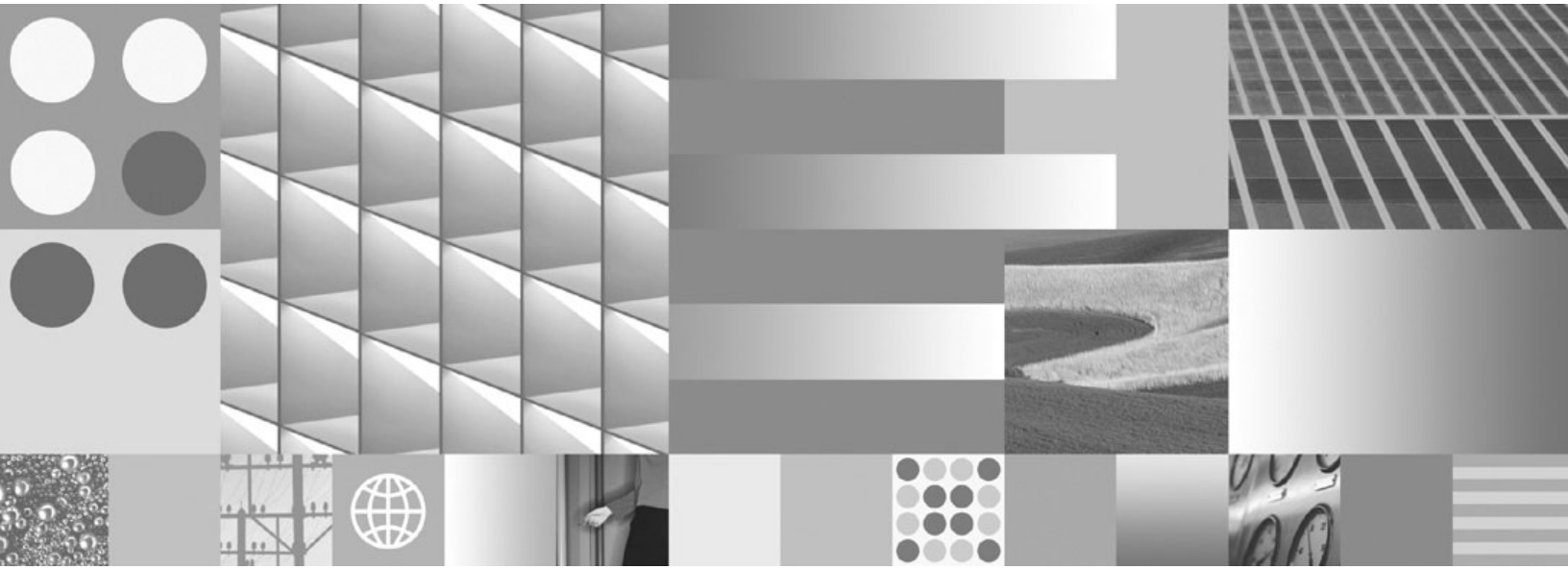
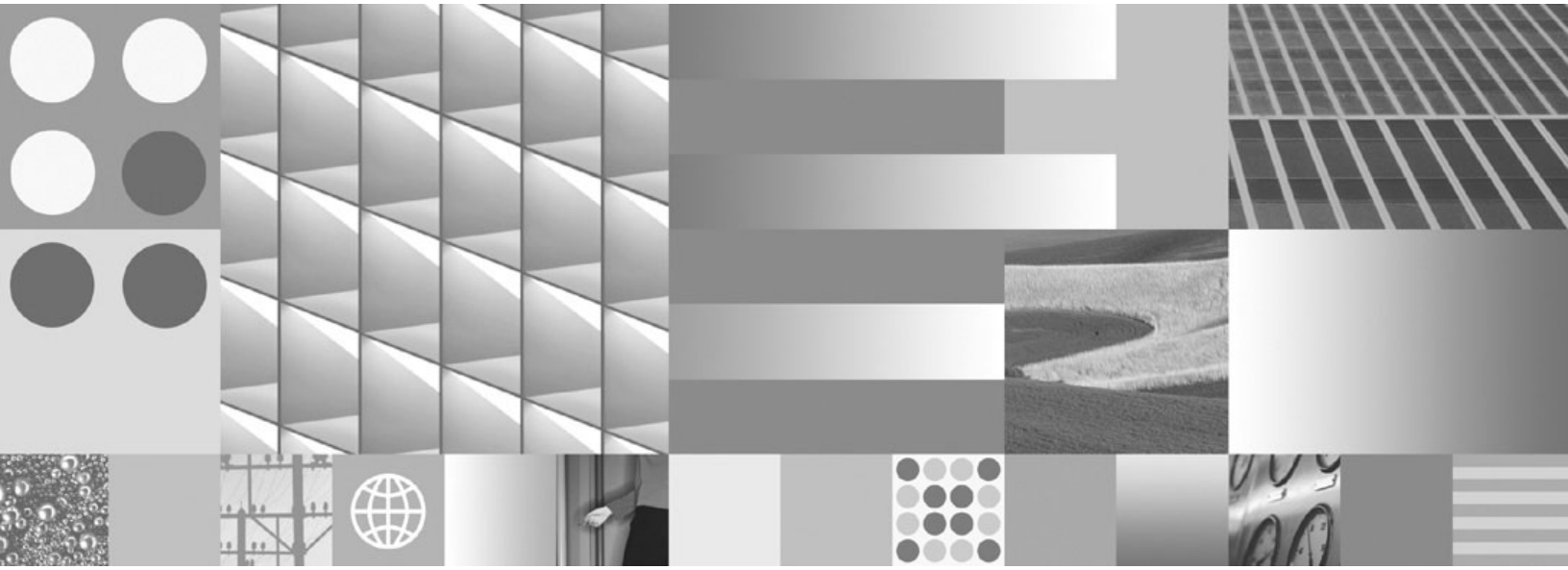
IBM InfoSphere Master Data Management Server



InfoSphere Master Data Management Server Version 8.5.0
Understanding and Planning Guide

IBM InfoSphere Master Data Management Server



InfoSphere Master Data Management Server Version 8.5.0
Understanding and Planning Guide

Note
Before using this information and the product it supports, read the general information under Appendix A, "Notices," on page 85.

Edition Notice

This edition applies to version 8.5.0 of IBM InfoSphere Master Data Management Server and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

Chapter 1. Understanding and planning to use IBM InfoSphere Master Data Management Server features

This guide contains information to help you understand IBM® InfoSphere™ Master Data Management Server (InfoSphere MDM Server) features, so you can decide which features you are going to use, and then create a plan to configure and implement those features.

The information for each feature includes:

- A description of the feature
- An example of how it can be used

For some features, the following information is also provided:

- Information on the behavior of the feature when it is configured on, when it is configured off, and when the configuration of the feature is changed in a production environment
- Whether the feature must be configured during installation
- Links to more information about how to modify the feature

Some InfoSphere MDM Server features must be configured **on** when InfoSphere MDM Server is being installed in order to be available at a later point, while other features can be turned on when you are ready to use them.

Other features cannot be configured on or off, but are always available in the installed InfoSphere MDM Server product.

For a list of the features that are new for this release of IBM InfoSphere Master Data Management Server, refer to the Release Notes.

Planning to modify IBM InfoSphere Master Data Management Server features

Modifying IBM InfoSphere Master Data Management Server features allows you to add more functionality to existing features and to scale up features to process larger volumes of data. This section includes information on how to plan for adding more functionality to features. For additional information, see the *Modifying IBM InfoSphere Master Data Management Server* section in the *IBM InfoSphere Master Data Management Server Developers Guide*.

InfoSphere MDM Server domains

IBM InfoSphere Master Data Management Server enables companies to extract maximum value from master data by centralizing multiple data domains and providing a comprehensive set of prebuilt business services that support a full range of master data management (MDM) functionality.

InfoSphere MDM Server integrates data from different domains using business services that interact with all applications and business processes that consume master data. These domains are:

Party Domain
The Party domain manages the entirety of data related to parties such as customers, vendors and suppliers, and it maintains a single, consistent version of this data.

Product Domain
The Product domain manages the definition of products. Its collection of products makes up a product catalog that is accessible to other systems across the enterprise.

Account Domain
The Account domain is an operational-styled hub that manages account data.

The majority of domain features are always available in the installed InfoSphere MDM Server product. You cannot configure them on or off. Configuration behavior information is not provided for such features because it is not applicable.

Related concepts

Chapter 3, "The Party domain," on page 55
The Party domain manages the entirety of data related to parties such as customers, vendors and suppliers, and maintains a single, consistent version of this data.

Chapter 4, "The Product domain," on page 69
The Product domain is an operational-styled hub that manages the definition of products.

Chapter 5, "The Account domain," on page 77

Understanding InfoSphere MDM Server consumers

InfoSphere MDM Server consumers are the methods that invoke InfoSphere MDM Server services.

For more information about InfoSphere MDM Server consumers, see the *IBM InfoSphere Master Data Management Server Developers Guide*.

The following sections provide understanding and planning information for InfoSphere MDM Server consumers.

- "Batch processing using MDMBatch" on page 11
- "Web Services" on page 20

Understanding the InfoSphere MDM Server Request Framework

The InfoSphere MDM Server Request Framework provides a consistent entry point to InfoSphere MDM Server and is used to receive requests and issues responses in any format.

For more information about the InfoSphere MDM Server Request Framework, see the *IBM InfoSphere Master Data Management Server Developers Guide*.

Related concepts

"Composite transactions" on page 9

"Request/Response Framework" on page 8

Chapter 2. InfoSphere MDM Server platform

IBM InfoSphere Master Data Management Server (InfoSphere MDM Server) is an enterprise application that provides the single version of truth for party, product and account master data, and an environment that processes updates to and from multiple channels, including databases. It aligns these front office systems with multiple back office systems in real time, providing a single source of customer truth. InfoSphere MDM Server uses a component-based Extensible Markup Language (XML) and Java™ 2 Platform, Enterprise Edition (J2EE) with full Enterprise Java Bean (EJB) architecture to rapidly integrate with other systems and deliver flexibility and scalability.

InfoSphere MDM Server can either be used in its standard configuration, or modified through customization. You can customize InfoSphere MDM Server through a number of externalized features—accessible to users—that control its operation.

InfoSphere MDM Server platform technical features

The following InfoSphere MDM Server platform features are mainly aimed at a technical audience.

Error handling and logging

Through logging application and system errors, you can:

- define errors and how they are logged
- customize error messages so that they are more meaningful to end-users than the error messages that are supplied with IBM InfoSphere Master Data Management Server
- change the level of an error, for example, from fatal to warning.

Error handling and exception logging example use

You want to use the add address transaction as part of a composite transaction. Normally, if you are entering a contract role location and entered an address that was already on file, the add transaction would return a fatal error and would fail, causing the whole composite transaction to fail. However, if you set the error to `warning` so the add transaction returns a warning that the address is already on file, the rest of the transaction will continue. To set the error to `warning` use the `varname` tag.

Error handling and logging configuration behavior

Behavior When Configured On

Error handling and logging does not need to be configured. Errors can be defined and logged, and error messages can be customized

Behavior When Configured Off

Error handling and logging does not need to be configured. Predefined errors can be logged, and the error messages supplied with IBM InfoSphere Master Data Management Server can be used.

Behavior When Configuration is Changed in a Production Environment
Errors can be defined and logged, and error messages can be customized at any time.

Configured During Installation
Error handling and logging can be configured at any time.

Modifying This Feature
Error handling and logging can be modified according to the procedures in the *Configuring and using MDM Server error handling and logging* section of the *IBM InfoSphere Master Data Management Server Developers Guide*.

Smart Inquiries

The IBM InfoSphere Master Data Management Server implementation only uses part of its data model,. The core product, however, does not distinguish between the used and unused parts of the data model, so some composite transactions—like getParty and getContract—read data from the unused model parts if the passed-in inquiry level includes objects from the unused part. This results in redundant database I/O, because the reads never return any data. The same problem does not exist for add and update operations as these only perform add or update I/O for the objects passed in the request.

Using the Smart Inquiries feature, you can turn off unused parts of the model. When these parts of the model are turned off, the core product does not issue any database I/O request against unused tables, and does not affect any functionality around the used parts of the model. These Smart Inquiries improve processing efficiency.

A few of the InfoSphere MDM Server built-in inquiry composite transactions, such as getParty and getContract, access nearly all the functional parts of the application. When unused parts of the data model are not turned off, these transactions do not execute as efficiently as they do in the limited functionality use scenario.

Smart Inquires example use

You might decide not to use the party financial profile and party relationships subject areas, but you still want to use the rest of the party objects. You can use Smart Inquiries to turn off the unused parts of the data model.

If the unused parts of the data model are not turned off, when you use the getParty transaction at inquiry level 4, IBM InfoSphere Master Data Management Server issues five SQLs—four for financial profile and one for relationships—that are not required for the transaction. If you disable these parts of the data model, the same getParty transaction does not return these five unnecessary SQLs, reducing the time it takes to process the transaction and load on the system.

Smart Inquires configuration behavior

Behavior When Configured On
Information about unused parts of the data model is not returned in transactions.

Behavior When Configured Off
Unnecessary information about unused parts of the data model is returned.

Behavior When Configuration is Changed in a Production Environment
Disabling parts of the data model can be changed at any time without impact other than changing the transaction responses.

Configured During Installation
This feature does not need to be installed.

Modifying This Feature
Disabling parts of the data model can be done according to the procedures in the *Modifying InfoSphere MDM Server* topic in the *IBM InfoSphere Master Data Management Server Developers Guide*.

Inquiry levels

Defining inquiry levels entails setting the parameters that determine the level of detail for objects being returned in a search or inquiry transaction. Defining the inquiry levels allows new combinations of objects to be returned. The core product business objects supported for inquiry-level customization are Person, Organization and Contract.

IBM InfoSphere Master Data Management Server offers a variety of inquiry transactions that accept one or more inquiry levels as parameters. IBM InfoSphere Master Data Management Server uses these parameters to select the correct objects to return as a part of the transaction.

Inquiry levels example use

There are two cases where you can configure a new child object for a parent business object:

- New child objects are added to the Person, Organization, or Contract objects to accommodate client requirements for an addition or extension to the IBM InfoSphere Master Data Management Server product.
- An existing child business object of Person, Organization or Contract is not currently being returned through its coarse-grained inquiry transaction—that is, through getPerson, Organization or Contract—for any of the product-defined inquiry levels, the extension framework may used to retrieve it as an extension. The parent object can also be configured to return this child object for new inquiry levels.

Inquiry levels configuration behavior

Behavior When Configured On
Inquiries with various levels can be run.

Behavior When Configured Off
Inquires can only have one level.

Behavior When Configuration is Changed in a Production Environment
Inquiry levels can be configured on or off at any time without impact other than changing the level of inquiry.

Configured During Installation
Inquiries do not need to be installed, however you must configure it in order to use the levels of inquiry.

Modifying This Feature
Inquiries can be modified according to the procedures in the *Defining Inquiry Levels* section of the *IBM InfoSphere Master Data Management Server Developers Guide*.

Summary Data Indicators

Summary Data Indicators are used to dynamically avoid reading the database for parts of the model that are not relevant to the current base entity being read. By de-normalizing the model and providing summary data indicators at the base table level, the system can decide whether it needs to read data from the child tables or not. This summary indicator must be updated whenever a relevant change is made to the data whose summary is being tracked.

Summary Data example use

An implementation of IBM InfoSphere Master Data Management Server can use the party relationship subject area, but relationships may not be present for all parties. If you use Summary Data Indicators while doing a `getParty` transaction for a party with no relationships, the system does not issue the unnecessary SQL to read from the relationship record.

Summary Data Indicators configuration behavior

Behavior When Configured On
When performing a transaction, the system can ignore parts of the data model that are not relevant.

Behavior When Configured Off
When performing a transaction, the system reads all parts of the data model.

Behavior When Configuration is Changed in a Production Environment
Summary Data Indicators can be configured on or off at any time without impact other than changing the way the data model is read.

Configured During Installation
Summary Data Indicators does not need to be installed.

Modifying This Feature
Summary Data Indicators can be modified according to the procedures in the *Customizing Summary Data Indicators* section of the *IBM InfoSphere Master Data Management Server Developers Guide*.

Pluggable primary keys

Pluggable keys provide a single point of entry for defining the primary key for a record into a database table. You can use your own implementation to create the primary key on specific tables or on all tables

Each record in the MDM Server database for operational data such as contact, product, and address, is identified by a single primary key. The primary key can be generated by one of three methods:

- By using the default key generator that comes with MDM Server
- By plugging in a custom key generator
- By passing a primary key with the object in the service request, known as a pluggable primary key

Pluggable primary keys example use

InfoSphere MDM Server can be used to manage parties that have relationships to credit cards. If the credit cards are stored as contracts, you can use the primary

pluggable primary key object on AddContract services to set the contract_id to the credit card number, as opposed to using the default key generator to generate a contract_id.

Pluggable primary keys configuration behavior

Behavior When Configured On
This feature does not need to be configured on. If a pluggable primary key object is provided in service requests then it will be used as the primary key. If there is no pluggable primary key object is provided in the request, then a primary key generator will be used, which will be either the default key generator or a key generator you can plug in as an alternate.

Behavior When Configured Off
This is not normally a feature that requires configuration change in a product environment. While it is possible to plug in a new key generator implementation or start using the pluggable primary key object in an existing production environment, care must be taken in doing so assuming the implementation already has taken into consideration a partitioning and clustering strategy for managing data.

Behavior When Configuration is Changed in a Production Environment
When pluggable primary keys is configured on, customized identifiers can be used to identify new parties If a customized primary key is used, InfoSphere MDM Server does not ensure that there are no duplicate keys.

Configured During Installation
Pluggable primary keys can be configured at any time.

Modifying This Feature
Pluggable primary keys can be modified following the procedures in the *Configuring pluggable primary keys* section of the *IBM InfoSphere Master Data Management Server Developers Guide*.

Service Activity Monitor facility

The Service Activity Monitor facility provides IBM InfoSphere Master Data Management Server with the ability to produce and extract the data necessary to generate Management Information System (MIS) reports.

InfoSphere MDM Server does not provide the reports, but the data captured in InfoSphere MDM Server enables reports to be generated.

For all types of InfoSphere MDM Server transactions, the reporting enablement feature can capture information such as:

- transaction name
- start time
- size of the request
- size of the response
- transaction duration
- transaction outcome

You can use this information to produce system reports for capacity planning and identifying areas of optimization, as well as demonstrating how InfoSphere MDM Server services and transactions are being used in a particular installation.

The impact to performance is minimal and the data capturing process is able to support reporting on a cyclical basis (such as minute-by-minute, hourly, daily, or weekly). The user can define the cycle at which InfoSphere MDM Server will capture the data.

Note: To learn more about implementing this feature, see the *IBM InfoSphere Master Data Management Server Developers Guide*.

