Introduction to the MX platform

Site map

What is the MX platform?

MX's platform is made up of dozens of distributed services, each contributing part of our solution to the complex problem of aggregating, enhancing, and delivering financial data to both end users and partner organizations. The platform includes many purely internal tools and services, as well as many products made for partners such as RESTful web APIs, embeddable browser-based digital money management tools, cross-platform mobile money management apps, powerful customer analytics and marketing software, and much more.

MX *aggregates* financial data from as many sources as possible. What does that mean? The idea is basically this:

- We get account and transaction data from thousands of financial institutions, both large and small. To do that, end users provide their online login credentials for a given financial account, whether it's a bank account, retirement account, investment account, etc.
- We then use various methods to bring data from those accounts into our system; sometimes we have direct data-sharing connections, sometimes we use third-party data providers with similar capabilities. But at the end of the day, we want an end user to be able to see every piece of their own financial data using the MX platform with total ease.
- Partner organizations can also send their own customers' data to MX en masse and/or on the fly using one or more of our APIs, such as MDX, Nexus, or Batch (beta).

MX then *enhances* that data with the help of systems built by some crazy-smart folks.

- Every piece of data is analyzed;
- We assign account types and subtypes like `PLAN_401_K`, `CHECKING`, or `SAVINGS`;
- We put every transaction into one of more than 100 default categories;
- Whenever possible, we assign additional classifiers like `is_bill_pay` or `is_direct_deposit`;
- We clean up transaction descriptions so actual humans can read them; something like `DBT CRD 1357 07/22/19 XXXX1357 JCWS RESTAU LEHI UT C#5144` becomes simply `JCW's Burgers`;
- We add dates, locations, and even provide merchant logos wherever possible.
- If we have enough data (a couple months' worth of transactions), we'll automatically calculate budgets for users to help them spend within their means;
- We can predict future spending based on past data;
- We can even let partners know when end users have financial accounts with competitors and give extensive analytics on how end users use their money, interact with products, or behave financially.
- And that's just a few examples. There are many, many more. It's awesome.

MX then *delivers* this enhanced financial data to partners and end users through various different APIs, web interfaces, or even daily change files, depending on the specific needs of the partner. We have a solution for the needs of virtually any partner.

All of this is built around a data model that informs and shapes all of our systems.

MX data architecture: the foundation of our platform

There are several core resources in the MX data architecture: users, members, accounts, transactions, institutions, clients, and accessors. There are many more resources, of course, but these make up the foundation on which everything else is built.

- *Accessors* represent the partner who accesses one or more of our products, such as Nexus, MoneyMap, Target, or all three. This is the highest level in the MX platform's hierarchy. Accessors have an associated API key which must be used in all API requests to MX as a layer of authentication. IP whitelisting is also managed through accessors.

  Accessors can be created only by MX.

- *Clients* also represent partners that interact with the MX platform, but one level down in the platform's hierarchy. An accessor can have one or multiple clients. For instance, a bank or credit union that partners with MX will likely have a single accessor with a single client. But a large online banking provider may partner with MX and integrate our products into an offering they make available to dozens or hundreds of banks; that partner would have one accessor but many clients, one for each specific financial institution.

  Clients have an associated a `client_id` which is used in API requests as an added layer of authentication.

  Clients are also where MX products are customized and branded to meet the needs of a specific partner.

  Clients can be created only by MX.

- *Users* represent the end user whose data MX is enhancing, analyzing, aggregating, etc. These end users may be using the Helios mobile app or MX widgets, or even a custom UI built on top of the Nexus API. Regardless, they are all represented by a `user` object.

  Users are created by partners, generally using the MDX Real Time API. Users belong to a specific `client`.

- *Members* are a little bit abstract, but they are one of the most important resources on the MX platform. Members represent the connection between a `user` and an `institution`. That `institution` may represent the partner, or it may represent another financial insitution from which MX is aggregating data.

  Members can have multiple accounts. For instance, an end user may have both a savings account and a checking account with the `institution` called Wells Fargo. Hence, there will be two `account` objects on the MX platform, both of which belong to a single `member` representing the overall connection.

