Technical Reference

Identity Governance Data Collection and Publication Implementation Guide

MICRO FOCUS

Contents

Overview

The term **Data Collection** refers to the features and processes used by Identity Governance to retrieve, validate, and format entity (Identity, Group, Application, Account, and Permission) data from desired data sources. The term **Data Publication** refers to the processes used to transfer the collected data to the operational Catalog of entities in Identity Governance and make it available for Governance operations. Data collection and publication is the critical first step in the Governance process, and it is an ongoing process that is needed to ensure that the access information that is being reviewed is up to date.

Data collection also requires a very thorough understanding of the sources from which the data is being retrieved. The people with the Data Administrator role are responsible for configuring the connectivity to the various Identity and Application data sources, ensuring proper security for those connections, determining which entity data should be retrieved, and establishing the various relationships between those entities. To help facilitate these tasks, the Identity Governance product provides a large set of collector templates that contain default data and configuration settings for many common enterprise and cloud data sources. These templates are intended to help the Data Administrator get started with the configuration process. However, each environment has custom requirements that may require unique transformation and configuration options. This document is intended to help facilitate the establishment of an accurate and efficient Data Collection configuration by explaining the details of the Identity Governance collection architecture and tools.

Terminology

There are 4 basic collected Identity Governance entities that will be discussed in this document:

- Identities (or Users)
- Groups (or Identity Groups)
- Accounts
- Permissions

NOTE: Application entities are generally not collected by Identity Governance -they are configured by the Data Administrator. The exception to this rule is the I*dentity Manager Advanced Edition Permission* collector, which collects entitlement-enabled Identity Manager driver objects as applications. Since that collector has unique capabilities, it is addressed in a separate document.

Identities

Identities are the actual people that are at the core of the processes within Identity Governance. Identities are the people being reviewed for “who has access to what”. They are also the entities who are doing the reviews and have administrative roles within Identity Governance.

Groups

Groups (or Identity Groups) are collected by Identity source collectors. Groups are comprised of collected Identities and are a useful entity for assigning administrative roles or reviews to a grouping of people without incurring the administrative overhead of direct assignment.

Accounts

Accounts generally represent Application access entities. If you log into Netflix, you are using an account. If you log into Gmail, you are using an account. Accounts are NOT identities, they are the representation of System, application, or data source access BY an Identity.

Permissions

Permissions, from an Identity Governance perspective, have multiple facets. Permissions can describe what you are allowed to do within an application, or they can describe something that you have in your possession or have access to. Some examples:

- Finance employees have access to the SAP Finance application (accounts). One employee has the rights granted to run Accounts Payable functions, another employee has right granted to run the Accounts Receivable functions. Both have accounts, but different permissions within the application. This is a case where the Permissions (Accounts Receivable, Accounts Payable) are granted to the application user (account).
- All employees have an electronic badge that allows them access to their office building. These employees do not have an account on the Building Access application to which they must login, they simply have the access granted to their person via the badge. This is a case where the Permission (badge) is granted directly to the Identity (person).

Permissions may also have hierarchical relationships with other permissions. For example, a corporate Role “Cambridge Employee” may consist of various child permissions, such as “Garage Access”, “Building Access”, etc.

In Identity Governance, collection and publication of Identities and Groups is provided by ***Identity*** Collectors. The purpose of these collectors is to build a catalog of all of the people and groupings of people within an organization. The Identity catalog should contain as much personally-identifying information about these entities that is needed to uniquely identify them, and to allow Application information to be associated with them.

Information about what systems the Identities can access, what level of access they have on those systems, what equipment they have access to, what buildings, doors, printers that they can use, etc. is collected and published from ***Application*** collectors. As previously mentioned, the Identity Governance product provides various application templates that are properly configured for applications with well-known schema and behaviors, but for many sources (JDBC, CSV, REST and SOAP in particular) the schema or API is not well known and it will be the responsibility of the Data Administrator to properly configure these data collectors.

Identity Collectors

Overview

Populating the Identity Governance Identity and Group catalog is the job of Identity Collectors. As briefly stated previously, Identities are at the core of the functions of Identity Governance. All Administrative and runtime roles within the product are assigned to Identities or groups of Identities. Each Identity collector is comprised of one or more collector “views” that can be customized to match the characteristics of the data source being collected.