Service activity monitoring example use

ABC Insurance uses the Service Activity Monitoring facility daily to retrieve the activity feeds from InfoSphere MDM Server. The data flow is as follows: each day, each transaction, request and response specific data are captured from InfoSphere MDM Server feeds and sent to the Service Activity Monitoring facility. The data is then made available for reporting, so a Daily Activity Report by User can be generated.

Service activity monitoring configuration behavior

Behavior When Configured On
The Service Activity Monitoring facility collects system information generated by InfoSphere MDM Server.

Behavior When Configured Off
The Service Activity Monitoring facility does not collect any data.

Behavior When Configuration is Changed in a Production Environment
The Service Activity Monitoring facility can be configured at any time, but will only collect data from the time that it is turned on to the time that it is turned off.

Configured During Installation
The Service Activity Monitoring facility can be configured at any time.

Modifying This Feature
The Service Activity Monitoring feature can be modified according to the procedures in the *IBM InfoSphere Master Data Management Server Developers Guide*.

Request/Response Framework

The Request/Response framework provides a consistent entry point into InfoSphere MDM Server. It offers common infrastructure services-such as authorization checking, transaction demarcation and others-for all incoming transactions. It is extendable because its various components are pluggable.

Some of the main components that make up the Request/Response framework are:

DWL Service Controller
A stateless session EJB that acts as the façade for all incoming requests to the Request/Response framework.

Request Handler
Contains the controller logic and dispatches the request to other components for parsing, processing, and others.

Parser Responsible for parsing the incoming request and converting it into a format understood by the target enterprise application. For example, in the case of an incoming XML request, this parser is responsible for parsing it into objects as required by the target application.

Constructor
Performs the opposite function of parser, and converts the data format returned from the target application into a format to be returned to the client.

Business Proxy
The component responsible for communicating with the target application, called after the request has gone through parsing.

The Request/Response framework allows the parser, constructor and business proxy to be pluggable. Both the parser and constructor have corresponding factory classes, which are also pluggable. A given implementation can have multiple instances of these components and use an instance based on the incoming transaction.

All IBM InfoSphere Master Data Management Server transactions are accessible through the Request/Response framework. A default parser and constructor are provided to handle the IBM InfoSphere Master Data Management Server XML transactions. If custom request or response formats are needed, new parsers and constructors can be developed and plugged into the framework. A default generic business proxy, `DWLTxnBP`, is also provided.

Related concepts

"Understanding the InfoSphere MDM Server Request Framework" on page 2
The InfoSphere MDM Server Request Framework provides a consistent entry point to InfoSphere MDM Server and is used to receive requests and issues responses in any format.

Request/Response Framework configuration behavior

The Request/Response Framework is always available in the InfoSphere MDM Server product. It does not need to be separately installed or configured in order for you to use it.

The Request/Response framework can be modified according to the procedures in the *Configuring the Request/Response framework* section of the *IBM InfoSphere Master Data Management Server Developers Guide*.

Composite transactions

A composite transaction consists of a series of single transactions. There are two methods for creating composite transactions:

- Using customized business proxies
- Using XML

Related concepts

"Understanding the InfoSphere MDM Server Request Framework" on page 2
The InfoSphere MDM Server Request Framework provides a consistent entry point to InfoSphere MDM Server and is used to receive requests and issues responses in any format.

Business requirements for using Composite XML transactions

You can consider using a composite XML transaction to group related business transactions that you want executed in one unit of work. Also, if you plan to implement simple if-then-else or looping logic among these transactions, a Composite XML transaction is also a good candidate. The Composite Transaction Framework provides syntax in XML format that you can use to create composite transactions easily to fulfill these requirements.

However, since single transactions in a composite are executed in one unit of work, you should refrain from grouping too many single transactions in one composite. The more single transactions there are in the composite, the longer it takes to complete the unit of work, and the more likely it is that you will have transaction timeout problems. Therefore, it is recommended you have no more than four single transactions in a composite.

Composite Transaction example use

The following are examples of how to use composite transactions.

Example use of a customized business proxy:

A company has a client's identification number and information for a new address, and wants to update the address information for the client. For example, the Ministry of Transportation receives a notification of a change of address from a driver with the driver license number *xyz*.

To process this request, the Ministry of Transportation may need to do several things:

- Find the driver using the driver license number provided
- If that driver is registered with the Ministry, check to see if he has a mailing address already entered. If he does not have a mailing address, add the mailing address as provided. If he already has a mailing address, update that address with the information provided.

Before the company can process the address, first the company has to query the back-office to find the driver, and determine whether it is necessary to add or update the address. This whole process encompasses a couple of queries and decision making.

One way to solve this requirement is to implement a composite `updatePartyAddress` transaction at the business proxy level. This is implemented by associating a customized business proxy to this transaction. This business proxy contains the business logic to search for the party and perform a match on the address. It then determines whether this composite transaction will invoke an "addPartyAddress" or an "updatePartyAddress" transaction.

Example use of composite XML transactions:

Normally, when you are required to update certain attributes in a Contract record (such as Contract Alert, Contract Components, Contract Party Role, or Contract Party Role Location), you need to provide the unique Contract ID along and the Last Update Date of the record you want to update. However, if the only information you have is the identifier used in an external administrative system that integrates with IBM InfoSphere Master Data Management Server and the administrative system type, then you must use a composite XML transaction to perform the update.

The composite transaction, named updateContractByAdminSysKey, may include `searchContract` or `getContractAdminSysKey`, `getContract`, `updateContract`, `updateContractComponent`, `updateContractPartyRole`, and `updateContractRoleLocation`.

If no record of the given object is found in the Contract record, this composite XML transaction invokes an Add transaction instead of an Update transaction.

Composite transactions configuration behavior

Composite transactions are always available in InfoSphere MDM Server. They do not need to be separately installed or configured in order for you to use them.

Composite transactions can be modified according to the procedures in the *IBM InfoSphere Master Data Management Server Developers Guide*.

Batch processing using MDMBatch

MDMBatch is included within the IBM InfoSphere Master Data Management Server product to you to perform batch transaction processing.

Using the MDMBatch framework, you can:

- Run existing InfoSphere MDM Server transactions in a batch mode for ready-to-use input and output file and data formats.
- Build custom batch jobs to execute custom transactions, and to support custom input and output files and data formats.

Example use of batch transactions

You can use MDMBatch to automatically perform tasks on large volumes of transactions. The following sections show examples of how batch transaction processing can be used.

Synchronizing data: A credit card company has implemented IBM InfoSphere Master Data Management Server. As part of the integration strategy with the back-end systems, a batch process is created to synchronize the credit card values, such as outstanding balance and other information, with the values that are duplicated within IBM InfoSphere Master Data Management Server. A standard file format is created that all of the back-end systems must adhere to. It contains:

- Credit Card Number
- Contract Status
- Zero or more Contract Value Type, Contract Value pairs
- For example, ″Outstanding Balance″, 1200.30
- For example, ″Minimum Payment″, 50.00
- For example, ″Minimum Payment Due Date″, April 10, 2003

The batch process reads the file defined above. For each record, it finds the contract based on the credit card native key, updates the status if required, and then updates contract component values.

- Batch run type—Contract Data Sync
- File types:
  - Incoming file—Back-End Contract Data
  - Outgoing file—Contract Data Sync Results

It is possible to send files from different back-end systems at any time. They do not have to be consolidated into one file during a traditional batch window at night.

Updating Party addresses: Every month, the post office provides a file of all residents that have moved in the previous month, their new address, and their previous address. Given that approximately ten percent of all residents move each year, the file can contain hundreds of thousands of records. The post office does not expect a file or report in return.

A batch process needs to be created to read through the file, determine if the parties listed in the file exist within IBM InfoSphere Master Data Management Server, and if they do, and if the information has not already been updated by some other means, update their address with the new address information.

- Batch run type—Post Address Update
- Incoming file type—Address Updates

For each record in the file, the batch transaction takes the following steps:

- Search the database for the party from the list, using the party's name and previous address
- If a party is found, change the party's address to the new address for each address that party uses
- Add one to a Parties Updated counter for the control report

Once the file has been processed, the control report containing the number of parties in the file and the number of parties updated is issued by print version or e-mail.

Identifying suspect duplicate parties: Periodically, the database must be searched for suspect duplicate parties that may need to be collapsed together. An inquiry batch process needs to be created that scans a pre-defined range of parties and for each party it needs to search for suspect duplicates and report on them.

- Batch run type—Suspect Duplicate Party Investigation
- Outgoing file types—Identified Suspects (outgoing file)
- Parameters—Party ID Start; Max Parties To Process

The batch transaction reads the Party ID Start and Max Parties To Process parameters and executes a query to read party information, ordered by last name starting with the Party ID Start parameter. For each party, a suspect duplicate search is performed and all found suspects (A1, A2, B) are written to the Identified Suspects outgoing file.

Once the maximum number of parties has been processed, based on the Max Parties to Process parameter, the Party ID Start parameter is updated in preparation for the next time this business run gets initiated. The total number of parties processed and the total number of suspects found are reported in the Control report.

Batch transaction processing configuration behavior

Behavior When Configured On
When MDMBatch is configured, the selected transactions can be processed automatically in batches.

Behavior When Configured Off
When MDMBatch is not configured, transactions must be processed individually.

Behavior When Configuration is Changed in a Production Environment
MDMBatch can be configured on or off at any time without impact other than changing the ability to process transactions in batches.

Configured During Installation
MDMBatch can be configured at any time. The Batch function is customized using the properties files and it is accessible through a

command line interface, which allows systems management tools, including schedulers, to start the batch in a non-interactive mode.

Modifying This Feature
Batch Transactions can be modified according to the procedures in the *Modifying InfoSphere MDM Server* section of the *IBM InfoSphere Master Data Management Server Developers Guide*.

Concurrent Execution Infrastructure for running parallel searches

Concurrent Execution Infrastructure, or CEI, provides the ability to do party or contract searches for operations that execute at the same time within the managed environment of the EJB container. CEI uses multithreading to perform multiple searches simultaneously, and then combines the results. Searches in parallel are done when one transaction executes multiple read access operations against the database. The Concurrent Execution Infrastructure can also be used for other operations that are independent and suited to be performed in parallel.

Concurrent Execution Infrastructure example use

The CEI can be used for large installations to increase IBM InfoSphere Master Data Management Server response times.

Concurrent execution infrastructure configuration behavior

Behavior When Configured On
Multiple searches can be carried out at the same time, and the results of the searches are presented together.

Behavior When Configured Off
Searches may only be performed one at a time and the results are presented separately.

Behavior When Configuration is Changed in a Production Environment
Concurrent Execution Infrastructure configuration can be changed at any time without impact other than changing the search capability.

Configured During Installation
Concurrent Execution Infrastructure does not require configuration.

Modifying This Feature
Concurrent Execution Infrastructure can be modified according to the procedures in the *Running tasks in parallel using concurrent execution infrastructure* topic in the *IBM InfoSphere Master Data Management Server Developers Guide*.

Configuration and Management Components

The Configuration and Management components support the operational configuration and management of applications. They enable administrative users to deploy, fine-tune, and manage applications within their runtime environment.

The Configuration and Management components support the configuration and management of both standalone and enterprise applications.

What are the Configuration and Management Components?

The Configuration and Management components are a series of distributed components that work together to realize the functions of configuration and management. These components are:

- **Configuration Repository**—A persistent store that holds the configuration for one or many applications. The configuration repository can distinguish between many installations of the same application-called deployments-and also between multiple running instances of the same applications, and store separate configuration for each of these.
- **Application Configuration Client**—A module running within the same process as the managed application that provides the application with read-only runtime access to the configuration stored in the Configuration Repository.
- **Management Console**—A thin client application that provides application administrators with a text-based user interface that enables them to manipulate the configuration of various applications. The console supports both interactive and unattended operation.
- **Management Agent**—A back-end system that acts on behalf of the application to realize the configuration and management functionality. The management agent supports the disconnected operation of the Configuration and Management functionality, and that of the managed application.

Who can use the Configuration and Management Components?

The Configuration and Management components have two types of users:

- **Applications**—Programmatic users that need to read the configuration information from the Configuration Repository. Applications only require the presence of the Configuration Repository and the Application Configuration Client. The Management Agent and the Management Console are not directly required for the application to function.
- **Administrators**—Human users that need the ability to view and change the configuration of the applications that they manage. Administrators use the Management Console and the Management Agent to access the Configuration Repository. The managed applications are not required to be operational at the time when their configuration is managed. If the applications are operational, the Management Agent informs them about any changes to their configuration so that they can dynamically load the new configuration values.

Configuration

To recognize the different sets of requirements that apply to application configuration before and after the application has been deployed in the operational environment, configuration is divided up into two categories: static and dynamic.

Configuration that is intended to be modified only at application development, assembly, or deployment time is considered to be static. This kind of configuration can only be modified before the application becomes operational. Changing any of the static configuration parameters fundamentally changes the application and, therefore, would require its redeployment. New code or resources can be added as a result of changing static configuration. Consequently, the application would have to be rebuilt, retested, and redeployed.

Configuration that controls the behavior of the application while operational is considered to be dynamic. The application observes and reacts to changes in its dynamic configuration without having to be rebuilt, retested or redeployed. Changes in dynamic configuration do not result in application resources having to be added, changed, or removed.

Note: Only a limited set of configuration settings are available through the new Configuration Repository in both IBM InfoSphere Master Data Management Server and InfoSphere MDM Server Event Manager. Priority has been given to the

dynamic configuration items that need to change while the system is in operation without the need to restart the Application. More configuration items will be migrated in future releases.

Configuration Definitions and Schemas: Configuration definitions are XML documents that contain all the configuration items and their values as defined during the development process. These definitions are packaged in the application archive. They can be modified during the application assembly phase and repackaged with the application. At deployment, the configuration definitions are used to establish the initial configuration in the configuration repository. They can also be used as the vehicle for replicating existing configuration.

Configuration definitions can contain both static and dynamic configuration. For an operational application, the Management Console only allows administrators to change dynamic configuration items.

The configuration definition distributed with the application is considered to contain the factory defaults for the configuration. Any changes to this configuration may potentially be overwritten by upgrades to subsequent versions of the application. Users that wish to change their configuration while retaining factory defaults should do so from the Management Console after the application (and its configuration) is deployed.

Configuration and Management Components configuration behavior

IBM InfoSphere Master Data Management Server has access to the features of the Configuration Repository, Application Configuration Client, Management Console, and Management Agent components.

Configuration and management components are installed with main IBM InfoSphere Master Data Management Server installation. See the *IBM InfoSphere Master Data Management Server Installation Guide* for more information.

Configuration and management components can be modified according to the procedures in the Using Configuration and Management Components section of the *IBM InfoSphere Master Data Management Server Developers Guide*.

Data validation

Data validation is carried out on data submitted in IBM InfoSphere Master Data Management Server transactions to ensure that the data satisfies certain requirements expressed in validation rules. Data can be validated by levels, for controller or business components, and by types, for internal or external types.

Data can be validated at two levels:

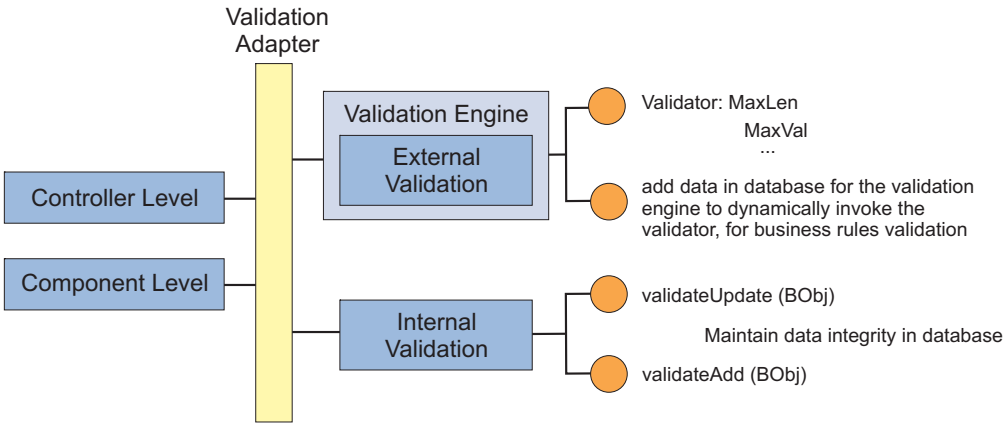
- **Controller level**—Used to process as many validations as possible at the pre-transaction stage. Most validations are performed at the controller level as opposed to the business component-level, in order to improve performance.
- **Business component level**—Used for validations that can only be done as the transaction is carried out.

There are also two types of validation:

- **Internal**—To maintain database integrity. This code is generally not accessible to developers and administrators

- **External**—Validates content, and uses information accessible and modifiable by developers and administrators.

The diagram below shows how the validation levels and types work together.



Data validation example use

The Controller level validations are used for accumulating and returning as many error messages as possible at the pre-transaction stage in order to reduce the number of validations during their session, and improve the performance for users. The Business component validations are performed during the transaction. For example, when an add party is suspected to be a duplicate, the system does additional business component validations.

Data validation configuration behavior

Behavior When Configured On
Both internal and external validation can be configured off or on. When configured on, data is validated.

Behavior When Configured Off
Both internal and external validation can be configured off or on. When configured off, data is not validated.

Behavior When Configuration is Changed in a Production Environment
Data validation can be configured on or off at any time without impact other than changing the data validation

Configured During Installation
Data validation can be configured at any time.

Modifying This Feature
Data validation can be modified according to the procedures in the *Validating Data* section of the *IBM InfoSphere Master Data Management Server Developers Guide*.

External business rules

External business rules are pieces of business logic external to IBM InfoSphere Master Data Management Server that are incorporated into InfoSphere MDM Server transaction processing. While every configuration option, validation requirement, or property file entry could be considered an external rule, this section discusses the rules that are used for complex processing and decision-making.

The External Rule Component is used by InfoSphere MDM Server when it performs a piece of business logic as part of a transaction. In performing that piece of business logic, InfoSphere MDM Server knows what data must be provided and what data must be returned. Based on the data returned, InfoSphere MDM Server can act conditionally on it.

External business rules example use

You can use as external rule to define the criteria for identifying a suspected duplicate-party matching. When you add a new party, a search is performed to determine if there are any suspect duplicates of the party being added. One step in the suspect search is to match the new party against the suspects found to determine how closely they relate. Because there are different methods for matching parties, and these methods may change, party matching is an externalized rule. InfoSphere MDM Server knows that party matching must be done, what data must be provided to the externalized matching—a source and target party—and what data is returned—match and non-match scores. The externalized rule that you create for party matching determines what elements are used to match the parties. These elements are called critical data. The match relevancy scores related to that critical data—for example, 5 points assigned to a match, and -5 points assigned to a non-match—are also externalized, but in a code table.

External business rules configuration behavior

External business rules are always available in the InfoSphere MDM Server product. They do not need to be separately installed or configured in order for you to use them.

External business rules can be modified according to the procedures in the *Configuring external business rules* section of the *IBM InfoSphere Master Data Management Server Developers Guide*.