  Aggregation takes place at the level of members, and every `account` belongs to only one `member`. End-user credentials are stored at the `member` level, and multi-factor authentication is handled with members. Practically speaking, this means members are the objects partners will interact with most often.

  Members can be created both by partners and by end users, depending on the specific MX product.

  MX does not allow duplicate members. More on what this means can be found in the technical documentation for a given product.

- *Accounts* represent an individual financial account of some kind, whether checking, savings, a mortgage, a 401(k), etc. Every account belongs to just one `member`. If a user has multiple accounts with a single financial institution, all of those accounts will belong to the same `member`, as described above.

Accounts can be created using several MX products. They are also created automatically by MX when a `member` is successfully aggregated and when given in response to an MDX On Demand request.

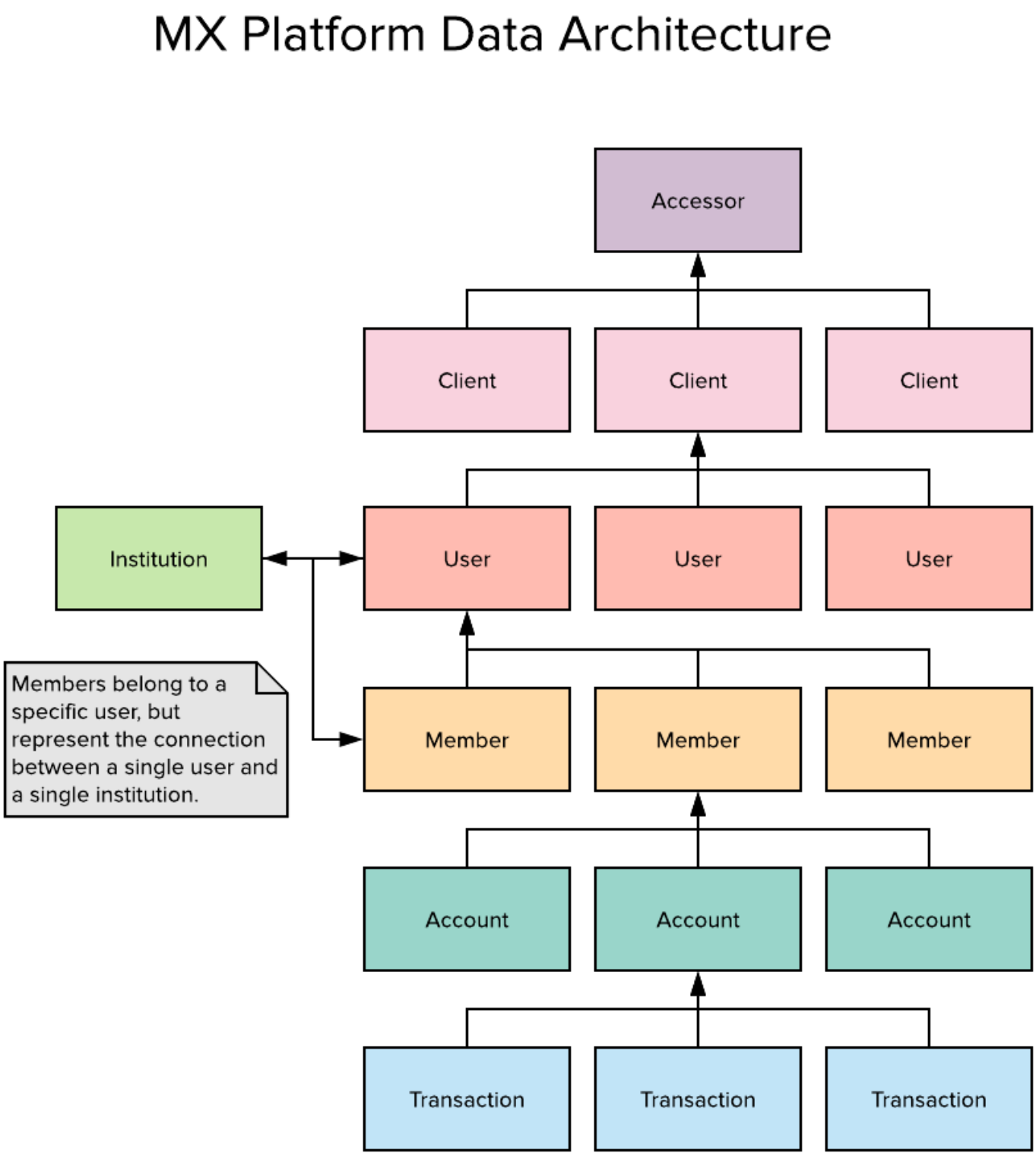
- *Transactions* represent any instance in which money moves into or out of an `account`. This could be a purchase at a business, a payroll deposit, a transfer from one account to another, an ATM withdrawal, etc. Each `transaction` belongs to only one `account`.