- **Collect Identity** view – This is the primary view for all Identity collectors. It is used to collect information about actual persons within the organization. It is critical that these views be configured to obtain as much uniquely identifying information as needed for the business, and as much information as needed to ***map*** (or ***join***)

application data to the identities. Some typical mapping attributes include *workforceId*, *email*, *fullName*, etc. If you must collect from multiple Identity sources to build a complete Identity catalog, make sure you utilize a similar strategy for selecting the mapping attributes from each source.

- **Collect Group** view – This view is for collecting organizational groups of Identities. These groups are not permission groups and are not available for access reviews. These groups are typically used to set the scope of reviews (eg: Review permissions of all members of "Finance" group) or for assigning Identity Governance administrative roles. Depending on the data source being collected, the Group view is also be used to collect the membership information between the Group and member Identities. Typical applications that support the simultaneous collection of Groups and their members are Active Directory, eDirectory, and Identity Manager. This method of collecting join information is also optimal if the Identity source does not support hierarchical Group-to-Group relationships.
- **Collect Group to User Membership** view – This view is utilized if the data source supports collection of the mapping relationship between Groups and member Identities. Typical applications that support this type of join are JDBC and CSV sources.
- **Collect Parent Group to Child Group Relationships** view – This view is utilized if the data source supports collection of the mapping relationship between parent Groups and child Groups. Typical applications that support this type of join are SAP, JDBC, and CSV sources.

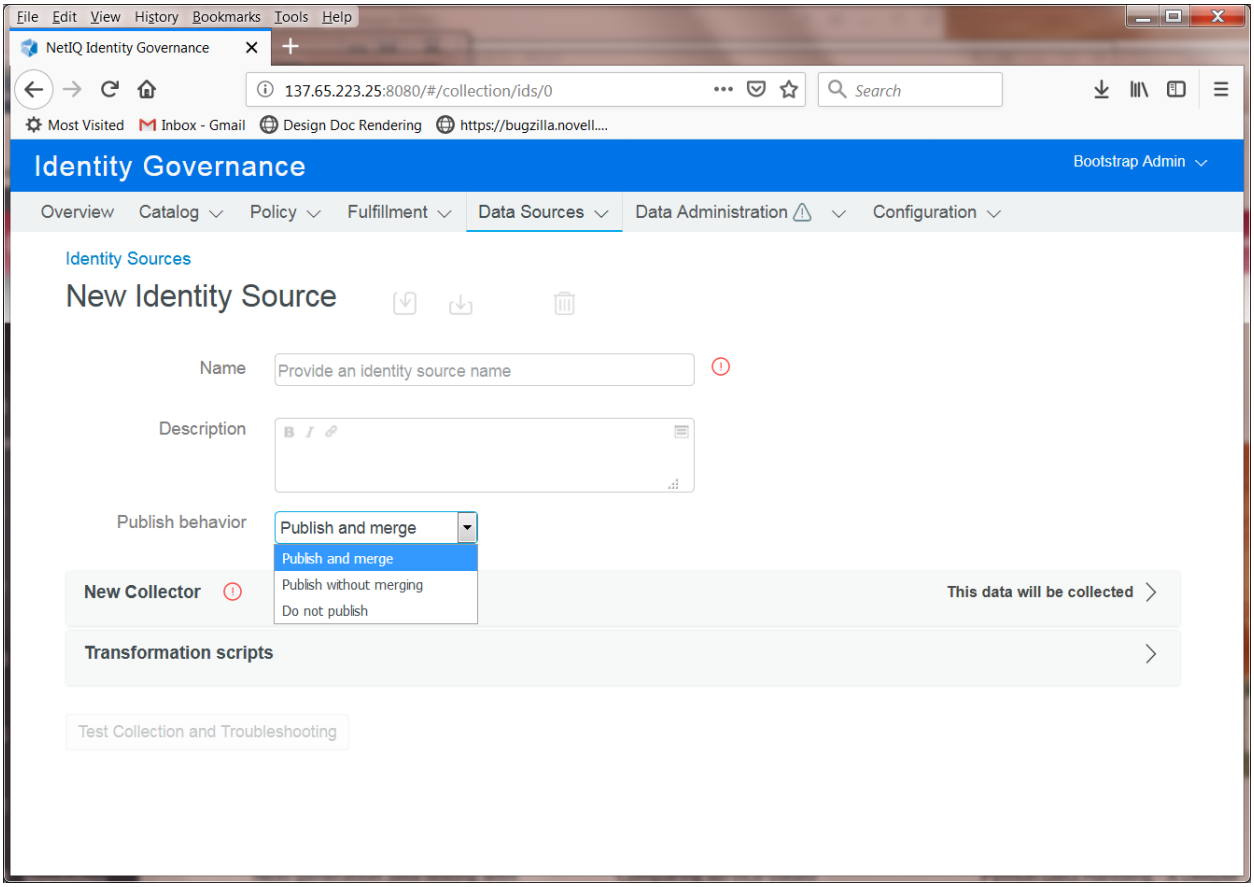
Notes:

1. Each of the enabled views is actually a distinct collection of data from the target system. For best performance, views should only be enabled if they are necessary to obtain the desired data and mapping information. For example:
    - If your data source does not contain (or if you are not interested in) Identity Group collection, disable all of the Group collection views.
    - In Active Directory it is possible to collect group information, User member information, and child group member information using the ***Collect Groups*** view. Disable all other Group collection views.
    - In eDirectory it is possible to collect group information and User member information using the ***Collect Groups*** view. However, the group to child group relationship utilizes a different membership attribute and must be collected using the ***Parent Group to Child Group Relationships*** view.
2. The ***Collect Group to User Membership*** view ALWAYS maps the relationship using the values of the *User ID from Source* attribute from the ***Collect Identity*** view and the *Group ID from Source* attribute of ***Collect Groups*** view from the same source. This is NOT configurable at this time.
3. The ***Collect Parent Group to Child Group Relationships*** view ALWAYS maps the relationship using the values of the *Group ID from Source* attribute of the ***Collect Groups*** view from the same source. This is NOT configurable at this time.

Mapping and Merging Identity Sources

Since all Identities collected from various sources are populated into a single catalog, it may be necessary to perform matching and merging operations on the data collected from multiple sources. To determine how to configure your

multiple Identity Sources, you need to understand the behavior of the following Identity Source publication options available during creation of your sources:



- **Publish and merge** – If you have multiple sources of Identity information and there is ANY chance that the same identities may exist in more than one source, you should choose this option.
- **Publish without merging** – If you have only one (1) Identity source OR if you have multiple sources and there is NO chance that the same identity exists in more than one source, you should choose this option. Performance is better with this option since no matching/merging (unification) must take place. You should be aware that regardless of whether or not a matching object from another source exists in the Identity catalog, this option will create Identities for each object collected.
- **Do not publish** – If you are in the early stages of configuring an Identity collector and you do not wish to publish the collected data into the Identity store, you should choose this option. This is a good option for creating Identity source prototypes for testing.

Publish and merge

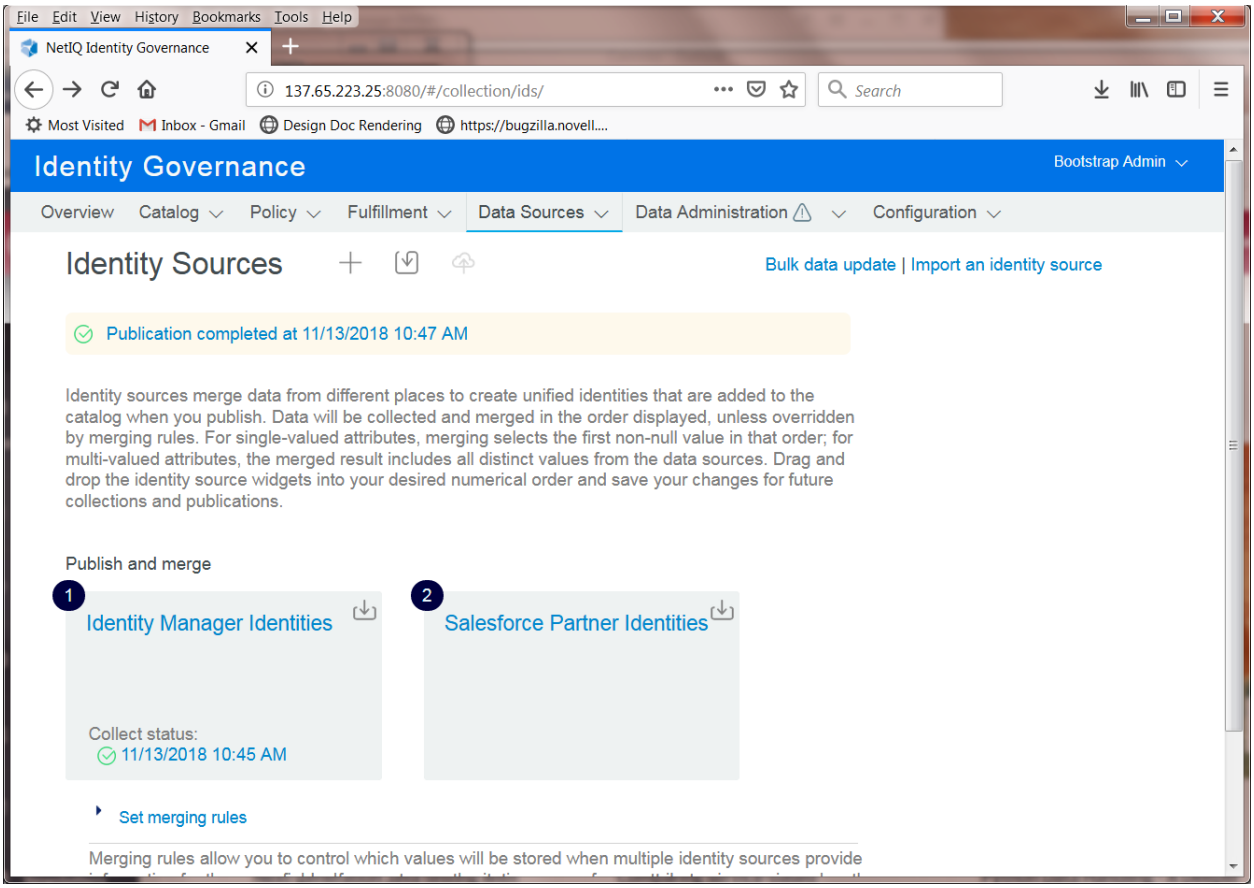
When using the ***Publish and merge*** option for your Identity Collectors, you will need to plan out the following:

- **Identity Source Order** - The order in which your Identity sources will be published
- **Attribute Matching** - The attribute(s) that will be used to match records from each source to Identities in the catalog
- **Merging Rules** - The preferred source of your Identity attributes.

You must also keep in mind that each Identity Source collector configured for ***Publish and merge*** can potentially create new Identities in the catalog. Therefore you should always ensure that the mandatory *User ID from Source* attribute mapping is configured to collect an acceptable unique identifier that is appropriate for the catalog.

Identity Source Order

On the top-level page of Data Sources > Identities, you are presented with a view of your Identity Sources similar to the following:



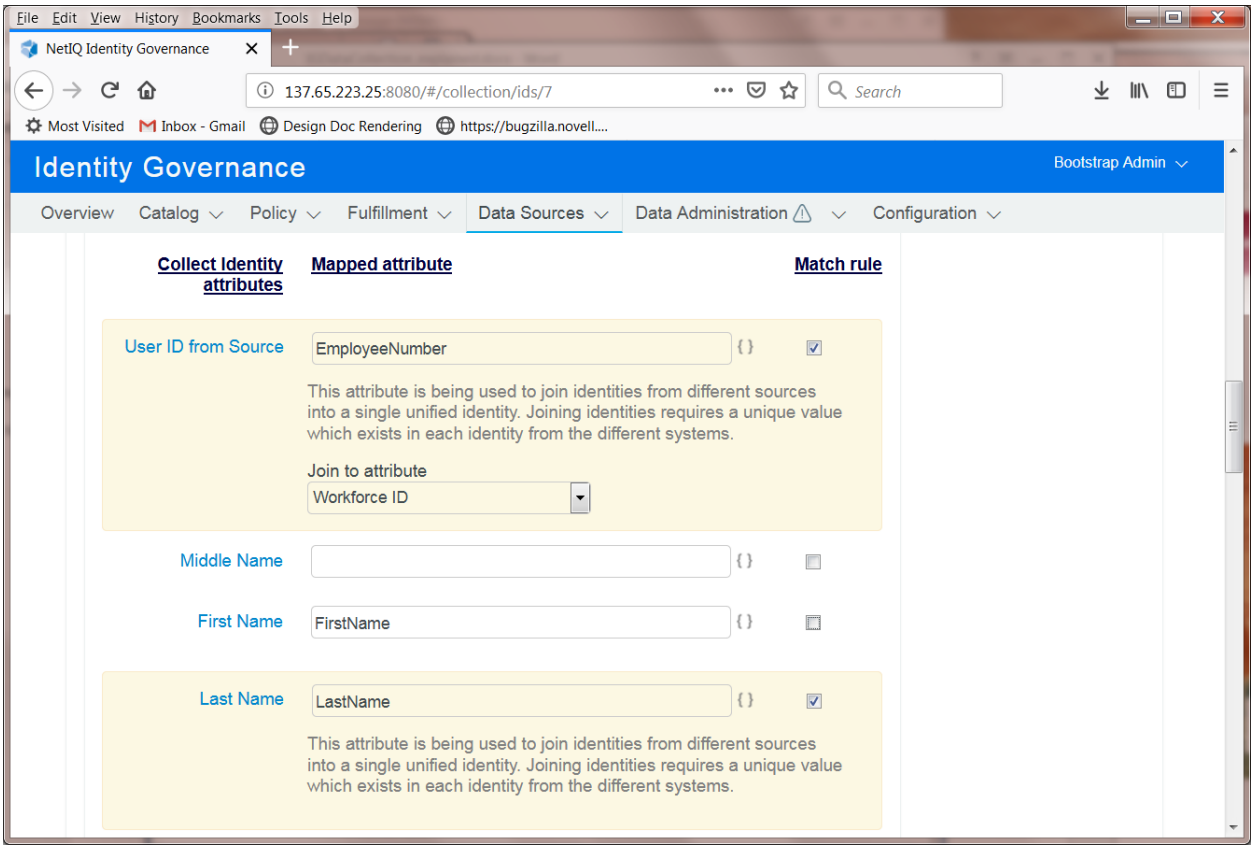
In general, it is desirable to place your most complete and authoritative source in position “1”. To change the order of publication of your Identity Sources, simply drag-and-drop the source to its desired position.