Notifications

Notifications are application-to-application messages containing data relevant to specific events that have occurred in the sending application. In IBM InfoSphere Master Data Management Server, the Notification Manager component is used to send notification messages from InfoSphere MDM Server to different destinations, for example, from InfoSphere MDM Server to other systems to let the other systems know about critical party data changes. Currently, the Notification Service uses the Java Message Service (JMS) to publish messages to, for example, WMQ for queuing to whatever listener applications subscribe to them.

Notifications example use

The following are examples of how the notification feature can be used:

- As part of your implementation, it is required that if a party's legal name is changed, that change must be propagated to another system, along with some other party details. You have determined that none of the notification samples included with IBM InfoSphere Master Data Management Server product can be used. Instead, a new notification called ″legal name change″ is created, and the content for that notification message is customized to include the party details required.
- You have recently added another mandatory element to the required party information. To ensure that end users add this element when entering party

information, you can customize the error notification to include this element and display the notification when the element is not entered with the party information.

Notifications configuration behavior

Behavior When Configured On
End users receive customized notifications of errors.

Behavior When Configured Off
End users receive the generic notifications included with IBM InfoSphere Master Data Management Server.

Behavior When Configuration is Changed in a Production Environment
End users who previously received the generic notifications now receive the customized notifications.

Configured During Installation
Notifications can be configured at any time.

Modifying This Feature
Notifications can be modified according to the procedures in the *Configuring and implementing notifications* topic in the *IBM InfoSphere Master Data Management Server Developers Guide*.

Security service

InfoSphere MDM Server security service provides a framework for implementing authentication and authorization services as well as services for managing the security data. This chapter discusses configuring and administering the security service.

The security service has two main components:

Runtime security services
Includes runtime user authentication and resource, or transaction, authorization checks. These security services are supported using a framework that security providers can be plugged into. Each security provider is responsible for performing the authentication and authorization checks, usually against a different security data repository or service. Multiple security providers can be configured.

Security data management
An administration interface to manage security data including user and group profiles, as well as their authorization for various resources

InfoSphere MDM Server security service comes with two security providers:

Default security provider
This provider uses the data stored in a relational database to authorize users and groups for incoming transactions.

LDAP security provider
This provider performs the transactional authorization check against an LDAP repository.

These providers only implement the authorization check, and do not perform authentication.

In addition to the security service, a security manager is provided to administer the security data in a relational database. This is the same repository used by the

default security provider to perform runtime authorization checks, however, the security service does not support security data administration for LDAP repository.

Security Service example use

Default security provider

You use the default security provider to set which groups and users are authorized to perform specific changes to the data when a transaction is run. For example, you can use this security provider to set which users have the authority to make changes to specified data, and then to ensure that a user has permission to perform changes to the data when a transaction is run.

LDAP security provider

Use the LDAP security provider to set and check which users can perform specific transactions. For example, you can use the LDAP Security Provider to set which users have the authority to run specified transactions, and then to ensure that a user has permission to perform those transactions when the transaction is run.

Security Service configuration behavior

Behavior When Configured On

When a user attempts to perform a transaction, a security checks is performed to determine if that user is authorized to perform that transaction. If the user is authorized, the transaction is completed; if the user is not authorized the transaction is cancelled.

Behavior When Configured Off

No security checks are performed when users perform transactions.

Behavior When Configuration is Changed in a Production Environment

Security Service configuration can be changed at any time without impact other than changing the security check capability.

Configured During Installation

Security Service can be configured at any time. See the procedures in the *Setting and administering the security service* topic in the *IBM InfoSphere Master Data Management Server Developers Guide*.

Modifying This Feature

Security Service can be modified according to the procedures in the *Setting and administering the security service* topic in the *IBM InfoSphere Master Data Management Server Developers Guide*.

Unique and persistent ID generation

The Extended Key Generation framework provides the ability to:

- Generate different types of identifiers such as numeric, alphanumeric, numeric string and alphabetic
- Generate different types of identifiers of variable length
- Return a set of identifiers instead of a single identifier
- Ability to plug custom ID generators
- Ability to configure validation rules for checking the generated identifiers

Unique and Persistent ID generation example use

A financial institution might use this feature to generate an enterprise Party ID that can a customer can use when logging into the banking system, instead of using the number for a specific account.

Unique and persistent ID generation configuration behavior

Unique and persistent ID generation is always available in the InfoSphere MDM Server product. It does not need to be separately installed or configured in order for you to use it.

Unique and persistent ID generation can be modified according to the procedures in the *Unique and persistent ID generation framework* topic in the *IBM InfoSphere Master Data Management Server Developers Guide*.

Web Services

Support for Web Services in IBM InfoSphere Master Data Management Server is currently available at two different levels:

- natively in the InfoSphere MDM Server Enterprise Application
- through the InfoSphere MDM Server Web Services Adapter

InfoSphere MDM Server Web Services

InfoSphere MDM Server Web Services are an intrinsic part of the InfoSphere MDM Server enterprise application and do not require additional deployment or configuration. The operations made available through the Web Services map one-to-one with InfoSphere MDM Server transactions.

InfoSphere MDM Server Web Services are compliant with the WS-I Basic Profile version 1.0. WSDL files that describe the services are available from the application server after the InfoSphere MDM Server enterprise application is deployed.

InfoSphere MDM Server Web Services can be invoked in two ways:

- by directly submitting SOAP requests over HTTP(S)
- by using the WSDL files to generate client code and then programmatically invoking the operations

InfoSphere MDM Server Web Services are always available, and cannot be configured off.

Restriction: As of this release, only a limited set of InfoSphere MDM Server transactions are exposed through InfoSphere MDM Server Web Services. For details, please see the IBM InfoSphere Master Data Management Server release notes.

Web Services Adapter

The Web Services Adapter is an add-on web application that is deployed and configured separately from the InfoSphere MDM Server enterprise application.

The Web Services Adapter consists of one Web Service with a single operation only. This operation allows InfoSphere MDM Server XML requests and responses to be tunneled through SOAP messages over HTTP(S). The InfoSphere MDM Server XML requests and responses are passed, as-is, to and from the Service Controller.

Web Services Adapter configuration behavior

Behavior When Configured On
The Web Services interface is available through the Adapter.

Behavior When Configured Off
The Web Services interface is not available through the Adapter.

Behavior When Configuration is Changed in a Production Environment
The Web Services Adapter can be configured on or off at any time without impact other than changing the access to the Web Services interface.

Configured During Installation
The Web Services Adapter can be installed and configured manually at any time.

Modifying This Feature
The Web Services Adapter can be modified according to the procedures in the *Using the external Web Services Adapter* topic in the *IBM InfoSphere Master Data Management Server Developers Guide*.

Tracking IBM InfoSphere Master Data Management Server performance

IBM InfoSphere Master Data Management Server provides the ability to capture performance statistics for the elapsed time of a transaction, including the time it takes to complete different parts of the transaction.

Performance tracking is a configurable option, and can be turned on or off through the Administration application interface. As well as enabling or disabling the collection of performance data, you can also specify the level of data collected.

See the *Performance Tracking* section of the *IBM InfoSphere Master Data Management Server System Management Guide* for more information.

Tracking Performance example use

If performance logging at Level 2 is turned on, InfoSphere MDM Server logs the elapsed times to perform an operation on a business component, such as addParty(). The breakdown of that operation—validations, database access, extension, and external services elapsed times—is also logged. This cataloging is made possible through the use of transaction correlators.

Tracking performance configuration behavior

Behavior When Configured On
When performance tracking is turned on, the Performance Monitor captures elapsed times for the following categories of a transaction depending on the level chosen. When performance tracking is configured to:

Level 1
Measures the overall transaction time from the time the thread enters the application controller.

Level 2
Measures components transaction time, validation, external components, such as Trillium and client extensions, as well as level 1 measurements.

Level 3

Measures the amount of time for DWLRequestHandler, XMLRequestParser and XMLResponseConstructor component to parse incoming XML and to prepare XML response, as well as level 2 measurements.

Behavior When Configured Off

No tracking is performed.

Behavior When Configuration is Changed in a Production Environment

If performance tracking is configured on, performance data is only available going forward from the point that the feature was turned on

If performance tracking is configured off, previously captured performance data is available, but no new data is collected.

Configured During Installation

Performance tracking can be configured at any time.

Modifying This Feature

Performance tracking can be modified according to the procedures in the *Tracking Performance* section of the *IBM InfoSphere Master Data Management Server Developers Guide*.

InfoSphere MDM Server platform business features

The following InfoSphere MDM Server platform features are mainly aimed at a business audience.

Point in time history

IBM InfoSphere Master Data Management Server has an audit, or history, database. The audit database is a duplicate of the operational database (with the exception of the code/rule tables) with additional audit attributes. The audit tables are populated at the time of execution of any IBM InfoSphere Master Data Management Server transaction, via the default set of triggers for the IBM InfoSphere Master Data Management Server product. These tables store the actual data that has been added or updated in the transaction. In other words, using the audit tables makes it possible to see exactly what a party, for example, looked like for a given point in time.

IBM InfoSphere Master Data Management Server allows any inquiry transaction (`get***`) to return either current or point-in-time data. If a valid `<inquireAsOf>` element occurs in the request control, the `get` transaction takes its data from the audit tables rather than the operational ones.

Related concepts

"History inquiry date range images" on page 23

Point in time example use

The point in time history feature can be used to review changes to a party's information over time. For example, you can use point in time history to review when address changes for a specific party were entered, and what those changes were. PIT can be also be used in conjunction with TAIL, to create an audit trail showing who made what changes and when.

Point in time configuration behavior

Behavior When Configured On

Gives users access to point in time history, which provides a snapshot of what the client file, or a portion of the client file, looked like as of a defined point (either a single date or range of dates) in the past.

Behavior When Configured Off

History inquiries are not available, and the only party information available is the information that is currently in the database.

Behavior When Configuration is Changed in a Production Environment

If the history inquiry date range images feature is configured on, PIT history is available from the time the feature is turned on-historical queries will not show changes to the business objects from before the feature was turned on.

If History Date Range Images is configured off, historical queries will show PIT history until the time that the feature was configured off, but not after this time.

Configured During Installation

Point in time history can be configured at any time. However, in order to have a complete PIT history record, you must configure this feature on during installation.

Modifying This Feature

Point in time history can be modified according to the procedures in the *Retrieving audit, or point in time, history* section of the *IBM InfoSphere Master Data Management Server Developers Guide*.

History inquiry date range images

The history inquiry date range images feature retrieves Point In Time (PIT) history for each change that has occurred to predefined business objects, within a particular date range. Historical queries show how data has changed over a set period of time. These queries are required for a number of reasons, such as:

- To audit data changes
- To verify that data has been changed
- To track the date when data changed
- To determine why an effective change was not propagated

InfoSphere MDM Server provides two sets of triggers with the database installation:

- A set of compound triggers: CreateTriggers_compound.sql
- A set of simple triggers: CreateTriggers_simple.sql

If the compound triggers are installed, each of the operational tables within the InfoSphere MDM Server product database has two active triggers. If simple triggers are installed, each of the operational tables within the InfoSphere MDM Server product database has only one trigger for update actions. For more detailed information, see the *Database considerations for history inquiry* topic in the *IBM InfoSphere Master Data Management Server Developers Guide*.

Related concepts

"Point in time history" on page 22

Definitions

The following list defines terms, acronyms, and abbreviations required to understand history inquiry date range images:

PIT Point In Time history transactions; a snapshot of what the client file, or a portion of the client file, looked like as of a defined point in time in the past (inquireAsOfDate).

TAIL Transaction Audit Information Log; the subsystem in which information is recorded about the transactions executed within InfoSphere MDM Server.

The terms *image* and *view* have been adopted within InfoSphere MDM Server to describe the following:

- An image is the result of one PIT history inquiry transaction triggered by changes in one or more selected object drivers
- A view is a collection of images gathered between two provided dates—for example, for the getImagesByParty transaction, each of the images returned are the result of a point in time getParty transaction

The term *history change drivers* has been adopted within InfoSphere MDM Server to describe a set of predefined objects for a particular view that, when updated or changed, may trigger the creation of a new image or view.

For more information, see the *Storing and retrieving the Transaction Audit Information Log* topic in the *IBM InfoSphere Master Data Management Server Developers Guide*.

History inquiry date range images example use

The history inquiry date range images feature allows users to query the history database:

- Using a party ID and date range to determine how contract party roles for the specified party changed during the specified date range
- Using a contract ID and date range to determine how contract components, contract alerts, contract relationships, contract party roles, contract role locations, contract role situations, and contract role identifiers have changed for the specified contract during the provided date range

History inquiry date range images configuration behavior

Behavior When Configured On

Gives users access to:

- Point in time history, which is a snapshot of what the client file, or a portion of the client file, looked like as of a defined point-either a single date or range of dates-in the past
- Transaction Audit Information Log, which is the subsystem in which information is recorded about the transactions executed within InfoSphere MDM Server.
- An image that is the result of one PIT history inquiry transaction triggered by changes in one or more selected object drivers
- A view that is a collection of images gathered between two provided dates: for example, for the getImagesByParty transaction, each of the images returned are the result of a point in time getParty transaction

Behavior When Configured Off

History inquiries are not available, and the only party information available is the information that is currently in the database.

Behavior When Configuration is Changed in a Production Environment

If the history inquiry date range images feature is configured on, PIT history is available from the time the feature is turned on-historical queries will not show changes to the business objects from before the feature was turned on.

If the history inquiry date range images feature is configured off, historical queries will show PIT history until the time that the feature was configured off, but not after this time.

Configured During Installation

In order to have a complete PIT history record, you must configure this feature on during installation. For more information, see the topic *Storing and retrieving the Transaction Audit Information Log* in the *IBM InfoSphere Master Data Management Server Developers Guide*.

Modifying This Feature

the history inquiry date range images feature can be modified according to the procedures in the *Storing and retrieving the Transaction Audit Information Log* topic in the *IBM InfoSphere Master Data Management Server Developers Guide* .

Storing and retrieving the Transaction Audit Information Log

The Transaction Audit Information Log (TAIL) module provides services for the storage and retrieval of transaction log information for IBM InfoSphere Master Data Management Server. TAIL has mainly database-driven configuration options. TAIL can be configured to log any persistent transactions—adds and updates—as well as some or all of their associated internal transactions, depending on the particular transaction. InfoSphere MDM Server transactions can consist of a number of sub-transactions that are executed as a part of the larger external transaction. For example, when TAIL logs an external transaction, also called a business transaction, it can be configured to also log some or all of its internal transactions. When an audit transaction is retrieved, any of its internal transactions that have been logged are also retrieved.

The TAIL module has three main features:

- configuring
- logging
- retrieving transactions

TAIL logging

Logging on to TAIL occurs seamlessly within the IBM InfoSphere Master Data Management Server product, as long as the add/update transaction has been configured for logging.

TAIL may also be configured to log particular external and internal transactions, such as flagging the transaction log indicator for a particular transaction listed in the CDBUSINESSTXTP table to Y. This mainly impacts the CDBUSINESSTXTP and BUSINTERNALTXN database tables of the InfoSphere MDM Server product.

Note: You should use the System Maintenance Transaction Audit Log screen to turn logging on and off at the transaction level. For more information, see *Tracing data changes using the Transaction Audit Information Log (TAIL)* in the *IBM InfoSphere Master Data Management Server System Management Guide.*

Information logged in TAIL

For a persistent transaction (add, update, or delete), the following information gets logged to the TAIL database tables:

TRANSACTIONLOG table
Logs an entry for the external/business transaction type being executed. It also logs items from the DWLControl object from the initial transaction request.

INTERNALLOG table
Logs all of the internal transactions executed within the context of the external transaction. For example, an addIncomeSource transaction may be an internal transaction to an external addPerson transaction.

INTERNALLOGTXNKEY table
Creates entries for each transaction key and its corresponding values for each internal transaction executed. The INTERNALTXNKEY database table is preconfigured/prepopulated with information on which keys are logged for a particular internal transaction. For example, for an IncomeSource business object the PartyId will be logged along with its actual value (element_value).

The above transaction information is stored in TAIL when **all** of the following conditions are met:

- The information is provided by the client.
- The transaction is persistent (add/update/delete).
- The TAIL engine is ″ON″.
- The transaction is marked to be logged in TAIL.
- The transaction is successful in InfoSphere MDM Server.
- The TAIL process is successful.

TAIL retrieval

TAIL information can be retrieved through the getTAIL request transaction. The more parameters you add to a TAIL request, the more specific the results of the request will be. In other words, the more parameters you supply, the narrower the result set is.

For example:

- If a PartyId and a Business Transaction Type are specified, then the only transaction logs returned in the result set area are those that satisfy both conditions.
- If the client specifies ExternalCorrelationId, ClientSystemName, and/or ClientTransactionName, then the transaction logs returned in the result set area are those that satisfy the initial condition.

Transaction Audit Information Log example use

A company determines that they require an audit trail for all address changes, so they log all external transactions that directly affect addresses, as well as all internal components that affect addresses. From these logs, they are able to perform getTAIL transactions to determine who made changes to the address information, and what changes were made.

Transaction Audit Information Log configuration behavior

Behavior When Configured On
Information about transactions that add and update the database is collected, and can be retrieved when required.

Behavior When Configured Off
Information is not collected and so the history of transactions is not available.

Behavior When Configuration is Changed in a Production Environment
If TAIL is configured on, transaction history is available from the time the feature is turned on-historical queries will not show transactions that were performed before the feature was turned on.

If TAIL is configured off, historical queries will show transactions until the time that the feature was configured off, but not after this time.

Configured During Installation
In order to have a complete TAIL log, you must configure this feature at installation.

Modifying This Feature
TAIL can be modified according to the procedures in the *Storing and retrieving the Transaction Audit Information Log* topic in the *IBM InfoSphere Master Data Management Server Developers Guide*.

Source values and data decay

The purpose of the source values feature is to establish a standardized approach to store and retrieve information, or values, that come from external sources when the attributes of those values do not fit the structure of IBM WebSphere® products. The source values feature allows you to identify the system, application, or user that provided the value. It also stores the date the value was collected, as well as a history of the source that provided the value and the date when that value changes.

The Source Value System adds, updates and gets the following information within IBM InfoSphere Master Data Management Server:

- Records the source-system, application or user-that provided the specified function details, for example privacy/preference, campaign, and others
- Records the date when the information was collected
- Keeps a history of the source and source date

The Source Value System is used as part of several function areas within InfoSphere MDM Server. The Source Value System works with any entity and it is also included in the specific function transaction. For example, when a change is made to a party's privacy preference, the Source Value System is a part of the function transaction. The Source Value System is required in the following function areas:

- Privacy/Preference
- Campaigns
- Source System is also included in seven entities in core Party Module. These entities are
- Person
- Organization
- Person Name

- Org Name
- Party Identification
- Party ContactMethod
- Party Address

Data decay

Similar to source values, the purpose of data decay, is to establish a method for storing and retrieving the last verified date and last used date of key attributes and objects. This feature permits you to store and update the last verified date, last used date, and the source of data at the attribute and object levels and gives you the ability to efficiently retrieve these values.