  Partners can create transactions using several MX products. They are also created automatically when a `member` is successfully aggregated and when given in the response to an MDX On Demand request.

- *Institutions* represent a financial institution like Chase or Wells Fargo. It's important to point out that what we may think of as a single real-world financial institution may actually have several different `institution` objects on the MX platform. This is because, for example, the mortgage division of Wells Fargo might use a separate system than its everyday banking division, which is different from its credit card division, etc.

  Institutions are important when creating members, as described above. Only MX can create an `institution`. In addition, a `client` may set a default `institution` representing the partner organization itself.

  Institutions can be created only by MX.



Ways of interacting with the MX platform

MX provides a number of products, APIs, websites, or other methods of interacting with the platform, each geared to a specific set of partner needs, and each differing in the degree of complexity and customizability. Many of these can be used in tandem.

Here's a list with brief descriptions.

MoneyMap

MoneyMap is a collection of web-based widgets that allow end users to manage their finances with a beautiful, responsive, browser-based design. Each widget (or all of them together in a "master widget") can be easily embedded in a partner organization's website or through webviews, and accessed through a one-time URL.

End users can add and aggregate their accounts from a wide array of financial institutions, all with a simple and appealing user interface. End users can also manage their budgets, track their spending, predict future cash flow, analyze and categorize, specific transactions, track their net worth, and even track subscriptions (like Netflix or LootCrate) which have been auto-detected by the platform.

MoneyMap can be used with a minimal integration that requires very little developer time or maintenance for partners, or it can be used in conjunction with the full suite of MX products in order to gain the full power of our APIs, data enhancement, and analytics.

REQUIRED TOOLS

MoneyMap *requires* the SSO API to get one-time use URLs for loading a requested widget.

MoneyMap requires the use of *either* the MDX Real Time API *or* the Batch API (beta) to create users on the MX Platform.

Using MoneyMap with only these minimally required tools is often called an "SSO-only integration."

COMMON INTEGRATION OPTIONS

Beyond the minimal requirements above, MoneyMap is often used in conjunction with the full capabilities of the *MDX Real Time or MDX On Demand APIs*. This allows partners to send basic transactional data directly to the MX platform on the fly rather than using *only* the aggregation built into MoneyMap. This type of integration is sometimes called a "full integration."

For an even deeper integration, partners may choose to include the *Nexus API*. This allows partners to gain API access to some of the features already built in to the MoneyMap user interface such as budgets, categories, goals, cash flow, net worth, and more. Partners may wish to do this when they want to use some of the MoneyMap widgets, but want to build their own custom user interface for another feature. This is sometimes called a "deep integration."

MoneyMap also has built-in support for MX's *Insight & Target* analytics and messaging suite as well as our *Discovered Accounts* analytics service. It can also be used with our *Reporting API* (sometimes called the Delta API) as well as our *webhooks* service.

MDX Real Time and On Demand

The MX Data Exchange (MDX) specification includes all the necessary data objects and data fields for recording and displaying basic financial information such as transaction amounts, account balances, holdings values, and other standard fare. Partners use the MDX APIs based around that specification to send this end-user data to MX. These APIs are not used for external account aggregation.