Attribute Matching

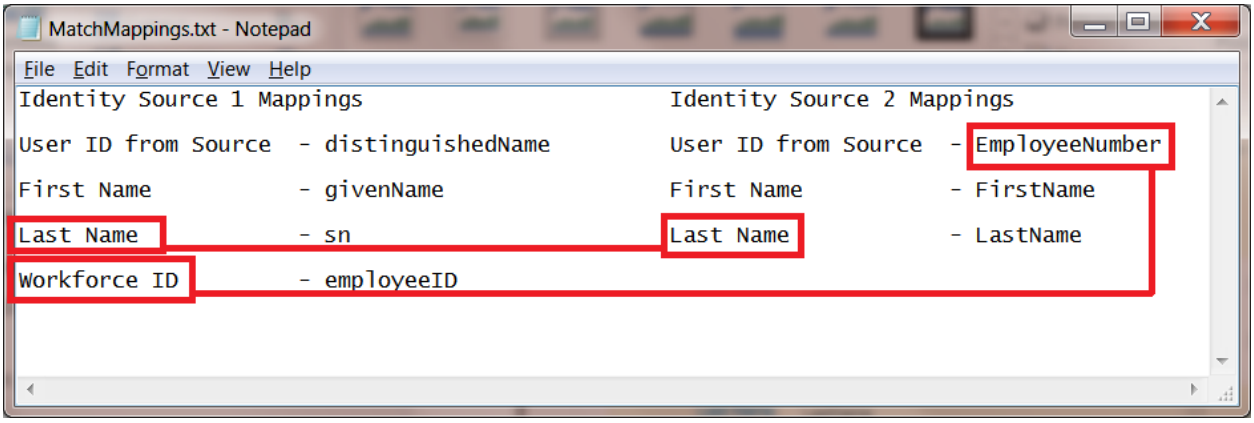
While editing the configuration of a ***Publish and merge*** collector, the schema mapping UI will present a “Match Rule” checkbox next to each attribute mapping row. You must select at least one matching attribute before you can save the configuration. Check the boxes next to the attribute values you wish to use for matching to Identities in the catalog. The following applies to matching attributes:

- Each collected identity record MUST contain a value for the matching attribute(s) or the record will be discarded by the collector as a “non-match”
- Each matching attribute must specify a “Join to attribute” which indicates which Identity attribute in the catalog will be used for matching. In most cases, the value of the “Join to attribute” selection will be the same as the Identity Governance “Users attribute”