The following table shows all the IBM InfoSphere Master Data Management Server entities with support for data decay and source values.

Entity	Data Decay	Source Values
Party	yes	yes
Person	yes	yes
Person Attributes	yes	no
Org	yes	yes
Org Attributes	yes	no
Person Name	yes	yes
Person Name Attributes	yes	no
Org Name	yes	yes
Org Name Attributes	yes	no
Party Address	yes	yes
Party Contact Method	yes	yes
Party Identification	yes	yes

Source Values example use

For examples of Source Values, see:

- “Party privacy preference”
- “Campaign”
- “Party value” on page 29
- “Party grouping” on page 29

Party privacy preference:

Cindy Clark, a banking customer, calls to request that she no longer receive marketing information from her bank. The IBM InfoSphere Master Data Management Server Service Representative changes Cindy’s profile so she no longer receives any marketing information. For auditing purposes, the institution must have a record of the source of the change—the Call Center application—and the date the change was made by the source.

Campaign:

A large enterprise institution has different systems involved with the campaigns for their Master Data Management Servers. Default Source keeps track of the source of the application for each particular campaign.

Party value:

Many large institutions have different business units that place value designations on their IBM InfoSphere Master Data Management Server base. Default Source records the source system that calculated these designations, which are determined by the data warehouse systems, applications or specific users.

Party grouping:

Many large institutions have business units that group Master Data Management Servers together based on similar behaviors, product portfolio and other characteristics. Default Source records which data warehouse, application or user assigned the group designation to the IBM InfoSphere Master Data Management Server.

Data Decay example use

A life insurance company might have multiple systems that store client information, such as the client's date of birth. The date of birth that is stored in the life insurance system must be accurate, as it is used to calculate premiums, mortality calculation, reserves and other items. However, the date of birth stored might be different from the date of birth that is stored in the Non Registered SegFund system, where the date of birth is not actually mandatory and could be stored as a defaulted value, for example, `Jan 01`.

When this party information with two different dates of birth is reconciled as part of the data stewardship suspect processing, the Data Decay and Source Value information in the data is used to ensure that the most accurate data is carried over to a new party during a collapse process. The decision process that determines which data to use is much more effective and less prone to error the system knows the source, the last used date, and the last verified date for key attributes and objects. In the example described, if the system knows that `Jan 01` is inserted as a default value, then it can more accurately determine that if the two birth dates supplied are different, then `Jan 01` is not the party's actual date of birth.

Source values and data decay configuration behavior

Behavior When Configured On
: Source values does not need to be configured. If you use source values you know what information was entered by default, where information came from, and so on.

Behavior When Configured Off
: If you do not use source values, information about the source of data is not stored, so you do not know where the data came from or whether it was entered by default.

Behavior When Configuration is Changed in a Production Environment
: Source values can be configured on or off at any time without impact other than changing the insertion of values.

Configured During Installation
: Source values can be configured at any time.

Modifying This Feature
Source values can be modified according to the procedures in the *Modifying IBM InfoSphere Master Data Management Server* section of the *IBM InfoSphere Master Data Management Server Developers Guide*.

Attach documents

The attach documents feature can be used to maintain the reference to a content asset in a Content Management System from an InfoSphere MDM Server entity. It enables you to attach a reference to any entity and it provides services for that purpose. In addition, the UI—if enabled—can search a CMS using the sample adapters and attach content using the services provided.

Attach documents example use

Attach documents can be used to maintain the reference to a driver's license image for a party, the reference to a birth certificate image for a party, or the reference to an insurance application for a contract.

Attach documents configuration behavior

Behavior When Configured On
The attach documents feature maintains the reference to content assets in the Content Management System by using an external CMS adapter implementation mapped to the property `<cms_name>`. This property represents the CMS type—such as IICE or FileNet—which is configured in the property file `CMSConfig.properties`. The provider URL and other details are maintained according to the specific requirements of the adapter.

You can configure information that pertains to the CMS repository where the external content reside through the `AdminEObjCdRepositoryTp` code table object.

Behavior When Configured Off
The default property settings in the configuration files apply.

Behavior When Configuration is Changed in a Production Environment
Attach documents can be configured on or off at any time without impact other than the reference to content assets in the Content Management System.

Configured During Installation
The attach documents feature can be configured at any time.

Modifying This Feature
For CMS Integration, a customer can provide their own implementation of the CM adapter for their CMS.

For Content Reference, this feature can be modified using InfoSphere MDM Server mechanisms like extensions to add additional attributes to the Content Reference business object, or External Rule for search.

Conditional storage of duplicate parties

The conditional storage of duplicate parties feature in InfoSphere MDM Server suspect processing services provides for the persistence of duplicate parties.

InfoSphere MDM Server has always been perceived as a customer data management system that maintains a ″golden″ copy of a party. To achieve this capability, suspect processing feature included with InfoSphere MDM Server has a

set of externalized and customizable business rules that prevent the persistence of duplicate parties in the system. While adding a new party, if an existing party is found with an exact match, known as an A1 match, then the new party, rather than being added as a new record, is used to update the existing party.

However, some business needs may require the persistence of these duplicate parties in the InfoSphere MDM Server database. The conditional storage of duplicate parties feature enables businesses to maintain multiple profiles (instances) of the same party based on a condition, such as one based on the Line of Business.

This feature is configurable and may be turned on or off.

Conditional storage of duplicate parties example use

This feature can be used when you are adding a new party to the InfoSphere MDM Server database, and the party has a Line of Business (LOB) relationship that is of a different type than the duplicate party (the A1 matching party, or best A1 matching party if more than one is found) in the database. The incoming party is added to InfoSphere MDM Server and the duplicate parties are marked as ″Parties are Duplicates. Collapse is not permitted″.

Conditional storage of duplicate parties configuration behavior

Behavior When Configured On

While adding a new party, if an existing party is found with an exact (A1) match, and the records vary across Lines of Business, then the new party is added to InfoSphere MDM Server and the two parties are marked as suspects with a status of ″Parties are Duplicates. Collapse is not permitted.″

While updating critical data for an existing party, if the suspect re-identification process finds an exact (A1) match and the records vary across Lines of Business, then the parties are marked as suspects with a status of ″Parties are Duplicates. Collapse is not permitted.″

While marking two parties as suspects, if the party records vary across Lines of Business, then the parties are marked as suspects with a status of ″Parties are Duplicates. Collapse is not permitted.″

Behavior When Configured Off

While adding a new party, if an existing party is found with an exact (A1) match, and the records vary across Lines of Business, the new party is used to update the existing party instead of adding a new record for that party.

Behavior When Configuration is Changed in a Production Environment

The conditional storage of duplicate parties to InfoSphere MDM Server is configurable and may be turned on or off. However, this affects whether an exact (A1) match is persisted or the new party is used to update the existing party instead of adding a new record for that party.

Configured During Installation

The conditional storage of duplicate parties to the InfoSphere MDM Server database is configurable, and may be turned on or off. However, this impacts whether exact matches are persisted or not.

Modifying This Feature

Rules for conditional storage of duplicate parties can be modified according to the procedures in the *IBM InfoSphere Master Data Management Server Developers Guide*.

Event Manager and Evergreening data

For details of the Event Manager and Evergreening data, see:

- "About Event Manager"
- "About the Evergreen application" on page 33
- "Typical Event Manager business scenarios" on page 34

About Event Manager

Event Manager is a generic triggering component that can provide the following types of capability:

- Trigger business transactions to be executed based on the passage of time
- Record various activities (events)
- Test if an business event has occurred
- Notify via JMS that a event has occurred

An event has no duration; it is a incident at a point in time. An event is always associated with a business object, for example, a party or a contract. To set up Event Manager, you create one or more business rules to define an event and then execute the business rules against the business object to detect the occurrence of the event.

The occurrence of a business event is recorded and a notification is sent out to other applications or business systems, using Publish and Subscribe. Other applications can listen for the occurrence of these business events and perform specific processing for those events.

There are three methods of capturing events.

- A detected event, where an application passes data to Event Manager. Event Manager gathers additional information if required and pass it to the event detector to determine if an event has occurred.
- A time-based event, where, on a periodic basis, Event Manager selects objects to be processed in order to determine if an event has occurred, for example, a person attained a specific age.
- Direct message interface, where a client sends a message to Event Manager indicating an event has occurred. This interface can be used by any application wishing to add this capability to their application, for example, a call center application where a client informs a call center rep that a particular thing has occurred, for example, I just inherited lots of money.

You can integrate Event Manager with any business system and track events for any business object. For example, if the business system is IBM InfoSphere Master Data Management Server, events can be associated with a party identified by party ID.

Event Manager executes the business rules to determine which events have already occurred and which events might potentially happen in the future. Business Objects with a Next Processing Date of today or earlier are processed. Event

Manager stores the information about events that occurred for the business object and generates a notification (JMS message) to notify other business systems about events that have happened.

The following are types of events:

Transaction data events
Occur when there is a change to important data in the database. The event is executed when a change is made as a result of a business transaction executed by the business system. The business system has to call Event Manager at the end of the transaction to trigger event processing. When the business system has changed important data and Event Manager (through external rules) determines that a happening (an event) has occurred, the event is persisted. To find out more about implementing this type of event, see the *Calling Event Manager from the business system* topic in the *IBM InfoSphere Master Data Management Server Developers Guide*.

Time passed events
Usually occur with the passage of time. By executing business rules against the business object, usually a date comparison, it is determined that the event has occurred. For example, by looking at party data it is determined that this person is turning 18 years old. To find out more about implementing this type of event, see the *Starting time-based event detection* topic in the *IBM InfoSphere Master Data Management Server Developers Guide*.

Explicit events
These events are typically user-defined. For instance, a Customer Sales Representative (CSR) knows that a party is getting married in two weeks or has won the lottery. The CSR may want to persist this type of event. To find out more about implementing this type of event, see the *Creating user explicit events* topic in the *IBM InfoSphere Master Data Management Server Developers Guide*.

Repeated events
The system uses Event Manager to report on the use of repeated events. InfoSphere MDM Server has the ability to report transaction usage patterns. For instance, if ABC Insurance is using a business rule defined in Event Manager to detect parties for which the Gender value has been changed three times in three months, then InfoSphere MDM Server can send a notification to the CSR when the Gender value for a party is changed for the third time in a three-month period. Users can further replicate this business rule to accommodate similar corruption scenarios.

About the Evergreen application

The Evergreen application, within InfoSphere MDM Server, provides the capacity to identify potential duplicate parties that may have been introduced into the InfoSphere MDM Server database—creating suspect transactions—and the ability to collapse parties without human intervention, using rules to support data survival-collapsing with rules.

In the context of IBM InfoSphere Master Data Management Server, the word "evergreen" refers to how the Evergreen application keeps the InfoSphere MDM Server operational data store current and enduring.

Keeping the InfoSphere MDM Server database as the golden book of record for client information requires both a rigorous data maintenance process and underlying support as a part of the InfoSphere MDM Server product. The primary objective of the process and product support is to ensure InfoSphere MDM Server

database stays up to date, containing data that is high-quality and current, with no duplicate party records. This support the concept of evergreening within the InfoSphere MDM Server product we use the Event Manager and have created an Evergreen application.

The Evergreen application works by:

- Periodically checking the data base for duplicate parties that may have been introduced during the normal processing or loading of data
- Collapsing the information from duplicate suspect parties into one confirmed party, by comparing the source party to the best match in the suspect table and collapsing the combined information into one new party, inactivating the suspect parties.

Detecting duplicate parties is executed for a number of reasons, including:

- When party matching rules have changed
- When suspect processing in enabled after the system has been running for a period of time with it disabled
- When cleaning up duplicates after loads or batch cycles when suspect processing has been bypassed, after the initial load, or when the load process (ETL) uses different suspect duplicate detecting rules—this may be caused when different tools are used

Typical Event Manager business scenarios

The following scenarios illustrate how Event Manager is used in business situations.

Changing insurance coverage: Mr. Michael Carlton calls the CSR of the Global Life and Group Insurance Company to change his coverage from Single to Family. He informed the CSR that he is now married and would like to add his spouse as a dependant. The CSR accesses Mr. Carlton's records and updates the coverage type as well as the marital status from single to married. The marriage event is persisted as it is considered a significant occurrence in a person's life. The insurer may want to send to the customer a congratulations letter as well as the Protect Your Family with Life Insurance brochure.

Handling automatic transactions: Mr. Frank Delhomme leases a new vehicle from BMW (Canada) Leasing. As part of the leasing agreement, the leasing company requires that Frank's payments be made via pre-authorized check (PAC). The lease also states that if there is any problem receiving payment, the funds will then be charged to Frank's credit card.

Some time after Frank leases the car, there is a problem with his PAC payment. The payment request to the bank is rejected and the reason provided is non-sufficient funds (NSF). BMW (Canada) Leasing records the rejected PAC payment, then debits Frank's credit card for the amount of the outstanding payment, plus any applicable service charges. Frank is sent an e-mail that outlines the details of the rejected PAC and the debit to his credit card.

Managing correspondence: Mr. Donald Crump turned 65 years old on March 12th and, prior to that, on March 2nd, converted one of his RRSPs into a last to die income annuity with his spouse. Based on these two events, the retirement survival kit event is recorded and a kit is mailed out. Note that turning 65 or converting the RRSP events were both recorded but did not generate any correspondence to the customer until both situations had occurred.

Example 1: The Bring a Smile Card and Gift Company has decided that sending a yearly birthday greeting card to all of their customers is too expensive. Bring a Smile Card and Gift Company will instead send birthday cards only on special birthdays. The Customers will now receive birthday greeting when they reach an age ending with a zero, providing they are still living and they are at least 10 years old.

Example 2: Miss Valerie Falcon is a frequent flyer with Fly Like an Eagle Airlines. She arrives at the airport to check in for a flight to Paris, France. At check-in, she hands her ticket and her frequent flyer miles card to the representative. The Representative records the miles on Miss Falcon's card. With these miles, Miss Falcon now qualifies for the incentive program. In Miss Falcon's case, she will receive a silver status card as well as a free economy ticket anywhere in North America because she surpassed the 20,000 miles in a 180 day period.

In this scenario, the Airline business system transaction was sent to the Event Manager for evaluation via the Services Layer. The Event Detection module executed all the event rules for Miss Valerie Falcon and detected that her Silver Status event rule is evaluated to true today. As a result, the information about the occurrence of this event was persisted and notification was posted. The Airline business system picked up the notification and sent Miss Falcon a congratulatory letter.

Event Manager and Evergreening configuration behavior

Behavior When Configured On
Data Evergreening and event management are available

Behavior When Configured Off
Data Evergreening and event management are not available.

Behavior When Configuration is Changed in a Production Environment
Event Manager and Evergreening can be configured on or off at any time without impact other than changing the use of those features.

Configured During Installation
Event Manager and Evergreening can be installed and configured manually at any time.

Modifying This Feature
Event Manager and Evergreening can be modified according to the procedures in the *Configuring real-time and offline SDP using InfoSphere MDM Server Evergreening* topic in the *IBM InfoSphere Master Data Management Server Developers Guide*.

Interactions

The IBM InfoSphere Master Data Management Server interactions feature redefines, or flattens, how interactions and their regarding details are stored and retrieved. Regarding details are the information related to what an interaction is about. Only one regarding detail can be captured per interaction. If more than one regarding detail is needed, all other regarding details must be captured in separate interactions.

Interaction example use

A client calls to inform you that he is moving and changing his phone number. Separate interactions with regarding details are created to capture the fact that the address has changed and that the phone number has changed.

Interactions configuration behavior

Interactions are always available in the InfoSphere MDM Server product. They do not need to be separately installed or configured in order for you to use them.

The Interaction feature can be modified according to the procedures in the *IBM InfoSphere Master Data Management Server Developers Guide*.

Language and locale customization

You can define the language and locale for your implementation of InfoSphere MDM Server.

At the application level, InfoSphere MDM Server globalization support allows for business and system error messages to be stored and retrieved based on a specific language or locale.

At the operational data level, InfoSphere MDM Server globalization support allows for code table (reference) data to be stored and retrieved based on a specific language or locale.

The Product domain extends this support by allowing product data to be maintained in multiple languages for both hard attributes (that is, data defined within physical database tables) and also soft attributes (that is, data defined within a spec). This product data can be maintained, retrieved, and even searched for based on a specific language or locale.

InfoSphere MDM Server globalization support enables InfoSphere MDM Server to be deployed and used across various geographies.

Some of the highlights of the globalization support include:

- A single executable code that is used for all supported locales.
- A single deployment that can support multiple locales simultaneously. The installation allows the installer to select additional languages to be deployed, in addition to the default English language.
- In addition to the default English language, translations for locale-sensitive strings are provided for the following languages:
  - French
  - German
  - Italian
  - Spanish
  - Brazilian Portuguese
  - Polish
  - Simplified Chinese
  - Traditional Chinese
  - Korean
  - Japanese.
- Operational data can be provided in any language in transactions, not just the languages listed above. InfoSphere MDM Server uses Unicode, which enables data to flow through the system without any loss or corruption.

Language and locale customization example use

XYZ Insurance Company must install InfoSphere MDM Server in several countries, including Korea, the United States, and Spain. The installers in each country can install the exact same code, needing only to define the languages and locale during the installation process. Each InfoSphere MDM Server installation will be properly localized according to the local language.

Language and locale customization configuration behavior

The language and locale customization feature is set up during installation and cannot be modified without reinstalling.

Name and Address Standardization

You can use a standardizer to ensure that names, addresses and phone numbers are stored in InfoSphere MDM Server using the same format. Depending on configuration, the standardizer can also normalize address and phone number information. You can use the default standardization that comes with MDM Server, or you can use standardizers from IBM IIS QualityStage or Trillium. You can also add a third-party standardizer, if required.

For information on normalization, see “Normalization” on page 60.

Name and Address Standardization example use

When you enter a party’s name as `Bob`:

- if standardization is on—the party name is stored in the SEARCH table as `Robert`; a search for `Bob`, `Robert` or `Rob` returns this party
- if standardization is off—the party name is stored as `Bob`; only a search for `"Bob"` returns this party
- if standardization was on when the party name was stored but off when the party name was searched for—the party name is stored as `Robert`; only a search for `Robert` returns this party
- if standardization was off when the party name was stored but on when the party name was searched for—the party name is stored as `"Bob`; only a search for `"Bob` returns this party

Name and address standardization configuration behavior

Behavior When Configured On
Name, address and phone number data is standardized before being added to the database.

Behavior When Configured Off
Name, address and phone number data is added to the data base in the format that it is entered.

Behavior When Configuration is Changed in a Production Environment
Standardization can be configured on or off at any time; however, turning it off affects how data is stored and how searches are executed, as described in the example.

Configured During Installation
Standardization can be configured at any time, however this will impact your subsequent searches.

Modifying This Feature
Standardization can be modified according to the procedures in the

Standardizing name, address and phone number information section of the *IBM InfoSphere Master Data Management Server Developers Guide*.

Party CDC processing

The Party Critical Data Change (CDC) processing business feature enables review and approval of changes to values that are defined as critical data before committing the changes to the party record.