MDX Real Time is a standard RESTful API built around the MDX specification. It allows partners to send data to MX servers using simple HTTPS requests.

On Demand is also built around the MDX specification; however, in this case, the partner sets up its own server and *MX makes requests to the partner* rather than the other way around. MX pulls data from the partner's servers and processes it on the MX platform. However, Real Time is still used to create users and members.

Depending on the needs of the partner, using one more of the MDX APIs may be a good fit.

COMMON INTEGRATION OPTIONS

Either of the MDX-based APIs can be used in conjunction with any other MX product. They are commonly used together with MoneyMap as part of either a full or deep integration.

Nexus API

Nexus is MX's fully-featured RESTful API, which includes dozens of advanced data objects that allow partners to easily build customized and bespoke user interfaces backed by the full power of the MX platform. It includes budgeting features, categorization classification, cleansing, net worth, cash flow, scheduled payments, and much, much more. In short, if you're looking to build a complete money management platform, you can do it with Nexus.

REQUIRED TOOLS

Nexus *requires* the use of the SSO API for authentication and the Real Time API to create users.

COMMON INTEGRATION OPTIONS

Nexus can be used in conjunction with any of the other MX products and tools listed here. It is commonly used with one or more MoneyMap widgets. Partners often also use the Insight & Target analytics and messaging suite and/or our Discovered Accounts analytics tool.

Reporting API

Also called the Delta API, Reporting delivers daily change files for select resources on the MX platform. It provides access to AVRO files that indicate which resources have changed, how they changed, and when they changed. Files are generated daily and can be downloaded for up to seven days.

Reporting can also provide a one-time, full-data historical snapshot at the partner's request with an additional statement of work.

COMMON INTEGRATION OPTIONS

Reporting requires only that the partner have data on the MX platform and that the data change over time. Therefore, reporting can be used with any combination of MX products including MoneyMap, MDX, or Nexus.

SSO API

The SSO API is used for only two purposes: authenticating users and providing URLs for loading MoneyMap widgets.

It is therefore required for use with Nexus and MoneyMap.

Sherlock API

The Sherlock API takes MX's incredible data cleansing and makes it available as a stand-alone RESTful API that requires no other integration options.

Partners can provide a list of raw transactions and Sherlock will return the same list, but cleansed, categorized, and classified. It will also return merchant and merchant location information when it is available.

Sherlock is useful for partners who want to build their own money management solution in-house, but want easy access to MX's superior data enhancement technology.

No data is persisted on MX servers when using Sherlock.

Webhooks

MXs can also alert partners to specific events on the MX platform in near-real time through webhooks. These are HTTPS `POST` requests made to a service specified and hosted by the partner. These can alert the partner to when a `member` has aggregated, when `account` data is available, when a user has updated a `budget`, when a user has created a new custom `category`, or any number of other specific events.

Partners may wish to use this tool to improve and expand their MoneyMap, MDX, or Nexus integrations.

Authentication and security

MX implements a number of security features to protect the sensitive financial data that we deal with on a regular basis.

Security feature	Description

Security feature	Description
Client ID	Each `client` a unique `client_id` that is used as an additional layer of authentication when making certain requests on one of MX's APIs and in some other situations as well.
MD-API-KEY	MX assigns each `accessor` a unique API key that is used when making requests to one of MX's APIs.
IP whitelisting	Partners must provide a list of IP addresses that have to be whitelisted. MX will reject requests that do not come from whitelisted IP addresses. Conversely, a partner must whitelist MX's IP addresses in cases where MX makes calls to their system, such as with MDX On Demand or webhooks.
TLS	All requests to the MX platform must be made using TLS 1.2 or higher with known-secure ciphers.

For requests that MX makes to partner servers, partners can choose from any combination of the following:

- IP whitelisting;
- Mutual authentication;
- HTTP basic authentication;
- HMAC signing of requests.

MX IP address ranges

MX maintains multiple data centers for failover and load balancing purposes. Each data center has a range of IP addresses and any IP address in these ranges can be active.

The DNS entries for the MX Platform API URLs can resolve to any IP address in our ranges and can change at any time. If you choose to whitelist the IP addresses that the API URLs are allowed to resolve to you must whitelist all of the IP address ranges.

