On the surface, this seems like the ideal solution. Application changes are minimized, and no retraining is required. Everybody keeps doing what they are doing, but with higher-quality, more complete data. This approach does have several issues:

- Update conflicts are possible and difficult to reconcile. What happens if two of the source systems change a constituent's address to different values? There's no way for the MDM system to decide which one to keep, so intervention by the data steward is required; in the meantime, the constituent has two different addresses. This must be addressed by creating data-governance rules and standard operating procedures, to ensure that update conflicts are reduced or eliminated.
- Additions must be remerged. When a constituent is added, there is a chance that another system has

already added the constituent. To deal with this situation, all data additions must go through the matching process again to prevent new duplicates in the master.
- Maintaining consistent values is more difficult.

No matter how you manage your master data, it's important to be able to understand how the data got to the current state. For example, if a constituent record was consolidated from two different merged records, you might need to know what the original records looked like, in case a data steward determines that the records were merged by mistake and really should be two different constituents. Version management should include a simple interface for displaying versions and reverting all or part of a change to a previous version. The normal branching of versions and grouping of changes that source-control systems use can also be very useful for maintaining different derivation changes and reverting groups of changes to a previous branch.

Data stewardship and compliance requirements will often include a way to determine who made each change and when it was made. To support these requirements, an MDM system should include a facility for auditing changes to the master data. In addition to keeping an audit log, the MDM system should include a simple way to find the particular change you are looking for. An MDM system can audit thousands of changes a day, so search and reporting facilities for the audit log are important.

CMA's Phase-Based Approach

Few organizations have the resources to move to a complete enterprise MDM environment in one step. The final phases and their sequence of implementation are impacted by the State's business priorities, budget, pre-existing IT infrastructure, and key skills of your current IT organization.

Based on CMA's experience across a wide array of public sector clients, the following phased based implementation approach to enterprise level MDM is recommended.

Phase 1 – Synchronize Master Data across all current system of data entry via Data Integration Hub (DIH).
Install and configure an integration hub to propagate master data changes between systems of entry to keep the master current across systems. Establish an identity registry in the hub that connects related master records in each of the systems of entry. The registry is used to create a virtual master data system of record. In addition to launching master data governance policies and procedures work groups, this phase focuses on configuring, deploying and operationalizing the organization's data integration hub architecture.

Phase 2 - Maintain a master data store as a separate system of record.
Extends Phase 1 by now maintaining a master data store within the integration hub. The hub's master data store acts as the current system of record. The integration hub uses master data services to do tasks such as conflict resolution, and to reconcile and merge master data records. The master data store is also used in Phase 4 to propagate reconciled historical master data to a data warehouse.

Phase 3 - Migrate business transactions to enterprise MDM.
Extends Phase 2 by exposing master data services in the integration hub through documented interfaces. These interfaces enable existing business transaction

applications to be modified to maintain master data in the hub, rather than in local data stores. The interfaces can also be used by new custom-built and packaged master data management applications to manage master data and master metadata in the hub. Establishes a MD "workbench" that allows administrators and business users to interact with the integration hub and its master data store.

Phase 4 - Consolidate Master Data in a Data Warehouse for analytical processing.
As master data changes in business transaction applications, the changes are consolidated in a data warehouse, which maintains a historical record of the master data. The historical master data in the warehouse reflects the MD system of record at different moments in time. As in any data warehousing environment, the historical master and transaction data in the warehouse can be used for BI analytical processing.

Business Objectives

One of the primary drivers of Master Data Management is reuse. In today's complex environments, a person's information needs to be shared across multiple applications. Because access to a master database is not always available, people start storing master data in various locations, such as spreadsheets and application private stores. If a master data entity is reused in multiple systems, it should be managed within a master data management system.

Here are some examples of what leveraging a well-maintained mastered data environment can do.

- Online Transaction Systems: Shape a verification and validation service that supports synchronous and asynchronous transaction processing
- Reporting Platforms: Create comprehensive views into all key facets of a single entity record (like a person, family, or provider organization)
- Analytics Platforms: Create higher quality visualizations of patterns (versus those derived from un-mastered source data systems), leading to better decision support

Critical Issues

Critical issues in master data management include:

- Errors & Risk: Because it is used by multiple applications, an error in master data can cause errors in all the applications that use it.
- Merging Lists: Few organizations have just one set of master data, and merging them together can be difficult. Normal database joins and searches are not able to resolve differences. A very sophisticated tool is required.
- Privacy: When subsets of data join the master data, the information might be visible to any of the applications that have access to the master. If parts of that data were obtained under a limited privacy policy, then merging with the master data becomes problematic. Data stewardship and compliance mean the necessity of a facility for auditing changes to the master data.
- Investment: Investing a lot of time, money, and effort in creating a clean, consistent set of master data is a wasted effort unless the solution includes tools and processes to keep the master data clean and consistent as it is updated and expanded.

However, Master Data Management is not just a technological problem. Fundamental changes to business process are required to maintain clean master data.

Conclusion

A phased approach, like the one detailed in this paper, is the best way to approach Master Data Management in the Public Sector. Alignment of the technological and business processes involved require it.

CMA has extensive experience in the development, maintenance, and operations of mastered data environments, as well as the public sector. If you would like to discuss our approach, please give us a call.

518-783-9003
info@cma.com

About CMA

At CMA, we believe in building something bigger than ourselves, every day. We support the missions of our client partners and build tailored business solutions that are efficient and value-based.

CMA has provided information technology (IT) products and services since 1984. We serve commercial, industrial, and public sector industries. CMA has offices around the country with corporate headquarters in New York's Tech Valley.

CMA creates solutions that help our clients manage their business more effectively. CMA provides its customers the best in proven technology and experienced professionals.

With more than 450 employees, CMA has conducted thousands of technology-oriented consulting engagements and developed hundreds of application system solutions for our customers.

CMA is a New York State certified woman-owned business.