Critical data elements are user-defined. By default, the following data elements are defined as critical:

- In a Person object:
  - Last Name
  - First Name
  - Gender
  - Date of Birth
  - Social Security/Social Insurance number
  - Address
- In an Organization object:
  - Organization Name
  - Tax Identification Number
  - Address

If Party CDC Processing is configured on and IBM InfoSphere Master Data Management Server recognizes any critical data elements in an update transaction, the party record will not immediately be updated. Instead, a record of the CDC request is created and given a status of ″pending″.

Upon creation of a pending CDC, the pending Critical Data Change indicator turns to `Y` on the party record. This party is thus prohibited from collapsing, splitting, and further critical data changes until this indicator turns to `N` upon resolution of all the pending changes.

A single change request can consist of multiple CDCs that can belong to more than one business object. Each CDC has a unique identifier that allows the changes to each business object to be dealt with separately. If all CDCs in a transaction apply to the same object, they must be processed as a whole and cannot be dealt with separately. For example, if an update transaction includes changes to Gender, Date of Birth, and Business Address, then the Gender and Date of Birth changes must be considered together since they both belong to the same TCRMPersonBObj, while the Business Address change can be considered separately.

Party CDC Processing also applies to fine-grained update party transactions, such as updatePersonName, updateOrganizationName, updatePartyAddress, and updatePartyIdentification. This process can be bypassed using the updatePartyCriticalData transaction, which allows you to immediately update the party record, without requiring a ″pending″ status.

Important: Use of the updatePartyCriticalData transaction should be restricted to users with data stewardship responsibility.

Party CDC processing example use

Party CDC processing can be used to change critical party information such as a birth date or a party's address. These pieces of critical data may have an impact on other lines of business associated with the party.

Party CDC processing configuration behavior

Behavior When Configured On
Updates to any of the defined critical data is put on hold pending acceptance. A notification of the CDC request is generated, and the pending critical data change indicator is changed to `Y` on the party record. The party record remains unchanged until the status of the CDC is updated. The pending indicator prohibits the party from collapsing with other parties, splitting into new parties, or receiving further updates to critical data or noncritical data belonging to the same business object as the pending CDC.

Behavior When Configured Off
Updates to party data, both critical and noncritical, are processed immediately upon submission of a transaction request. If configured, a notification can be generated upon a change to critical data.

Behavior When Configuration is Changed in a Production Environment
Party CDC Processing is configured at the application level upon implementation. Any changes to the configuration will impact all update party transactions.

Configured During Installation
Critical data change processing is configured at the application level upon implementation.

Modifying This Feature
Critical data elements are user-defined. For details about modifying the definition of critical data, see the *IBM InfoSphere Master Data Management Server Developers Guide*.

Party searches in IBM InfoSphere Master Data Management Server

IBM InfoSphere Master Data Management Server provides the ability to customize the search feature, so clients can either:

- Write their own SQL statements to execute searches—This method allows you to define exactly what you want to search for
- Use the predefined existing search methods—This method searches using the optimized query that most closely matches the search request.

Using customized searches can return more specific results but the searches are slower. Using the built-in InfoSphere MDM Server search returns results faster, but the results may not be as precise as a customized search.

InfoSphere MDM Server also enables you to perform special searches using the extended search functionality.

Related concepts

"Common name exclusion search" on page 40

"Partial criteria searches" on page 41

"Phonetic searches" on page 42

"Search party by admin system key" on page 43
"Search by party macro role" on page 44

Search example use

Search can be configured to create specific searches that are not predefined. For example, you need to ensure that the same social security number has not been entered for two different parties. The predefined search that comes with IBM InfoSphere Master Data Management Server searches only for an SSN and ignores last names, because an SSN is supposed to be a unique identifier. In this case you need to create a customized search for both a last name and social security number so the results will show both the duplicated SSNs and the names of the parties with the duplicate SSNs.

Search configuration behavior

Behavior When Configured On
Search can be configured to return specific results. This offers flexibility but the results are returned slower.

Behavior When Configured Off
Search returns predefined results. These results are returned faster, but may not meet the users specific needs.

Behavior When Configuration is Changed in a Production Environment
Search can be configured on or off at any time without impact other than changing how search works.

Configured During Installation
Search can be configured at any time.

Modifying This Feature
Search can be modified according to the procedures in the *Configuring Party Search* topic in the *IBM InfoSphere Master Data Management Server Developers Guide*.

Common name exclusion search

The common name exclusion search feature provides the ability to search criteria exclusions, such as common last names, when search for Person objects.

With this feature, long-running queries and searches that yield large, meaningless result sets can be prevented. Specifically, searches based on the following exclusion rules can be prohibited:

- common last names
- common last names with certain given names
- common last names in selected cities

Common names are those names that occur within the InfoSphere MDM Server database more than a specified number of times; this exclusion threshold is configurable. In addition, phonetic variations and standardized names are taken into consideration when forming the above exclusion rules.

The common name exclusion rules can be automatically populated by executing a simple script. For details, see the *IBM InfoSphere Master Data Management Server Developers Guide*.

Common name exclusion search example use:

A company might wish to avoid searches using criteria with common last names, common last names paired with common first names, and common last names within a given city. For example, if XYZ Insurance Company's InfoSphere MDM Server database contains 10,000 parties with a last name `Smith` and their exclusion threshold is less than 10,000, then a searchPerson or searchParty transaction with exclusive search criteria of `Last Name = Smith` would result in an error response. In order to search for a party with the last name Smith, more search criteria would have to be provided.

Common name exclusion search configuration behavior:

Behavior When Configured On
If search criteria match those found in the common name exclusion set, or if they represent an even broader search using wildcard or look-alike characters, then the search will not execute and the transaction will fail.

Behavior When Configured Off
No exclusions are applied, and the search function behaves normally.

Behavior When Configuration is Changed in a Production Environment
Common name exclusion search can be configured on or off at any time without impact other than changing whether or not the search feature searches criteria exclusions.

Configured During Installation
During installation, a script should be run to populate the criteria exclusion set. The Common Name Exclusion search functionality can be configured on or off at any time, at either a global or user level.

Modifying This Feature
Search can be modified according to the procedures in the *IBM InfoSphere Master Data Management Server Developers Guide*.

Partial criteria searches

The Partial Criteria Search feature provides the ability to perform various forms of searches based on partial search criteria, such as wildcard and look-alike characters. The wildcard character (%) represents any combination of zero or more characters (a string). The look-alike character (?) represents a single character.

InfoSphere MDM Server search services support partial criteria searches across several fields, such as:

- Last Name
- Organization Name
- Given Name One
- Identification
- Contact Method
- ZIP/Postal Code
- and others

Partial criteria search example use:

You may wish to search for a party using a partial telephone number or address element criteria. For example, if you wish to search for a party, but do not know the last four digits of the party's telephone number, you can use a search criterion of ″416-555%″. Similarly, you can search for a party knowing only specific characters of their ZIP code using a search criterion of ″9?2?0″.

Partial criteria search configuration behavior:

Behavior When Configured On
On an per-search field basis, you can configure the required minimum number of non-wildcard or look-alike characters. Any searches whose criteria does not contain the minimum number of non-wildcard or look-alike characters will fail.

Behavior When Configured Off
Search criteria can include any number of non-wildcard or look-alike characters.

Behavior When Configuration is Changed in a Production Environment
Partial criteria search can be configured on or off at any time without impact other than changing whether or not the search feature searches using partial search criteria.

Configured During Installation
The minimum number of non-wildcard or look-alike characters required in a partial criteria search can be configured on or off at any time, for any specific search field.

Modifying This Feature
Partial Criteria Search can be modified according to the procedures in the *IBM InfoSphere Master Data Management Server Developers Guide*.

Phonetic searches

The Phonetic Search feature provides the ability to perform party searches based on phonetic ("sounds like") name and address search criteria.

The results of a phonetic search include both exact and phonetic matches. The score for a phonetic match is calculated by multiplying the exact score by a phonetic weighting factor.

If name standardization is turned on, the search criteria are matched phonetically with the standardized name.

Phonetic searching is supported for a range of Latin-based languages, specifically:

- Western European languages
- Eastern European languages
- Slavic languages

Phonetic searching is not supported on search criteria containing wildcard (%) or look-alike (?) characters.

Phonetic search example use:

You may wish to search for a party using phonetic organization name criteria, phonetic last name criteria, phonetic given name criteria, or phonetic city criteria. For example, you can search for the party `Stephen Leighton` using search criteria of `Steven Layton`.

Phonetic searches configuration behavior:

Behavior When Configured On
Search result set includes both exact and phonetic matches.

Behavior When Configured Off
Search result set includes only exact matches.

Behavior When Configuration is Changed in a Production Environment
The phonetic search feature can be configured on or off at any time without impact other than changing whether or not the search feature includes phonetic matches in search results.

Configured During Installation
Phonetic name searching or phonetic address (city) searching can be configured on or off at any time, at either a global or user level. You also have the ability to configure the phonetic key length at a global level for address and name items.

Modifying This Feature
The phonetic search feature can be modified according to the procedures in the *IBM InfoSphere Master Data Management Server Developers Guide*.

Search party by admin system key

The Search Party by Admin System Key feature provides the ability to perform various forms of party searches based on full or partial administrative system key values. Partial administrative system keys can be used as search criterion by including a wildcard and look-alike characters. The wildcard character (%) represents any combination of zero or more characters (a string). The look-alike character (?) represents a single character.

This feature enables searches for a party, regardless of whether it is a Person or an Organization, by basing the search on an AdminPartyId. AdminPartyIds are identifiers maintained by an external administrative system that integrates with InfoSphere MDM Server.

In a business environment that captures customer information, there may be more than one administrative system coexisting. Using an AdminPartyId, or part of one, InfoSphere MDM Server can search the database to find a corresponding party record that contains all information pertaining to the customer, regardless of where the information was captured.

Search party by admin system key example use:

A customer service representative at XYZ Insurance company needs to search for a party, but only knows the first two digits of the party identifier. By inputting ″22%″, the CSR can view all party records that begin with the digits ″22″. Similarly, a CSR can search for a party knowing only certain numbers within the identifier by using a search criterion of ″1%3?3″.

Search party by admin system key configuration behavior:

Behavior When Configured On
Users can search by Admin System Keys. At least one non-wildcard/look-alike character is required in combination with the wildcard or look-alike character. Any search whose criteria does not contain the minimum number of non-wildcard/look-alike characters will fail.

Behavior When Configured Off
Search functionality behaves normally.

Behavior When Configuration is Changed in a Production Environment
Search party by admin system key can be configured on or off at any time without impact other than changing whether or not the search feature searches by Admin System Keys.

Configured During Installation
Search Party by Admin System Key can be configured at any time.

Modifying This Feature
Search can be modified according to the procedures in the *IBM InfoSphere Master Data Management Server Developers Guide*.

Search by party macro role

The search by party macro role feature provides the ability to search for a party based on the party macro role, plus other search criteria.

When searching for a party by macro role, the party macro role must match the macro role search criteria. For each search criteria type provided, a corresponding party macro role association (such as Name, Address, Contact Method, Identifier, and Party Equivalency) for that party macro role must exist, otherwise the search will fail.

The result set only includes Child Address, Contact Method, Identification, and Name information that matches both the existing party macro role associations and the usage types specified in the properties file. If more than one instance occurs that matches the above conditions, then the first instance is returned in the basic party data (inquiry level 0).

Party macro role search example use:

A customer service representative from XYZ Insurance company needs to search for a party within the context of a specific party macro role. Specifically, the CSR can use the Party Macro Role search to locate a Person party with the last name `Peterson` within the party macro role context of `Prospect`.

Party macro role search configuration behavior:

Behavior When Configured On
Users can search by party macro roles. At least one non-wildcard or look-alike character is required in combination with the wildcard or look-alike character. Any search whose criteria does not contain the minimum number of non-wildcard or look-alike characters will fail.

Behavior When Configured Off
Search functionality behaves normally.

Behavior When Configuration is Changed in a Production Environment
Party macro role search can be configured on or off at any time without impact other than changing whether or not the search feature searches by party macro roles.

Configured During Installation
Party Macro Role Search can be configured on or off at any time.

Modifying This Feature
Search can be modified according to the procedures in the *IBM InfoSphere Master Data Management Server Developers Guide*.

Party deletion

IBM InfoSphere Master Data Management Server can delete a party, and its related child objects and associated entities, from ODS and the history tables.

Party deletion example use

If a party no longer has a business relationship with your company, you can delete all records and all history for that party from your data.

Party deletion configuration behavior

The ability to delete parties is always available in the InfoSphere MDM Server product. The feature does not need to be separately installed or configured in order for you to use it.

Delete capability can be modified according to the procedures in the *Deleting party information from IBM InfoSphere Master Data Management Server* topic in the *IBM InfoSphere Master Data Management Server Developers Guide*.

Rules of Visibility and Data Persistency Entitlements

The Rules of Visibility and Data Persistency Entitlements modules are used to assign what the user is entitled to see and do (in other words, what the user can view, add and update). These entitlements depend on the data values on a transactional basis, rather than which transactions the user can execute. The rules are defined in the System Maintenance Security option.

The two types of data level entitlements discussed in this manual, Data Persistency Entitlements and Rules of Visibility, are closely linked. The Rules of Visibility (RoV) module allows administrators to define what elements and sub-elements users and user groups can see based on defined constraints. At runtime, the rules are evaluated at post-transaction, and transaction data sets are filtered based on the defined rules. For more information about defining RoV, see the *IBM InfoSphere Master Data Management Server System Management Guide*.

The Data Persistency Entitlements module allows administrators to define the elements and sub-elements that users and user groups can add and update, based on defined constraints. At runtime, the rules are evaluated pre-transaction to determine if the user has been entitled to add or update the data within the transaction set based on the defined rules.

Rules of Visibility example use

Using Rules of Visibility and Data Persistency Entitlements, you can add a new employee to a group and define what that group is allowed to do. For example, you can specify that the group can add new parties, changing party information, and perform queries. Using Rules of Visibility you can define what that group can see. For example, you might allow members of the group to run a search to return party information, but through the Rules of Visibility settings for the group, you can ensure that the search results will show the party address, but not the party's annual income.

Rules of Visibility and Data Persistency configuration behavior

Behavior When Configured On
Employees can be added to groups with specified access to party information and specified actions that they can perform.

Behavior When Configured Off
Any employee with security access to the transaction can view and change party information.

Behavior When Configuration is Changed in a Production Environment
Rules of Visibility and Data Persistency configuration can be changed at any time without impact other than changing the viewing capability and ability to change party information.

Configured During Installation
Rules of Visibility can be enabled or disabled at any time.

Modifying This Feature
Rules of Visibility and Data Persistency can be modified according to the procedures in the *Setting Rules of Visibility* topic in the *IBM InfoSphere Master Data Management Server Developers Guide*.

Suspect Duplicate Processing

The Suspect Duplicate Processing feature in IBM InfoSphere Master Data Management Server provides a mechanism to keep a ″golden″ copy of a given party. It works by searching and matching existing parties that are duplicates of the party being added or updated, and then associating each of the duplicate parties—the suspects—with the input party.

When you are adding a new party, if an existing party is found that is an exact match and there are no pending critical data changes for the existing party, then the new party is used to update the data in the existing party instead of adding the new party.

Definitions of terms related to Suspect Duplicate Processing

Several specific terms are used when discussing Suspect Duplicate Processing and it is important to understand them.

The following terms are used when discussing suspect processing:

`AddParty` control attributes
There are some attributes on the party business object (TCRMPartyBObj) that are not persisted with the party but are used as informational attributes in the request and response of transactions that add a party, such as `AddParty` and `AddContract`.

The `AddPartyStatus` flag is the result of adding a party when the transaction was successful. The status can be one of the following values:

- 1—New party created - no suspects found
- 2—New party created - suspects found
- 3—Existing party used - A1 match
- 4—Existing party used with pending data - A1 match
- 6—Existing party used with pending data - A2 match
- 7—A2 (and B) matched parties returned - party not added
- 8—Existing party used – best A1 match selected
- 9—Party not yet added – transaction cancelled (when adding multiple parties through addcontract when A2 parties found for a different source party)
- 11—New Party Created — A1 Match is not used because it has pending critical data change

The `MandatorySearchDone` flag indicates whether or not a mandatory suspect duplicate party search has been completed as party of the process of adding the party. This indicator is used in conjunction with the `/IBM/Party/SuspectProcessing/AddParty/returnSuspect` configuration

option which determines whether or not A2 suspects that are found should be returned to the user for investigation prior to adding the party.

If this flag is set to No on the incoming transaction request and the `returnSuspect` option is configured to Yes and if A2 parties are found and no A1 parties then the A2 parties are returned to the user, no new party is added. The transaction is essentially halted and the `AddPartyStatus` attribute on the party business object will have a value of 7, as described above.

If this flag is set to Yes on the incoming transaction request then the above logic is bypassed. In other words, it is an indicator that the search has been completed, A2 suspect duplicate parties have been return to the user for investigation and the user has resubmitted the request.

If this flag is set to Yes, then a mandatory search has been completed as part of adding the party. This attribute becomes the state information that is passed in and out of the `addParty()` operation to indicate where the operation is as part of adding the party.

The **SearchPartyDone** field indicates whether or not a suspect duplicate party search was performed for the given party prior to the request to add the party. If yes and one or more suspects were found, then the party object should contain a list of suspect objects that were passed in with the request to InfoSphere MDM Server.

If this field indicates that the suspect search was completed it does not imply that the party was matched against the suspects found and therefore party matching still proceeds internally as part of the add party request. However, the step to retrieve candidates from the database (suspect duplicate party search) is skipped. This field is introduced for performance reasons; it controls whether or not a suspect duplicate party search is executed internally as part of adding a party.

Critical Data

These are key elements of a party that are used as criteria in suspect duplicate party searching and party matching. In other words, critical data is used for finding suspect duplicate parties and then weighing how closely each suspect matches and doesn't match. Examples include name elements, address and tax identifier.

Evergreening

Evergreening is the process of continuously monitoring and improving the quality of data, for example, by identifying suspect duplicate parties. It is possible run suspect duplicate processing in a real-time mode, during an AddParty transaction, or in an off-line mode using the Event Manager component. Also, based on implementation requirements, a combination of the two can be used in order to achieve performance objectives and still manage data quality during online transactions. For more information, the topic *Suspect Duplicate Processing configuration points* in the *IBM InfoSphere Master Data Management Server Developers Guide*.

Match Category, or Suspect Type

Match category is derived based on the combination of match and non-match relevancies. The following are the match categories defined within the product:

- A1: 100% confident the two parties are the same
- A2: Reasonably sure the two parties are the same

- B: Not sure if the two parties are the same, most likely due to insufficient data for matching
- C: Confident two parties are not the same

Match Category Adjustment
A match category can be adjusted up or down (for example, from an A2 to A1) based on additional rules.

Example 1: If two female persons have the same first name, date of birth and address but different last names, perhaps an upgrade from a B to A2 is required as this may fit a marriage scenario and should be investigated.