When MX initiates an MDX On Demand request to a partner's MDX service, it can originate from any of our IP addresses. If you choose to whitelist IP addresses that are allowed to, reach your MDX On Demand service you must whitelist all of the IP address ranges.

CIDR	Range
64.77.254.32/27	30 addresses from 64.77.254.33 to 64.77.254.62
68.142.151.128/26	62 addresses from 68.142.151.129 to 68.142.151.190
97.75.178.32/27	30 addresses from 97.75.178.33 to 97.75.178.62
192.41.25.128/26	62 addresses from 192.41.25.129 to 192.41.25.190
192.41.58.128/26	62 addresses from 192.41.58.129 to 192.41.58.190

IP address ranges change log

Date	Change
2016-09-16	The 192.41.25.128/26 range was added
2016-08-17	MX retired the 209.90.81.64/26 range that was previously in use. Partners can remove it from their whitelist.
2015-09-24	The 64.77.254.32/27 range was added
2015-09-24	The 97.75.178.32/28 range was increased in size to 97.75.178.32/27

Changelog

Date	Documentation Version	Description of Changes
2019.10.24	2019.008	**Overview, Types of Integrations, Best Practices, FAQ** Combined and re-wrote several sections (the integration guides and the overview) into one page called "Introduction to the MX Platform." Removed the "Types of Integrations" page. Moved the section on MX test credentials to the "FAQ and Best practices" page after the section on performance testing. *MX Reference Number: G-11*
2019.10.07	2019.007	**Best Practices** Updated the best practice about the deprecated *List URLs* endpoint to recommend using *get configurable widget URL* instead; several links were also updated. *MX Reference Number: 1289*
2019.08.14	2019.006	**Overview** Added clarifications to the *MX IP Address Ranges* section, including the fact that both the MX Intgration and Production environments can use the same ranges. *MX Reference Number: 1259*
2019.07.09	2019.005	**Best Practices** Updated the best practice about using List URLs endpoint in SSOv3 to indicate that the endpoint is officially deprecated. *MX Reference Number: 1223*
2019.06.04	2019.004	**Best Practices** Added the "Storing data in the metadata field" article with a recommendation on how to store multiple key-value pairs in the `metadata` field. *MX Reference Number: 1218*
2019.05.30	2019.003	**FAQ** Added the "Can two clients be merged on the MX platform?" article explaining the process and requirements for merging two clients into one. *MX Reference Number: 1213*
2019.05.03	2019.002	**FAQ** Updated the "What happens to an Account that no longer is provided in the data feed?" article to address how to remove an account that a user should no longer have access to. *MX Reference Number: 1194*
2019.01.31	2019.001	**Best Practices** Added the section "When to use List Widget URLs vs Read Widget URL in SSOv3 API" explaining why partners should avoid using List URLs in their production applications. *MX Reference Number: 1132*
2018.10.31	2018.003	**Best Practices** Added the section "What to provide in a data field when a value is not available" explaining best practices to follow when a partner cannot provide a value for a data field. *MX Reference Number: 1084*
2018.08.22	2018.002	**Best Practices** Added the section "Auto-enrollment and login in workflow" explaining best practices for automatically creating users when they log in to the partner's service. *MX Reference Number: 1047*