Example 2: The match category may be adjusted if integrating with IBM Information Server QualityStage to augment the InfoSphere MDM Server deterministic matching with a probabilistic match score.

Match Relevancy
Match relevancy is the critical data elements that matched between two parties.

Matching Engine
Different matching engines can be used for party matching. Examples include the InfoSphere MDM Server default deterministic matching engine and the IBM Information Server QualityStage probabilistic matching engine.

Non-Match Relevancy
Non-match relevancy is the critical data elements that did not match between two parties. In the data model, this is also known as the "suspect reason".

Suspect Action
A match category directs the course of action taken with particular transaction types.

Example 1: When adding a new party, if that party matches to an existing party as an A1, then the transaction can turn into an update of the existing party.

Example 2: When updating an existing party, if that party now matches another existing party as an A1, create a suspect entry between the two and notify the user.

Suspect Augmentation
Multiple matching engines can contribute to details of a particular suspect entry. For example, IBM Information Server QualityStage can be used to augment the InfoSphere MDM Server default deterministic match results, and the he suspect entry contains the results of both.

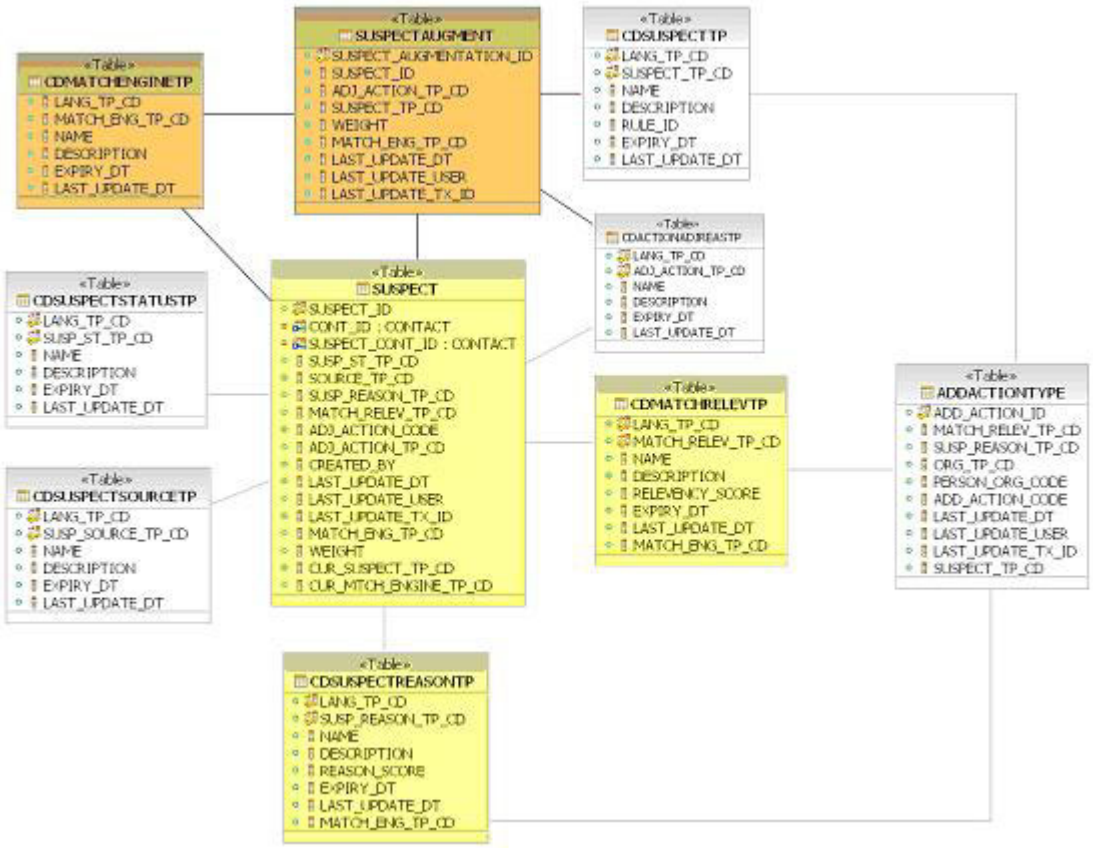
Suspect Entry
A suspect entry is a record between two parties that identifies that the parties may be suspect duplicates of each other. Details on a suspect entry include:

- The parties that are suspect duplicates
- Matching engine used to produce the match results
- Match relevancy
- Non-match relevancy
- Match category (derived)
- Match category adjustment and reason

- Weight, which is probabilistic match score if supported by the matching engine used
- Status
- Who created the suspect entry, either user or system

Suspect Processing data model

This data model shows the relation of many of the defined concepts:



Suspect Status

A suspect entry can go through a number of different states in its lifecycle. The states are:

Under Investigation – Parties Suspect Duplicate

Two parties are suspected to be duplicates and it is assumed they are under investigation to see if they are in fact duplicates. If so, they may be collapsed. If not, they may be marked as not duplicates.

Under Investigation – Pending Critical Data Change

Two parties are strongly suspected to be duplicates however there is a critical data difference between the two that must first be resolved before they can be collapsed together. For example, perhaps all elements match except the address. In this scenario, a change of address is expected and must be resolved.

Investigated – Parties Not Duplicate

An investigation revealed that two parties marked as being suspect duplicates are in fact not duplicates.

Investigated – Critical Data Change Resolved
Two parties that were strongly suspected to be duplicates have been investigated and any critical data elements in conflict have been resolved.

Parties Duplicate – Do Not Collapse
Two parties are known to be duplicates of each other but must be kept as separate and distinct party records and should never be collapsed together.

Third Party Integration
The InfoSphere MDM Server SDP feature supports using third party tools. InfoSphere MDM Server supports IBM Information Server QualityStage, which can be used either to:

- Augment the InfoSphere MDM Server default deterministic match results in a near-realtime fashion with a probabilistic match that can be used to either upgrade or downgrade a match
- Replace the InfoSphere MDM Server default deterministic matching engine with a probabilistic matching engine

Suspect category names and descriptions

Default suspect categories in the product are defined as follows:

Suspect Category	Name	Category Description
1	A1	Match and non-match relevancy scores indicate that a definite duplicate party has been found. A type 1 suspect category is guaranteed to be a duplicate, with 100% confident that the parties are the same.
2	A2	Match and non-match relevancy scores indicate that the suspect party found has a high probability of being a duplicate. A type 2 suspect category indicates that it is reasonably likely that two parties are the same.
3	B	Match and non-match relevancy scores indicate that the suspect party found might be duplicate. A type 3 suspect category indicates that it is fairly unlikely that the two parties are the same.
4	C	Match and non-match relevancy scores indicate that the suspect party found is not a duplicate. A type 4 suspect category indicates that it is definite that two parties are not the same.

How the Suspect Duplicate Processing feature works

For details about how Suspect Duplicate Processing works, see:

- "Adding a party"
- "Adding additional critical data to a party or updating existing critical data" on page 51

Adding a party: Adding a new party triggers suspect duplicate searching and matching. The results of these operations determine the actions taken within the

add party transaction. These actions within the add party transaction rely on externalized rules so that clients can define their own add actions and suspect processing actions:

Example 1

In a case where suspects of category type 2 have been found, creating a suspect entry identifies the active parties as suspects of each other. The actions taken might be to add the party to the database (an add action) then to create a suspect entry for each potential duplicate party found (a suspect processing action).

Example 2

In a case where one suspect of category type 1 is found, and it does not have any pending critical data changes, the actions taken might be to update the party found as suspect (an add action) rather than adding a new party and, because in this case the suspect is guaranteed to be duplicate to the source party, to not create a suspect entry (suspect processing action).

Adding additional critical data to a party or updating existing critical data: Adding a new critical data element to a party, or changing an existing critical data element, results in suspects being re-identified. Defining which party elements are critical is external to IBM InfoSphere Master Data Management Server. The core of the product relies on a number of externalized rules so that each client can define their own critical data, determine how to search for suspect duplicate parties, how to match parties, and so on. Additionally, the external validation component can be used to identify which critical data elements are mandatory in different transactions, such as adding a party.

Suspect Duplicate Processing example use

Suspect processing can be modified to include specific information in the party matching. For example, Acxiom AbiliTec consumer and business links that uniquely identify a person and an organization type parties can be added to the suspect processing party matching.

Suspect Duplicate Processing configuration behavior

Behavior When Configured On

New parties being are compared to parties that already exist in the database to ensure that there is only one record of information for each party.

Behavior When Configured Off

New parties are not compared to existing parties.

Behavior When Configuration is Changed in a Production Environment

When Suspect Processing is configured on, the system starts identifying suspects. When configured off, the system stops identifying suspects.

Configured During Installation

Suspect processing can be configured at any time.

Modifying This Feature

Suspect processing can be modified according to the procedures in the *Configuring Suspect Duplicate Processing* section of the *IBM InfoSphere Master Data Management Server Developers Guide*.

Task Management Services

The Task Management Services feature addresses the need for entity lifecycle management, as well as orchestration and business process management. The Task Management Services feature manages the life cycle of tasks, provides task management services to other components, and provides a runtime execution environment for each task.

Task management supports generic task-oriented design. It provides services for task lifecycle management and a generic runtime execution environment for any task in the InfoSphere MDM Server system. Tasks and the Workbasket are used collectively to define the tasks designated for a user to achieving a particular activity in a business process.

Task Management Services example use

Ryan, a model designer, uses the UI to define three task definitions in an InfoSphere MDM Server solution:

- Type 1: Add Item
- Type 2: Enrich Item with Marketing Info
- Type 3: Enrich Item with Financial Info

In a business process, Dale, a business administrator, uses the Administrative UI to create two task instances based on the solution data/business model that Ryan created

Patricia, a business manager for new item registration, logs in to the Business UI Task Inbox and finds that there two new tasks, Task1 and Task2 on her list. She assigns both Task1 and Task2 to Sara, who is a business analyst.

- Task1: Add a new laptop (SKU# 987001) to the inventory, using task type `Type 1`
- Task2: Add another new laptop (SKU# 987002) to the inventory, using task type `Type 1`

Sara logs in to the Business UI and finds Task1 and Task2 on her list. She launches Task1. After running through the UI actions, which are defined in the activity diagram of a task definition, she comes back to Business UI and sets Task1 to `Success`. Then she does the same for Task2.

After the completion of Task1 and Task2, Dale, the business administrator, creates the following task instances:

- Task3: Enrich the first laptop (SKU# 987001) with marketing info, using task type `Type 2`
- Task4: Enrich the second laptop (SKU# 987002) with financial info, using task type `Type 3`

After creating these task instances, Dale assigns Task3 to Ivan, a marketing manager, and he assigns Task4 to Theresa, a finance manager.

Task Management Services configuration behavior

Task Management Services are always available in the InfoSphere MDM Server product. They do not need to be separately installed or configured in order for you to use them.

Task Management Services can be modified according to the procedures in the *Modifying task management* topic in the *IBM InfoSphere Master Data Management Server Developers Guide*.

Chapter 3. The Party domain

The Party domain manages the entirety of data related to parties such as customers, vendors and suppliers, and maintains a single, consistent version of this data.

Party is the central entity of the domain. Each party is unique and is stored only once. However, each party can play one or more roles. A party has relationships to other objects, such as contracts, locations, other parties.

Related concepts

"InfoSphere MDM Server domains" on page 1
IBM InfoSphere Master Data Management Server enables companies to extract maximum value from master data by centralizing multiple data domains and providing a comprehensive set of prebuilt business services that support a full range of master data management (MDM) functionality.

Aggregated party view

The aggregated party view business feature enables you to view an aggregated party record created from aggregated data about a party, from across a number of duplicate profiles, instances, or exact matches.

An exact match is an A1 match with a suspect status of 6 (parties are duplicates and collapse is not permitted).

For example, an aggregated party view could be a collection of all the resolved conflicting party data across exact matches, based on data rules of survivorship.

Aggregated party view example use

The aggregated party view feature can be used in the following ways:

Single party ID
: You want to view a record that shows all the information that is different for the suspect duplicates of a particular party. You provide a party ID, and the transaction returns a single, aggregated party record view of all suspects in the database that are linked to the provided party, with a special suspect status of 6 "Parties are Duplicates - Collapse not permitted". This aggregated view shows the collected differences between the related suspect parties.

Multiple party IDs
: You want to view a record that shows the differences between several parties. You provide a list of parties to be viewed as a single party, and the transaction returns a single, aggregated party record view that contains the resolved data of all the parties. This aggregated view shows the differences for the provided parties.

Campaigns

The campaigns business feature stores and retrieves information regarding marketing campaigns.

A marketing campaign promotes awareness of something—products, information, parties—and its target audience can be a person or an organization. Marketing campaign information about products and other business functions such as fee changes can be associated with one or more parties.

Campaign implementation resides behind an interface that provides business services to add, update and retrieve campaign details.

Campaigns example use

A financial institution might use the campaign feature to associate information about a special interest rate to parties who qualify for that rate. Once this is done, a customer service representative can then present the offer of the special interest rate to the qualified parties.

Financial profile

The financial profile business feature is a collection of details for a party, including income source and all payment source entities.

Payment source is the super-type of charge card, bank account, and payroll deduction.

Financial profile example use

Financial profiles can be used to show all income sources, payroll deductions, bank accounts and charge cards for a given party.

Grouping

The grouping business feature identifies a collection of items with a common thread. The Grouping feature allows you to capture and manage groupings of entities within the IBM InfoSphere Master Data Management Server product, and to associate miscellaneous values, addresses and contact methods to a grouping.

You can also refine retrieving addresses and contact methods by searching for a particular grouping, such as only those of a particular role or type. For example, you can configure a search to return the primary address—the address usage type—for the head of household—the role type.

Party grouping services can be used to create, update, or inquire on Party groupings and Party grouping associations. There is no limit for the number of Party grouping associations that can be created in a Party grouping.

The effective start date value for a Party grouping association must be after or on the corresponding Party grouping effective start date. Similarly, the effective end date value for a Party grouping association must be before or on the corresponding grouping effective end date.

All active Party grouping associations associated with an expiring Party grouping will expire with the same date. After a Party grouping has expired, no Party grouping associations can be added or updated for that Party grouping.

Generic grouping services can be used when creating, updating, or inquiring on grouping for entities listed in the `EntityNameInstancePK.properties` file. The list includes:

- CONTACT
- CONTRACT
- PERSON
- ORG
- GROUPING
- CONTRACTROLE
- ADDRESS
- PERSONNAME
- ORGNAME
- CONTACTMETHOD
- IDENTIFIER
- CONTEQUIV
- ALERT
- BANKACCOUNT
- CHARGECARD
- INCOMESOURCE
- PAYROLLDEDUCTION
- PRIVPREF
- CONTACTREL
- MISCVALUE
- LOBREL
- ADDRESSGROUP
- CONTACTMETHODGROUP

Generic Grouping services have similar functionality with PartyGrouping services.

You can add, update and query party groupings through the IBM InfoSphere Master Data Management Server Data Stewardship interface. See the *Data Stewardship User Interface* for more information.

Grouping example use

Grouping could be use to show:

- A Household group that is comprised of parties who live at the same address
- A High Health Risk group of males over the age of 40 who smoke 2 packs of cigarettes a day
- A group of Unsolicited Internet Homes that have yet to receive high speed internet cable access in their neighborhoods

Hierarchy

The Hierarchy business feature services manage generic hierarchies in the system.

Hierarchy services enable users to search for hierarchies, create hierarchies, update them, add and to update individual nodes, their parent or child relationships, and the roles the nodes have in the hierarchy. Additionally, hierarchy services allows

you to perform inquiries on the entire hierarchy or just a section of it. Hierarchy services also provide users with the ability to search for a party entity type node by name or by role.

The following terms are used when discussing the hierarchy feature:

Hierarchy
A structure that contains two or more entities with parent-child relationships.

Node An entity in the hierarchy.

Root Node
An entity in the hierarchy that is the top-most parent in a given branch of a hierarchy.

Hierarchy Relationship
A parent-child relationship between two nodes in the hierarchy.

Ultimate Parent
A node that is the top-most parent in the hierarchy. A hierarchy can only have one ultimate parent at a time.

Ancestors
All nodes that are parents, directly or indirectly, of a node are collectively called ancestors of that node.

Descendents
All nodes that are children, directly or indirectly, of a node are collectively called descendents of that node.

Hierarchy example use

Hierarchy can be used to show the organizational structure of the legal entities that make up international companies. Hierarchy can show the international parent, local parent, divisions and all other levels of the organization. Hierarchy can also be used to show the responsibility and level of people within an organization: director; manager; team leads; team members, and others.

A marketing company might want to find a person that has a specific responsibility within a hierarchy: for example, a chief executive officer, director, or manager.

Know Your Customer

The Know Your Customer (KYC) feature provides a generic framework that can be used to provide paginated returns for newly-developed search or getAll transactions.

Compliance
Maintains compliance requirements and party compliance information, and schedules revalidation of compliance.

Questionnaire
Maintains questionnaires and a party's response to each questionnaire.

Know Your Customer example use

The Know Your Customer feature can be used as follows.

Compliance

A compliance requirement defined for a bank might require that a customer verify his residential address at least once per year by providing two pieces of identification and a piece of mail that shows his address. The bank's customer service representative would store the details of this information in the system and create an Event Manager action that will send a notification a year from the date, requesting that the compliance be re-validated.

Questionnaire

A banking customer might fill out a identity verification questionnaire that will be used when he logs onto his bank account online. The customer would choose a number of questions that he will be asked when he logs onto his account. He also provides the correct answers that will be required to log on successfully. These answers are stored and are used to verify the answers the customer provides.

Line of business

The line of business (LoB) relationship business feature is used to associate parties to specific lines of business and manage individual party-to-LoB relationships as well as adding, updating and getting these relationships as part of party-level composite transactions.

Line of Business example use

A business might use this feature to associate a party with the different lines of business that the party uses, such as mortgages, credit cards, insurance and others. For example, the Line of Business feature shows the relationship that the party John Smith has to Home Insurance, and that the party Jane Doe has to Retail banking and Life insurance.

Macro and entity roles

The Party module provides services to capture and manage one or more macro roles that a party plays in the system.

The Party module also provides services that manage party roles in party grouping and party relationships. The common business services module provides services for capturing and managing entity roles within a hierarchy.

As a part of playing the macro role, some of the existing party data can be associated with a macro role to create a view for the given party in a role. For example, a party may have the role of ″prospect″, and an email address is collected with this role, thus linking the party macro role with this contact method.

In general, IBM InfoSphere Master Data Management Server enables you to manage the following roles:

- Contract party roles
- Claim party roles
- Party macro roles
- Entity roles

Contract party roles and claim party roles provide services to capture and manage the various roles that a party plays in the system with regard to contracts and claims. That is, these roles only exist within the context of the contracts or claims that they are associated with.

Party entity roles are similar to contract and claim roles, in that these roles only exist within the context of the entities that they are associated with. In this case, however, the entities are party-centric party relationships, party groups, and party hierarchies.

Party macro roles, in contrast, provide the context in which to view the associations. That is, in a given party role, a given set of associations are valid.

This section concentrates on the specifics of these new party-centric macro and entity roles.

Macro and Entity Roles example use

Behavior When Configured On
Macro and Entity Roles do not need to be configured.

Behavior When Configured Off
Macro and Entity Roles do not need to be configured.

Behavior When Configuration is Changed in a Production Environment
Macro and Entity Roles do not need to be configured.

Configured During Installation
Macro and Entity Roles do not need to be installed.

Modifying This Feature
Roles can be modified according to the procedures in the *IBM InfoSphere Master Data Management Server Developers Guide*.

Normalization

The Normalization feature provides the option of allowing for normalization of the address and contact method data.

Normalization involves taking a set of data that is provided in a single field, and parsing it out into its individual elements, so that they can be stored separately.

Normalization example use

If `555 Acme Rd.` is received in line 1 of the address, then, with either IBM Information Server QualityStage or Trillium Standardizer configured, MDM Server can store `555` as the street number, `Acme` as the street name and `ROAD` as the street type. The unnormalized data is the address line 1, and the normalized data is the separately stored street number, street name and street type.

For more information on IBM Information Server QualityStage, see "IBM Information Server QualityStage integration" on page 65.

For more information on the Trillium Software System, see http://www.trilliumsoft.com.

Party demographics

The party demographics business feature maintains custom party demographic data.

The type of party demographic data required varies by clients depends upon their implementation and could include, for example, employment history, military experience, education information, and personal data, such as height and weight. The party demographics feature uses Specs to allow you to easily define custom demographic attributes that are suited to your specific needs, and then store the variable data as XML columns.

Related concepts

"Specifications" on page 74
The Specifications feature allows you to define the metadata needed to support dynamic (or soft) attributes in InfoSphere MDM Server.

Party demographics example use

The party demographics feature can be used to store personal demographics information or organizational demographics information. The following are examples of each type:

Personal demographics example

A medical organization might use this feature to store and monitor patients' weight and height information over a given time period in an effort to evaluate a weight loss program for diabetes prevention.

Organizational demographics example

A Reinsurance company might choose to store the following types of demographic information about each company it insures:

- Organization Legal history details, such as if the company has convictions, past litigations, or current litigations, as well as the relevant judgment details
- Financial Status, such as if the company has ever declared bankruptcy, if it failed to meet NASDAQ continued listing requirements, or if it missed regulatory filing deadline
- Fiscal Year Start Date or a similar Business Year End indicator
- Gross Revenue
- Number of Employees, as a number, or selected from a range of figures
- Legal Structure, such as whether it is a corporation or not
- Tax State Code

These details can help further define the 360 degree view of each organization in the system. A company can choose to proactively record these dates in acknowledgment of internal weaknesses in customer information data or they can record them as a result of regulatory mandates.

Party equivalencies

A party that is managed within the Party domain can also exist in another system.

A party equivalence key provides you with the ability to identify how the party is known in other systems; in other words, the party's identifier in the other system.

Party equivalencies example use

A Party within InfoSphere MDM Server might be known in an external, back office system by the administrative key of P896493. This equivalent administrative key, P896493, is added to the Party record within InfoSphere MDM Server. Multiple equivalency keys can be associated with a single Party record within InfoSphere MDM Server.

Party life events

The Party life events business feature allows you to manage life event information related to a party, by providing services to explicitly add, update or get party life events, as well as automatically detecting events when party data is modified or when events occur with the passage of time.

Party life events uses the Event Manager for all events related logic. For the explicit event services, it uses a IBM InfoSphere Master Data Management Server controller as a façade to provide the same service-level interface as the other services, and uses the Event Manager as persistent layer.

Related concepts

"Event Manager and Evergreening data" on page 32

Party life events example use

Part life events can be used to add party information, such as the fact that the party won the lottery. It can also be used scheduled manner, for example, detecting that a party has turned 65 years old.

Party location

Party location refers to the addresses and contact methods that can be associated with parties.

Contact methods include client-defined types such as home telephone numbers, fax numbers, cellular telephone numbers and e-mail addresses.

Addresses can be shared across many parties. If you are adding or updating a party's address an existing address can be associated with it if the address refers to the same physical location; otherwise, a new address is created. A party can use an address of any client-defined type, such as home address or work address.

Addresses can be standardized and validated by third party standardizers.

Party location example use

This feature can be used to add address information (such as home or business address) and contact methods (such as telephone number or email address) when adding a new party to InfoSphere MDM Server. When you add the party's spouse, the spouse's record can reference the same address information associated with the original party, as long as they reside in the same residence. Similarly, if you are adding a business contact to the original party, the business contact's record can reference the same business address information associated with the party, as long as they work in the same location.

Party privacy

The Party privacy business feature incorporates privacy legislation with a party's specific privacy preferences to ensure that the party is only contacted when permission has been given, and the party's information is only shared in an agreed-upon manner.

Specifically, Party privacy ensures that institutions comply with the different privacy regulations from all levels of government as well as with the individual's wishes for privacy regarding their personal information. At a high level, the focus of privacy legislation is to record a party's choices regarding permission to contact them about new products and services, and permission to share the party's information with affiliated businesses and other third parties.

Party privacy helps administrators maintain default privacy preferences for their organization and provides services around managing parties' privacy preferences. Because privacy legislation can vary from local to regional to federal, and in some regions is still being crafted, Party Privacy is also able to be updated for changing government requirements.

Party privacy example use

The Party privacy feature can be used as follows.

Client privacy requirements
A client requests that they not be called at home and that their contact information not be shared with third parties. You can use Party Privacy to add that information to the client's party information.

Legislative privacy requirements
New legislation requires that clients not be contacted by phone unless they give permission to do so. You can use Party Privacy to set ″do not call″ as the default, and ″permission granted″ if the client allows phone contact.

Party roles

A given party can play different types of roles in different contexts. Supported roles include roles on contracts, roles in claims and macro roles.

Roles serve as the mechanism that relates parties to other structures, thereby providing a complete view of the party.

The Party module provides services to capture and manage one or more macro roles that a party plays in the system. The Party module also provides services that manage party roles in party grouping and party relationships. The common business services module provides services for capturing and managing entity roles within a hierarchy.

As a part of playing the macro role, some of the existing party data can be associated with a macro role to create a view for the given party in a role. For example, a party may have the role of ″prospect″, and an e-mail address is collected with this role, thus linking the party macro role with this contact method.

In general, InfoSphere MDM Server enables you to manage the following roles:

- Contract party roles
- Claim party roles

- Party macro roles
- Entity roles

Contract party roles and claim party roles provide services to capture and manage the various roles that a party plays in the system with regard to contracts and claims. That is, these roles only exist within the context of the contracts or claims that they are associated with.

Party entity roles are similar to contract and claim roles, in that these roles only exist within the context of the entities that they are associated with. In this case, however, the entities are party-centric party relationships, party groups, and party hierarchies.

Party macro roles, in contrast, provide the context in which to view the associations. That is, in a given party role, a given set of associations are valid.

Party roles example use

Example party entity role

If there is a party named Marcel LeClair who is part of the LeClair Family Household group, you could add a party entity role called Male Head of Household for him.

Example party macro role

You can add the role of Client to a specific party, which would allow you to then link to additional party information, such as address or contact information

Party values

The Party values business feature allows you to classify and store different values for a party generated from external systems like data warehouses, as well as values defined in an implementation, such as demographic information.

Party values are part of the party object, as well as part of party-level composite transactions. Ten value attributes can be persisted for each value type, giving extra storage slots for additional information related to the party. Party values can be returned by party ID, and value category, category code and value are returned as part of the party value response. This data helps determine the party's value, and provides a more comprehensive view of the party. The data collected can be anything from an organization's demographic information to a party's risk score.

Party values example use

A business might use the Party values feature to store fact that a specific party is a high-value client, and to store the score associated with that.

InfoSphere MDM Server integration with third party products

IBM InfoSphere Master Data Management Server integrates with several third party products in order to enhance the functionality provided by InfoSphere MDM Server.

For more information about integrating third party products into InfoSphere MDM Server, see the *IBM InfoSphere Master Data Management Server Developers Guide*.

The following sections provide understanding and planning information to help you integrate third party products into InfoSphere MDM Server.

IBM Information Server QualityStage integration

IBM InfoSphere Master Data Management Server can be configured to use the standardization and matching capabilities of QualityStage.

IBM Information Server QualityStage (QS) is a comprehensive development environment for building applications to re-engineer data. It provides a set of integrated modules for accomplishing data re-engineering tasks such as Conditioning (Standardization), Matching, Searching, and others.

For more information about integrating IBM InfoSphere Master Data Management Server with QualityStage, see the *IBM InfoSphere Master Data Management Server Developers Guide*.

IBM Information Server QualityStage integration example use

If several suspect parties exist, IBM InfoSphere Master Data Management Server will use QualityStage matching to help determine the best match.

IBM Information Server QualityStage Integration configuration behavior

Behavior When Configured On
InfoSphere MDM Server uses the standardization and matching capabilities of QualityStage.

Behavior When Configured Off
InfoSphere MDM Server does not use the standardization and matching capabilities of QualityStage.

Behavior When Configuration is Changed in a Production Environment
This feature can be configured on or off at any time without impact other than changing the availability of the QualityStage capabilities.

Configured During Installation
This feature can be configured at any time.

Modifying This Feature
The IBM Information Server QualityStage Integration feature can be modified according to the procedures in the *IBM InfoSphere Master Data Management Server Developers Guide*.

Dun and Bradstreet integration

Dun and Bradstreet's D-U-N-S® Number can be used as a party identifier in InfoSphere MDM Server.

Dun and Bradstreet provide commercial insight and provides a database businesses uniquely identified with a nine digit number, called the D-U-N-S® Number. The D-U-N-S Number is widely used for keeping track of the world's businesses and many major corporations and governments require their suppliers and contractors to have a D-U-N-S® Number.

InfoSphere MDM Server can be integrated with Dun and Bradstreet in order to store the D-U-N-S Number as party identifier for organizations and provide an additional means of party matching. Additionally, InfoSphere MDM Server provides sample code to demonstrate how InfoSphere MDM Server data can be

enriched with business intelligence from the Dun and Bradstreet global database, adding information such as demographic data and legal hierarchies.

InfoSphere MDM Server clients must have a license agreement with Dun and Bradstreet in order to take advantage of the Dun and Bradstreet integration offered by InfoSphere MDM Server . If InfoSphere MDM Server clients do not renew their license agreement with Dun and Bradstreet, it is the client's responsibility to remove Dun and Bradstreet-licensed information from the InfoSphere MDM Server database if necessary.

Dun and Bradstreet Integration example use

ABC Insurance uses the Dun and Bradstreet integration with InfoSphere MDM Server after the original data load to use Data Stage to produce a matching batch request and send it to Dun and Bradstreet. The request contains information for all organizations that do not have D-U-N-S Numbers as identification.

Dun and Bradstreet Integration configuration behavior

Behavior When Configured On
InfoSphere MDM Server can use Dun and Bradstreet D-U-N-S Numbers as a party identifier.

Behavior When Configured Off
InfoSphere MDM Server cannot use Dun and Bradstreet D-U-N-S Numbers as a party identifier.

Behavior When Configuration is Changed in a Production Environment
Configuration can be changed at any time without impacting anything other that whether or not InfoSphere MDM Server can use this feature.

Configured During Installation
Configuration can be changed at any time.

Modifying This Feature
The Dun and Bradstreet Integration feature can be modified according to the procedures in the *IBM InfoSphere Master Data Management Server Developers Guide*.

Entity Analytics Solutions Integration

InfoSphere MDM Server provides the ability to integrate to the IBM Entity Analytic Solutions (EAS) products.

EAS products are a set of cross-platform, cross-database products that answer ″Who is who?″ (DB2® Identify Resolution), ″Who knows who?″ (DB2 Relationship Resolution) and ″Who knows who anonymously?″ (DB2 Anonymous Resolution) from multiple data sources in near real-time.

InfoSphere MDM Server integrates with both DB2 Relationship Resolution and DB2 Anonymous Resolution as a source system, with a one-way feed from InfoSphere MDM Server to EAS.

If InfoSphere MDM Server is integrated with EAS, a feed from EAS is produced when a party is:

- Added
- Updated with new details
- Collapsed into another party

- Split from another party
- Inactivated
- Deleted

InfoSphere MDM Server is responsible for maintaining its database of customers, and EAS is responsible for maintaining its database of entities. Because the integration is a one-way feed from InfoSphere MDM Server to EAS, InfoSphere MDM Server does not store or maintain the EAS entity ID in the InfoSphere MDM Server database.

Entity Analytics Solutions Integration configuration behavior

Behavior When Configured On
InfoSphere MDM Server is integrated with EAS products, and uses its entity identity and relationship resolution facilities.

Behavior When Configured Off
InfoSphere MDM Server is not integrated with EAS products.

Behavior When Configuration is Changed in a Production Environment
This feature can be configured at any time without impact other than enabling the EAS integration.

Configured During Installation
This feature can be configured at any time.

Modifying This Feature
The Entity Analytics Solutions Integration feature can be modified according to the procedures in the *IBM InfoSphere Master Data Management Server Developers Guide*.

Chapter 4. The Product domain

The Product domain is an operational-styled hub that manages the definition of products.

Its collection of products makes up a product catalog that is accessible to other systems across the enterprise.

Related concepts

"InfoSphere MDM Server domains" on page 1
IBM InfoSphere Master Data Management Server enables companies to extract maximum value from master data by centralizing multiple data domains and providing a comprehensive set of prebuilt business services that support a full range of master data management (MDM) functionality.

Product type hierarchy

The core product services business feature enables enterprises to maintain operational product data, including basic product information, product identifiers, product administration system keys, and product relationships. It also provides services to manage product type hierarchies, which are supported using both hardened product subtypes (that is, Goods, Services, Financial, and Insurance), and soft product types which are managed using Specs.

Specs are a mechanism to support attributes that are not hardcoded into the object model.

A product can also be categorized. Categorization is the process of classifying a product into a category. This process entails creating associations between products and categories. One or more products may be categorized into multiple categories in multiple hierarchies.

Related concepts

"Specifications" on page 74
The Specifications feature allows you to define the metadata needed to support dynamic (or soft) attributes in InfoSphere MDM Server.

"Product categories and product hierarchies"
The category hierarchy feature enables you to classify products into one or more hierarchical structures by providing the services to create and maintain product categories in a hierarchy, and to map products to these categories.

Product type hierarchy example use

A financial institution might create a new Everyday Savings Account banking product that is of the financial product type. A bundle type of relationship might then be created between an existing Premier Banking Package and the new Everyday Savings Account banking products.

Product categories and product hierarchies

The category hierarchy feature enables you to classify products into one or more hierarchical structures by providing the services to create and maintain product categories in a hierarchy, and to map products to these categories.

The feature also provides services to: search for category hierarchies; search for categories within a category hierarchy; and to manage category administration system keys.

In a category hierarchy, each node is a category, and the hierarchical structure is formed using category relationships that define the parent-child relationship between two categories. Products can be associated with one or more categories in the same category hierarchy, as well as categories in other category hierarchies.

Related concepts

"Product type hierarchy" on page 69
The core product services business feature enables enterprises to maintain operational product data, including basic product information, product identifiers, product administration system keys, and product relationships. It also provides services to manage product type hierarchies, which are supported using both hardened product subtypes (that is, Goods, Services, Financial, and Insurance), and soft product types which are managed using Specs.

Product categories and product hierarchies example use

A retailer might set up two category hierarchies to classify its products. In one category hierarchy, the categories are named and organized with an internal merchandising view, while in the other category hierarchy, the hierarchical structure reflects the view of the end-consumer. The retailer can use either category hierarchy to drill down to the products by their categorizations within the respective category hierarchy. However, on the retailer's online store, the end-consumers can view products using the category hierarchy with the customer view only.

Generally speaking, the retail can create as many category hierarchies as required to categorize the same set of products or different sets of products.

Product category attributes

The Product category attributes feature provides the ability to capture additional data for products with dynamic—or soft—attributes, based on how the product is categorized. Category hierarchies are defined using the Specifications feature in InfoSphere MDM Server. For more information, see "Specifications" on page 74.

Product category attributes example use

An electronics retailer carries a wide range of products, from high-definition television sets to microwaves. They require different sets of product attributes for different kinds of products. For example, product dimensions and product warranty might be common attributes across all electronic products, but high-definition television sets require additional attributes such as `AspectRatio` and `DisplayResolution`, while microwaves have attributes such as `OvenCapacity` and `TurntableType`.

To capture all the attributes required by different products, the Product category attributes feature can be used to define the product attributes based on how each product is categorized into category hierarchies.

First, a category hierarchy is created with the relevant product categories. Using the example above, the electronics retailer could define a category hierarchy that

includes a category called `High Definition Televisions` and another category called `Microwaves`. Then, the various product specs are defined as required, added to InfoSphere MDM Server, and associated with categories in the category hierarchy.

For example, the specification for high-definition television sets that is associated with the `High Definition Televisions` category might look like this:

```
HDTVSpec

AspectRatio  String Enum ("16:9", "4:3")
DisplayResolution String
InputCapabilities String
Picture-In-Picture String Enum ("Yes", "No")
```

The specification for microwaves that is associated with the `Microwaves` category might look like this:

```
MicrowaveSpec

OvenCapacity (Cu. Ft.) Long
TurntableType  String
KitchenTimer  String Enum ("Yes", "No")
```

When you associate the specification with a category, you can also indicate whether the specification is to cascade down to all subcategories, so that even products categorized into an n-level deep subcategory can also access the specification.

Next, products are associated with the categories, as appropriate. Finally, based on the product category associations, the attributes defined in the specifications can be added, updated or retrieved for all of the products associated with the category. For instance, any product associated with the `High Definition Televisions` category has access to the attributes in HDTVSpec, while any product associated with the `Microwaves` category has access to the attributes in MicrowaveSpec.

For more information on how to associate specifications with categories, refer to the *IBM InfoSphere Master Data Management Server Developers Guide*.

Product relationship

The Product relationship business feature holds product information, and structures the product information through relationships.

It stores product information such as:

- Product Family
- Product Category
- Product Type
- Product Name
- Product Descriptions

The Product relationship feature captures the product information from a system, from users, or both, and presents the product details to other systems, and users. It allows systems and users to perform updates, and to date and delete the information. The Product relationship feature records the source of the information—whether it is from a system or user, and record the date of the source of the information, and it keeps a history of the source and source date.

Product Relationship example use

Product family, category, and type are ways you can structure your product information. You can decide what to name the groups and how to structure the relationships of the various groups. You can determine what information you need to hold, depending on your business model.

Product equivalencies

A product that is defined within the Product domain might also exist in another system. A product equivalence key allows you to determine the product's identifier in the other system.

Since the product's identifier can be made up of a number of different parts, the product equivalence key allows you to store the identifier in its parts or as a concatenated string.

Product equivalencies example use

Within a financials application system, a checking product is identified by the key CH100889. Within InfoSphere MDM Server, the checking product is identified by the native key 1283794. You can associate the financials application equivalence key CH100889 with the InfoSphere MDM Server native key 1283794.