Date	Documentation Version	Description of Changes
2018.08.20	2018.001	**Best Practices** Added the section "Data Management" explaining best practices for using all CRUD endpoints when doing an integration. **FAQ** Added the FAQ "Does MX fulfill requests from partners to delete User data?" explaining partners' responsibility for managing all data that is sent to MX servers via API endpoints. *MX Reference Number: 1046*
2017.12.08	2017.018	**Overview** Updated MX Bank guidance to include details explaining not to add duplicate connections for testing. *MX Reference Number: 903*
2017.10.31	2017.017	**Integration Guides** Updated the Overview to better describe typical integrations. *MX Reference Number: 875*
2017.09.18	2017.016	**Overview** Added Job Status information to the MX Bank test scenarios. *MX Reference Number: 843*
2017.07.31	2017.015	**FAQ** Added the following FAQs: • "What is the difference between the fields "id", "external_guid", and "guid" in MX's various APIs?" • What are the different methods of doing User management? *MX Reference Number: 805*
2017.07.26	2017.014	**Full Integration Guide** Added reference to the Best Practices article "Choosing the Best Protocol To Provide Account and Transaction Data to MX" *MX Reference Number: 797*
2017.07.25	2017.013	**Best Practices** Added the section "Choosing the Best Protocol To Provide Account and Transaction Data to MX" *MX Reference Number: 795*
2017.07.13	2017.012	**FAQ** Added the FAQ "What do I return in an MDX On Demand request for an account that has no transactions?" *MX Reference Number: 787*
2017.07.10	2017.011	**Overview** Changed the MX Bank user from "test_atrium" to "mxuser". *MX Reference Number: 783*
2017.06.29	2017.010	**FAQ** Added the FAQ "Queueing MDX On Demand Requests" *MX Reference Number: 779*

Date	Documentation Version	Description of Changes
2017.06.27	2017.009	**FAQ** Added the FAQ "Who does the metadata field belong to?" *MX Reference Number: 773*
2017.05.09	2017.008	**FAQ** Added the FAQ "What is the difference between date fields ending in _on vs. _at and which one should I use?" *MX Reference Number: 741*
2017.03.23	2017.007	**Overview** Added the "MX Test Credentials" section. *MX Reference Number: 714*
2017.03.17	2017.006	**Best Practices** Added section "Caching Institution List and Required Credentials" *MX Reference Number: 712*
2017.02.14	2017.005	**FAQ** Added FAQ: • "How can I prevent width display issues with mobile widgets?" *MX Reference Number: 684*
2017.02.08	2017.004	**Overview** Updated "Secure Protocols" section to indicate that requests not using TLS 1.2 or higher will no longer be accepted as of March 15, 2017. *MX Reference Number: 682*
2017.02.03	2017.003	**Overview** Updated wording from previous change regarding TLS 1.2 *MX Reference Number: 680*
2017.01.31	2017.002	**FAQ** Added FAQ: • "Can MDX Real Time and MDX On Demand be used together to provide account and transactions data?" *MX Reference Number: 676*
2017.01.30	2017.001	**FAQ** Added FAQ: • "When does MX make On Demand requests to retrieve a user's account and transaction data?" *MX Reference Number: 675*
2017.01.24	2017.000	**Best Practices** Added section "Handling Closed Accounts". **FAQ** Added/Updated FAQs: • "What is the difference between a User and a Member?" • "What happens to an Account that no longer is provided in the data feed?" *MX Reference Number: 671, 672*

Date	Documentation Version	Description of Changes
2016.11.18	2016.005	**Performance Testing** Added new Performance Testing guide explaining how partners can use JMeter with provided templates to test performace on their own MDX On Demand service. *MX Reference Number: 613*
2016.09.19	2016.004	**Overview** Added a new IP range 192.41.25.128/26 to whitelist section and added a "IP Address Ranges Change Log" section showing the history of MX IP Range changes. *MX Reference ID: 600*
2016.09.08	2016.003	**Overview** Added a note explaining the scope of uniqueness required for Member IDs. *MX Reference Number: 590*
2016.08.17	2016.002	**Overview** Removed retired IP range 209.90.81.64/26 from whitelist section and added a note explaining the retirement. *MX Reference Number: 575*
2016.08.03	2016.001	**Overview** Added action points to the "Overview" section. *MX Reference Number: 561*
2016.07.17	2016.000	**Overview** Added "Secure Protocols" section explaining that TLS 1.2 or higher should be used and support for TLS 1.0 and 1.1 will be deprecated on September 5, 2016. Moved "MX Authentication" section closer to the top of the list of topics in the Overview guide. *MX Reference Number: 524, 526*
2016.05.11	2015.000	Added "Getting Started" to the developer portal top menu including the following pages: • Overview • Best Practices • FAQ *MX Reference Number: 477*