Product identifiers

This feature provides the ability to store known identifiers of the product that might be assigned by third parties.

Examples include NSIN, CUSIP, ISIN in financial services and GTIN, barcodes, UPC in retail.

Product identifiers example use

A retail store might use the Product identifiers feature to store the Global Trade Item Number) (GTIN) for each item it sells. The GTIN is used to uniquely identify items in the retail sector.

Product search

Using predefined search methods, the Product search business feature enables a user to search for one or more products by issuing a single search request using one or more primary search criteria and zero or more secondary search criteria.

The following are the available primary search criteria:

- Product Name
- Alternate Identifier
- Administrative System Key

The following are the available secondary search criteria:

- Product Relationship Type
- Product Type
- Product Short Description

- Status Type

This feature supports wildcard and look-a-like searches. Request parameters include optional inquiry levels. When you search by product relationship type, you can provide an optional filter value to refine the search results.

This feature also supports searches for localized content, and also supports the pagination feature.

Product search example use

A business might use the feature to search for a product with a name that contains the word `VISA` and that has a Status Type of 1.

Product search enhancements

The search product enhancements provide the following additional capabilities:

- To search for products within category hierarchies as well as across category hierarchies.
- To search for products based on structure type.

The additional search criteria include:

- Product structure type, and
- Entity Category Search BObj, which consists of:
  - Category hierarchy type
  - Category hierarchy name
  - Category hierarchy Id
  - Category hierarchy start date
  - Category hierarchy end date
  - Category name
  - Category code
  - Category Id
  - Category start date
  - Category end date
  - Include sub categories indicator

Product search enhancements example use

An item specialist with a consumer electronics retailer would like to find all value package products with “Bundle” in the name, that are categorized under any hierarchies of type Consumer, and that the hierarchies are affective between 2009 and 2020. The item specialist specifies search criteria such as here to find the products.

- Category hierarchy type = `consumer`
- Category hierarchy starting date <= `January 1, 2009`
- Category hierarchy ending date >= `December 31, 2020`
- Product name like `%bundle%`
- Product structure type = `Bundled`

Product terms and conditions

The terms and conditions business feature represents a common component that is shared by both the Product and Account domains, and provides services to manage terms and conditions.

Terms and conditions can be of a static nature containing freeform text or reference to an external document, or they can contain structured data used as parameters to evaluate conditions implemented in external programs or rules engines.

Terms and conditions must be associated with one or more products, product relationships, or agreements. They are hierarchical and therefore can contain sub-conditions. A Managed Account can inherit a product's terms and conditions, and can also override the term and conditions, if it is allowed by the product.

Product terms and conditions example use

A financial institution might create a new term and condition stating that a `Value Package` product must contain at least one `anchor` product.

Specifications

The Specifications feature allows you to define the metadata needed to support dynamic (or soft) attributes in InfoSphere MDM Server.

Dynamic attributes are ones that are not hardcoded into the object/data model. They are used in the Party Demographics feature, as well as in the product domain, to provide easy support for highly dynamic and flexible custom data models.

Specs define the hierarchical structure of dynamic attributes and their types, and they provide other related metadata through the use of XSD technology. The variable data is stored as XML columns in the database. Specifications and related metadata—Specs, Spec Formats and Spec Format Translations—are defined through the InfoSphere MDM Server Workbench and then deployed to an InfoSphere MDM Server instance. In InfoSphere MDM Server, Specs are managed through a set of Spec-related administrative services.

Related concepts

"Party demographics" on page 61
The party demographics business feature maintains custom party demographic data.

"Product type hierarchy" on page 69
The core product services business feature enables enterprises to maintain operational product data, including basic product information, product identifiers, product administration system keys, and product relationships. It also provides services to manage product type hierarchies, which are supported using both hardened product subtypes (that is, Goods, Services, Financial, and Insurance), and soft product types which are managed using Specs.

Specifications example use

Specifications are used in the product domain so that the limited data models provided for the built-in products types can be easily extended. For example, a company selling financial products might have a Customer Loyalty Program for

each of their products, for which they need to capture and store data. Although they could add extra attributes to the existing data model using the Extension mechanism, a simpler and more flexible way to support these attribute is to define a Loyalty Program Specification for them. Conceptually, the Specification for this might look like this:

```
LoyaltyProgramSpec

AmountDerivedType  Decimal  Pattern="#,###.##"

RewardType         String  Enum("Travel", "Auto",...)
BonusAmount        AmountDerivedType
PointsBalanceRatio AmountDerivedType
RewardPartnerInfo  Complex
     PartnerName   String  MaxLength(25)
     AgreementNum  Long
```

The company could then add this specification to InfoSphere MDM Server and associate it with the FinancialProduct type. Once this is done, these custom attributes will apply to all products that are instances of a FinancialProduct.

The company could also create custom product types and use specifications to define a set of custom attributes for these types. For more information, refer to the topic *InfoSphere MDM Server Metadata specs* in the *IBM InfoSphere Master Data Management Server Developers Guide*

Chapter 5. The Account domain

The Account domain is an operational-styled hub that manages account data.

An account can be either a managed account or a referenced account.

- A *managed account* is an account that is managed fully by the Account domain and is a system of record.
- A *referenced account* is an account that is managed in a different system, which can be either internal or external to the organization.

An account is made up of an agreement and set of accounting units.

- An *agreement* manages legally binding terms for a relationship between an institution and any party to which that institution has a legal relationship. It contains the legal terms that are required for financial transactions.
- An *accounting unit* tracks all debits and credits for a particular type of balance that must be managed for an agreement. The accounting unit entity is not provided in Account domain.

Related concepts

"InfoSphere MDM Server domains" on page 1
IBM InfoSphere Master Data Management Server enables companies to extract maximum value from master data by centralizing multiple data domains and providing a comprehensive set of prebuilt business services that support a full range of master data management (MDM) functionality.

Agreement Business Services

The Agreement Business Services business feature provides a framework through which customers—on their own—can determine if they are eligible to receive a particular product or service.

Agreement Business Services also allows them to run "what-if" scenarios to simulate the effects of modifying their existing agreement without actually modifying the agreement.

Agreement Business Services is able to distinguish between existing customers and unregistered users. When existing customers access the eligibility determination feature, Agreement Business Services automatically looks up information from the Party and Account domains to determine if the customers are eligible to receive the requested product or service. When unregistered users access the eligibility determination feature, they are required to provide additional input to help Agreement Business Services determine if they are eligible to receive the product or service.

Agreement Business Services enables customers of an organization to run what-if scenarios on existing agreements to help them understand if modifying an agreement in a particular way would be advantageous or not. The actual agreements are not modified when customers run what-if scenarios. Unregistered users can also use this service to simulate the effects of modifying an agreement—even though an agreement does not yet exist—so that they can make an informed purchase decision.

The Agreement Business Services thereby provides both existing customers and unregistered users a self-service mechanism through which they can assess their eligibility for a product or a service and determine the effects of account modifications, all on their own.

The terms and conditions business feature is used to model the product or service eligibility criteria, as well as to determine the effects of account modification. Agreement Business Services uses the relevant terms and conditions data to evaluate and present the output to the requestor. For more information, see "Agreement terms and conditions."

Agreement Business Services example use

A financial institution might use the Agreement Business Services feature on its website to provide existing customers and unregistered users with the ability to determine their eligibility for a new financial advisory product bundle all on their own.

A financial institution can also use the Agreement Business Services feature to allow its users to simulate modifications to their existing agreements so that they can understand the impact of any changes. For instance, a customer that has a product bundle containing a checking account and a savings account can run a what-if scenario to determine what benefits would be lost if they closed their checking account. Thee existing agreement is not modified in this process. .

Agreement terms and conditions

An agreement inherits the terms and conditions provided by the base product. However, the agreement terms and conditions feature you to can augment or override the terms and conditions for a product, if necessary.

An agreement can have multiple product relationships for the purpose of describing products as part of the supply agreement. Consequently, products that are connected to an agreement via product relationships do not have any impact on the terms and conditions of an agreement.

Agreement Terms and Conditions example use

A party enters into an agreement with a bank to open a new savings account. The savings account product has a term and condition that states it will bear an interest rate of 2%. However, because the party is a long standing and valued customer of the bank, the interest rate is upgraded to 3%. Within InfoSphere MDM Server, the agreement will include a term and condition that overrides the product term and condition.

Terms and conditions rules setup

The terms and conditions feature gives users the ability to define the terms and conditions associated with agreements and products, and to capture the text associated with the conditions, as well as the condition attributes.

The terms and conditions rules setup framework is shared by both the Product and Account domains. It provides a general method to setting up the terms and conditions rules that can be used for either domain.

A rule is a set of conditions and a set of associated actions. The set of condition determines the validity of something—for example, if the customer is qualified to purchase a product. The set of actions specifies something for the system to perform—for example, reduce the mortgage rate by .5%. A rule must have one or more conditions, and zero or more actions.

Terms and conditions rules setup example use

The eligibility rule for a checking and investment combination product might look something like the following:

- Select a checking account from the following list: Regular, Express, or Interest.
  - For each account, the following fee condition applies:
    - $9.95 per month for ten transactions.
    - Each additional transaction costs $.50.
  - The monthly fee condition is waived when the following conditions are met:
    - The minimum balance is $1000.
    - The account status is Active.
- Select an investment account from the following list: Discount, or Full.
  - For each account, the following fee condition applies:
    - $6.95 per month.
    - $5.00 per transaction.
  - The following promotion conditions also apply:
    - Monthly fee is reduced to $3.95, if the account is opened online.
    - A $50 credit is provided if the account is opened online.
    - Transaction charges are only $2.50 per transaction if they are done online.

The conditions are then used to determine the amount that the customer is charged per month for each account.

Agreement dynamic attributes

In the Account domain, an agreement manages legally binding terms for a relationship between an institution and any party to which that institution has a legal relationship. The agreement dynamic attributes feature provides the ability to extend the data model for agreements with dynamic—or soft—attributes using the Specifications feature in InfoSphere MDM Server.

Agreement dynamic attributes example use

Specifications are used in the account domain so that the limited data model provided for agreements, which use the same data model and transactions as contracts and accounts, can be easily extended based on agreement type.

Using value packages as an example, a bank might sell a savings account product and a checking account product as a value package. When a customer purchases the bundle, the bundle itself is stored as a managed account with an agreement type of `Value Package`; the savings account and the checking account are stored as two separate referenced accounts. The bank might have other agreement types, such as `Vendor Agreement` or `Service Agreement`, for which it needs to capture and store data. One method of storing this data, could be to add additional attributes to the existing physical data model for the different agreement types. Alternatively,

a more flexible way to support additional attributes for different agreement types is to define unique agreement specifications for each agreement type. In this value package example, the specification for value packages might look like this:

```
ValuePackageSpec

NumberOfAccounts Long
ConsolidatedMonthlyFee AmountDerivedType
EligibilityIndicator String Enum ("Y", "N")
ValuePackageSubType String Enum ("Student Value Package", "VIP Package", ...)
WelcomeGiftClaimed Boolean
```

Once this specification is added to InfoSphere MDM Server and associated with the `Value Package` agreement type, these custom attributes will apply to all agreements created with the `Value Package` agreement type.

For more information on how to associate specifications with agreement types, refer to the *MDM Server Metadata specs* topic in the *IBM InfoSphere Master Data Management Server Developers Guide*.

Agreement dynamic attributes configuration behavior

Behavior When Configured On

Once the specifications and agreement types are set up properly, agreement dynamic attribute values, which are provided using the ContractSpecValueBObj child business object, can be added and updated at any time without special configuration.

To enable the agreement dynamic attribute values—using ContractSpecValueBObj—to be returned with the parent agreement, you must first make sure that the Smart Inquiries setting for this child object is set to `INACTIVE`. In other words, configure agreement dynamic attributes on, you need to set `INACTIVE_IND = 'Y'` in the EXTENSIONSET table for ContractSpecValueBObj. Once this is set, if an agreement has agreement dynamic attribute values, the values can be retrieved using the appropriate inquiry transactions and inquiry level.

Behavior When Configured Off

Agreement dynamic attribute values can be added or updated, but they cannot be retrieved and are not returned by any of the inquiry transactions. By default the Smart Inquiries setting for the ContractSpecValueBObj child object is `ACTIVE`. This means that by default agreement dynamic attribute values are configured off, so the system does not attempt to access unused parts of the system.

Behavior When Configuration is Changed in a Production Environment

The Smart Inquiries setting for agreement dynamic attribute values, ContractSpecValueBObj, can be changed at any time without impact other than determining whether the agreement dynamic attribute values are returned or not in inquiry transactions.

Configured During Installation

Smart Inquiries setting for agreement dynamic attribute values, ContractSpecValueBObj, can be enabled or disabled at any time.

Modifying This Feature

For more information on changing the configuration for agreement dynamic attributes, refer to *Manipulating dynamic attribute data* topic in the *IBM InfoSphere Master Data Management Server Developers Guide*.

Billing

The billing business feature provides services to manage billing summaries and their associated miscellaneous values in the system, including the Create, Read, Update and Delete services for billing data.

This feature does not perform any billing-related calculations, invoicing, or billing status tracking—it is not intended to replace any billing system.

Two terms that are used when discussing Billing are:

- **Billing Summary**—a summary bill for a contract or a contract component. There can be many billing summaries for a given contract or contract component.
- **Billing Misc Value**—any additional billing value not captured directly within the billing summary. Each value is associated with one a billing summary.

Billing example use

The billing summary provides an institution with a consolidated enterprise view of all of the bills for an individual party or contract, as well as the related accounts, or contracts, across all line of businesses. Also, the Billing Summary provides the ability to modify billing information in context for a contract or contract component. In general terms, billing is flexible enough to allow bills to be related to a contract or contract component as well as to a party, using the contract role.

Claims

The claims business feature provides institutions with a consolidated, enterprise-wide view of all the claims for a party and the policy or contract to which the claim applies. This helps identify potential problems such as fraudulent claims.

Claims to items or holdings are required to track the moneys issued against a particular personal belonging. As part of the claims design, holdings or items that may be called assets and liabilities are persisted and linked to a claim through the Agreement (Contract) component.

A claim is a request for insurance agreement benefits from one or more interested parties. A claim can have one or more other claim items associated with it; for example, a single-loss event such as a flood catastrophe triggers different claims on different insurance agreements—household, car, life policy and so on—and every claim is eventually composed of one or more claim coverage parts, such as the water damage in the household policy. Depending on the particular line of business, claim files can have a very short to a very long life cycle. The life cycle of a claim can exceed the end date of the insurance agreement.

Claims is dependent on the Financial and Party project packages—that is, claims is related to one or more policies and can involve one or more parties. Interfaces, classes and other elements of claims are packaged into a project called claims residing within the Financial project. Holdings is dependent on the Party project packages—a holding can exist without a link to a party in this design, but it is assumed that roles are associated to the holding.

Related concepts

"Holdings"
The holdings business feature provides institutions with a consolidated, enterprise-wide view of all the holdings for a party, and the policy or contract to which the claim applies.

Claims example use

The claim summary provides an institution with a consolidated enterprise view of all of the claims for an individual party and the related property insured in a contract across all lines of business.

Contract Values

The Contract Values business feature enables you to classify and store different values for contracts generated from external systems such as data warehouses, as well as values defined in an implementation.

Contract values are part of the contract object. Ten value attributes can be persisted for each value type, providing extra storage slots for additional information related to the contract.

Contract values can be retrieved by providing the contract ID and value category. As part of a contract value response, the category code and value are returned. This data helps determine the contract's value, and provides a more comprehensive view of the contract.

Contract Values example use

Contract values can be used to store any extra information about a contract, such as special contract pricing, or the fact that pricing is subject to review and is dependent upon performance or compliance with certain restrictions.

Holdings

The holdings business feature provides institutions with a consolidated, enterprise-wide view of all the holdings for a party, and the policy or contract to which the claim applies.

This helps to identify potential problems such as fraudulent claims. Claims to items or holdings are required to track the moneys issued against a particular personal belonging.

As part of the claims business feature, holdings or items that may be called assets and liabilities are persisted and linked to a claim through the Agreement (Contract) component.

Terms that are used when discussing holdings are:

- **DTO**—Data Transfer Objects
- **Party Items** or **Party Holdings**—the personal belongings a Party owns

Related concepts

"Claims" on page 81
The claims business feature provides institutions with a consolidated,

enterprise-wide view of all the claims for a party and the policy or contract to which the claim applies. This helps identify potential problems such as fraudulent claims.

Holdings example use

Holdings can be used to identify a party's personal assets and business assets, and identify whether personal claims and business claims are related to the same asset.

Relationships

The relationships feature allows you to create a relationship between different accounts.

Related concepts

Chapter 3, "The Party domain," on page 55
The Party domain manages the entirety of data related to parties such as customers, vendors and suppliers, and maintains a single, consistent version of this data.

Relationships example use

A Value Package managed account might be related to both a 401K Plan reference account and a Mortgage reference account.

Value packages

A value package allows you to bundle two or more products to be sold to your customers.

Value packages example use

A bank might sell a savings account product and a checking account product as a value package. When a customer purchases the bundle, the bundle itself is stored as a managed account; the savings account and the checking account are stored as two separate referenced accounts.

Appendix A. Notices

This information was developed for products and services offered in the Canada.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country/region or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country/region where such provisions are inconsistent with local law:
INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions; therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

This document may provide links or references to non-IBM Web sites and resources. IBM makes no representations, warranties, or other commitments whatsoever about any non-IBM Web sites or third-party resources that may be referenced, accessible from, or linked from this document. A link to a non-IBM Web site does not mean that IBM endorses the content or use of such Web site or

its owner. In addition, IBM is not a party to or responsible for any transactions you may enter into with third parties, even if you learn of such parties (or use a link to such parties) from an IBM site. Accordingly, you acknowledge and agree that IBM is not responsible for the availability of such external sites or resources, and is not responsible or liable for any content, services, products, or other materials on or available from those sites or resources. Any software provided by third parties is subject to the terms and conditions of the license that accompanies that software.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information that has been exchanged, should contact:

IBM Canada Limited
Office of the Lab Director
8200 Warden Avenue
Markham, Ontario
L6G 1C7
CANADA

Such information may be available, subject to appropriate terms and conditions, including in some cases payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems, and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements, or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information may contain examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious, and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information may contain sample application programs, in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Each copy or any portion of these sample programs or any derivative work must include a copyright notice as follows:

© (*your company name*) (*year*). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. *_enter the year or years_*. All rights reserved.

Appendix B. Trademarks

Company, product, or service names identified in the documents of the text may be trademarks or service marks of International Business Machines Corporation or other companies. Information on the trademarks of IBM Corporation in the United States, other countries, or both is located at http://www.ibm.com/legal/copytrade.shtml.

Windows is a trademark of Microsoft Corporation in the United States, other countries, or both.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, or service names may be trademarks or service marks of others.

Index

I

K

L

M

N

P

R

S

T

U

V

W

Printed in USA