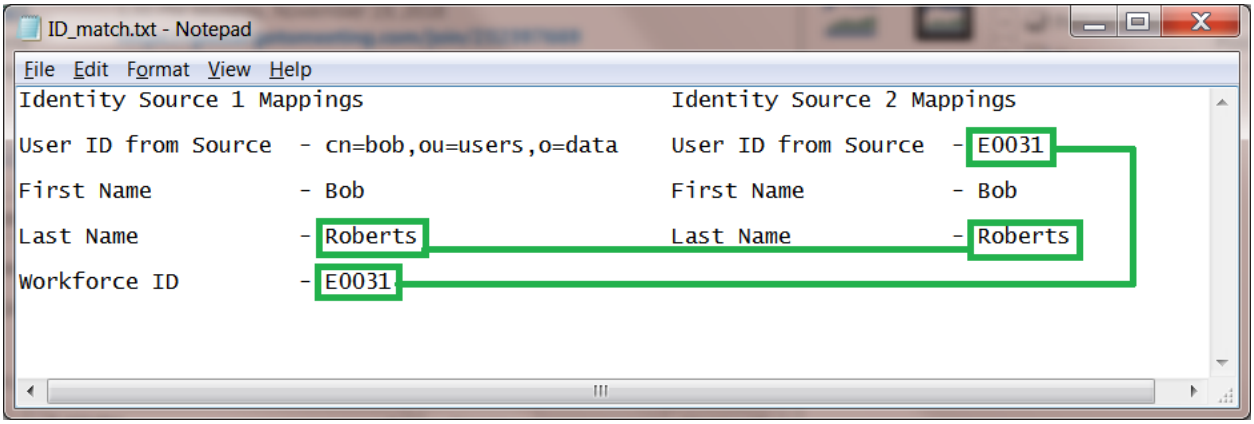
The following screen shows an Identity collector that will attempt to match Salesforce attribute *EmployeeNumber* to the "Workforce ID" of previously collected records AND also match attribute *Last Name* to the "Last Name" of previously collected records:



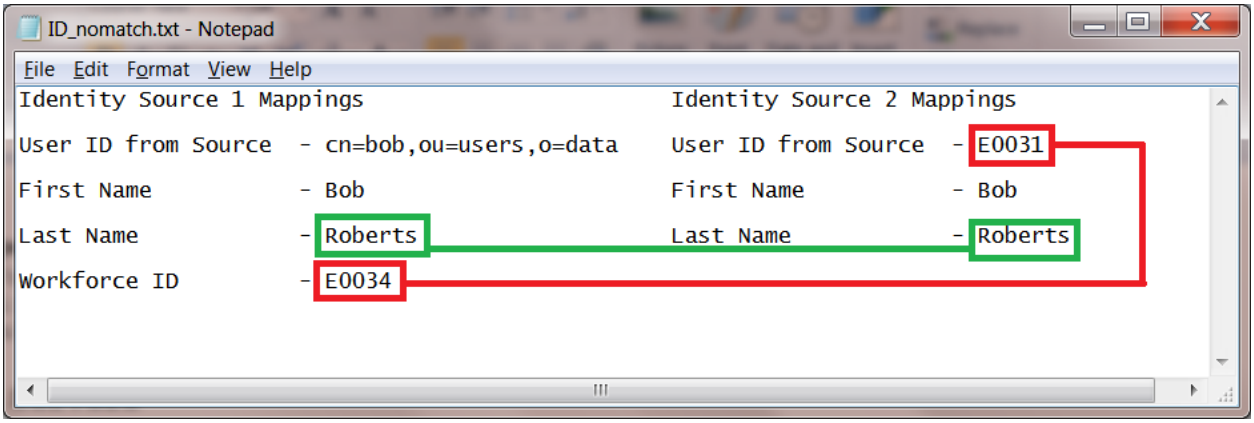
Here is a sample of data to illustrate the mapping specified above:



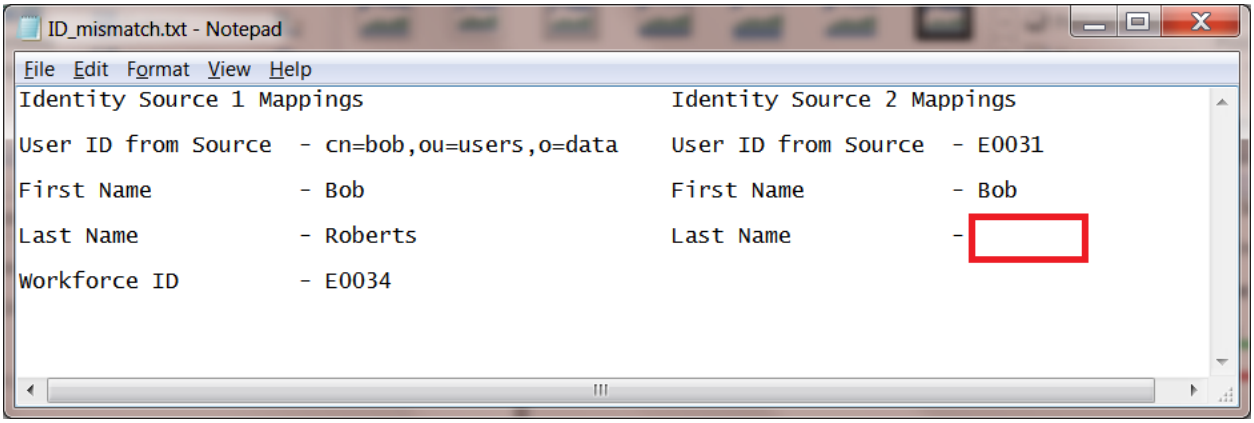
Now plug in a record from 2 sources. This illustrates a successful map and would result in a single, unified identity:



However, if both fields contain values, but EITHER of the two field values do not match, the result will be an addition of a new Identity record from the second source:



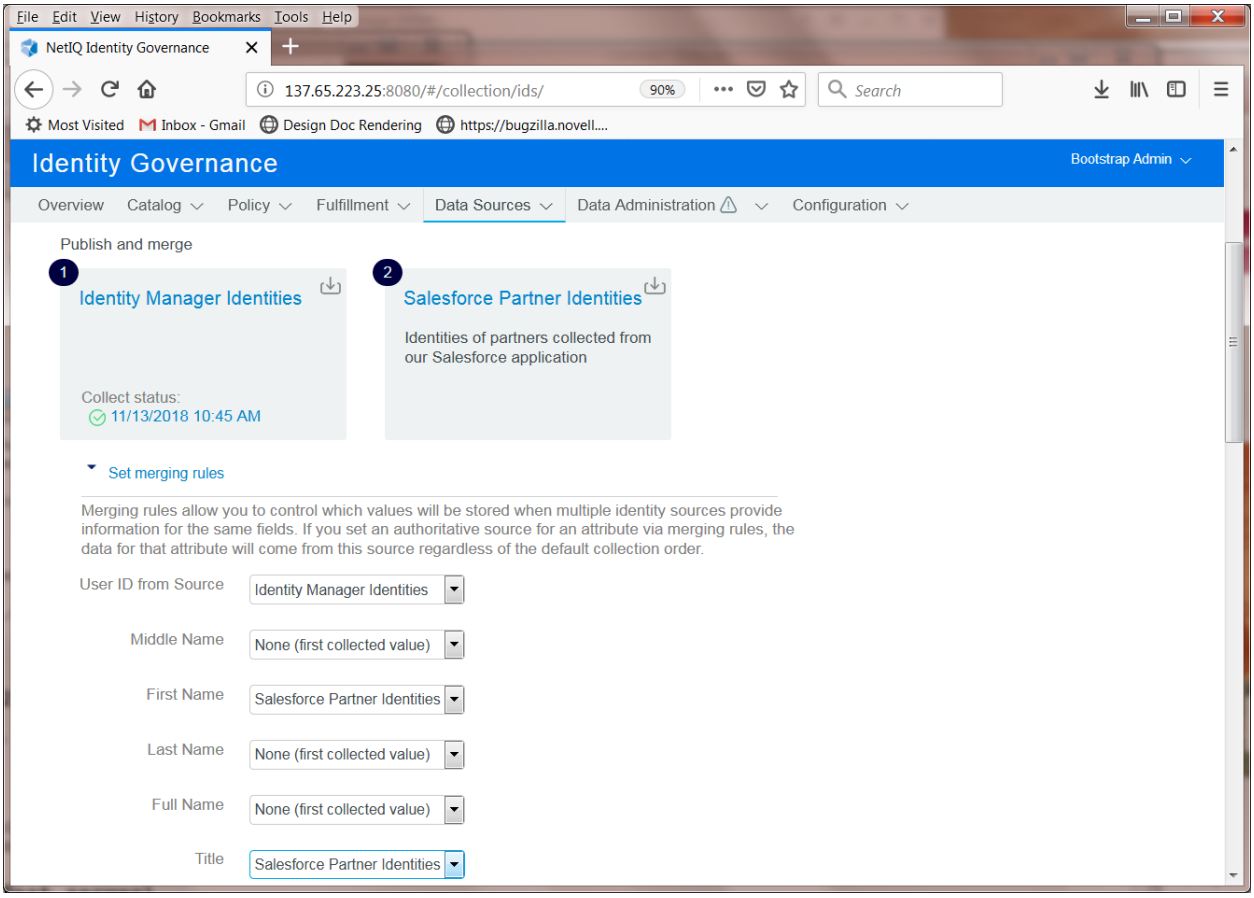
And finally, if any of the selected matching attributes are not present in the second source (even non-mandatory attributes) the result will be that the record from the second source is discarded:



Merging Rules

When you have multiple sources of Identities that will be merged, you will quite often discover that multiple sources can provide the same attribute information (eg: lastName, email, etc.). By default, Identity Governance will utilize a “first collected value” mechanism. This means the first source that provides a value will set the value for the merged Identity. If you wish to modify that behavior to specify a desired source for any attribute value, you can set your preference using the

Set merging rules function on the Identity Sources page. If your specified source does not provide a value for any particular record, the default “first collected value” will be applied.



Collection and Publication

Once all Identity collectors have been configured, they can be individually collected or a collection schedule may be configured. Since the Identity catalog is comprised of the data contributed by all published sources of Identity data, you must perform a Publication of Identity data only after you have performed a collection from all sources. The Publication process will unify your collected data sources and populate the Identity catalog.

Account Collectors

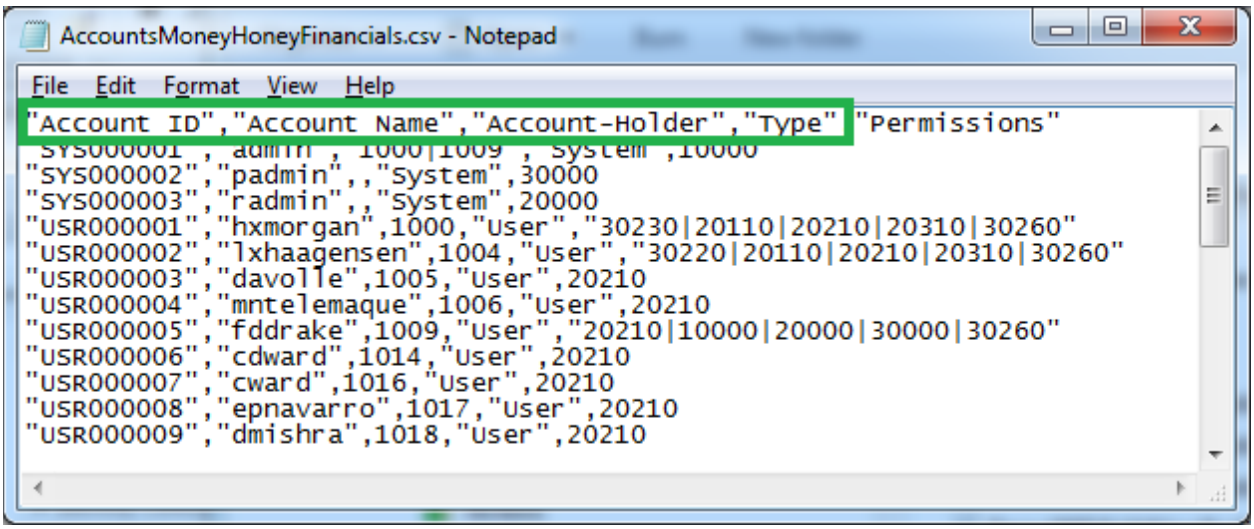
Overview

Account collectors are one of the collector types available within an Application Source. As previously discussed, whenever you have a system or application that you log into you are dealing with an Account. In general, personal information about the account holders is not maintained within application data stores since it is not needed for operation of the application. These systems hold some Account Identifier (or login ID), a password, and the set of Permissions that have been granted to the Account users (group memberships, roles, ACLs, etc.). In a typical enterprise, there will also be some account attribute (or combination of them) that can be used to associate (or “join”) the Account to the Identity that uses the account. However, this is not true for all accounts. Many systems have “admin”, or “system” accounts that are used by IT staff and administrators to maintain the system, grant access to others, etc. Often, these “admin/system” accounts are granted the greatest level of permissions for the system. Additionally, sometimes these “superuser”

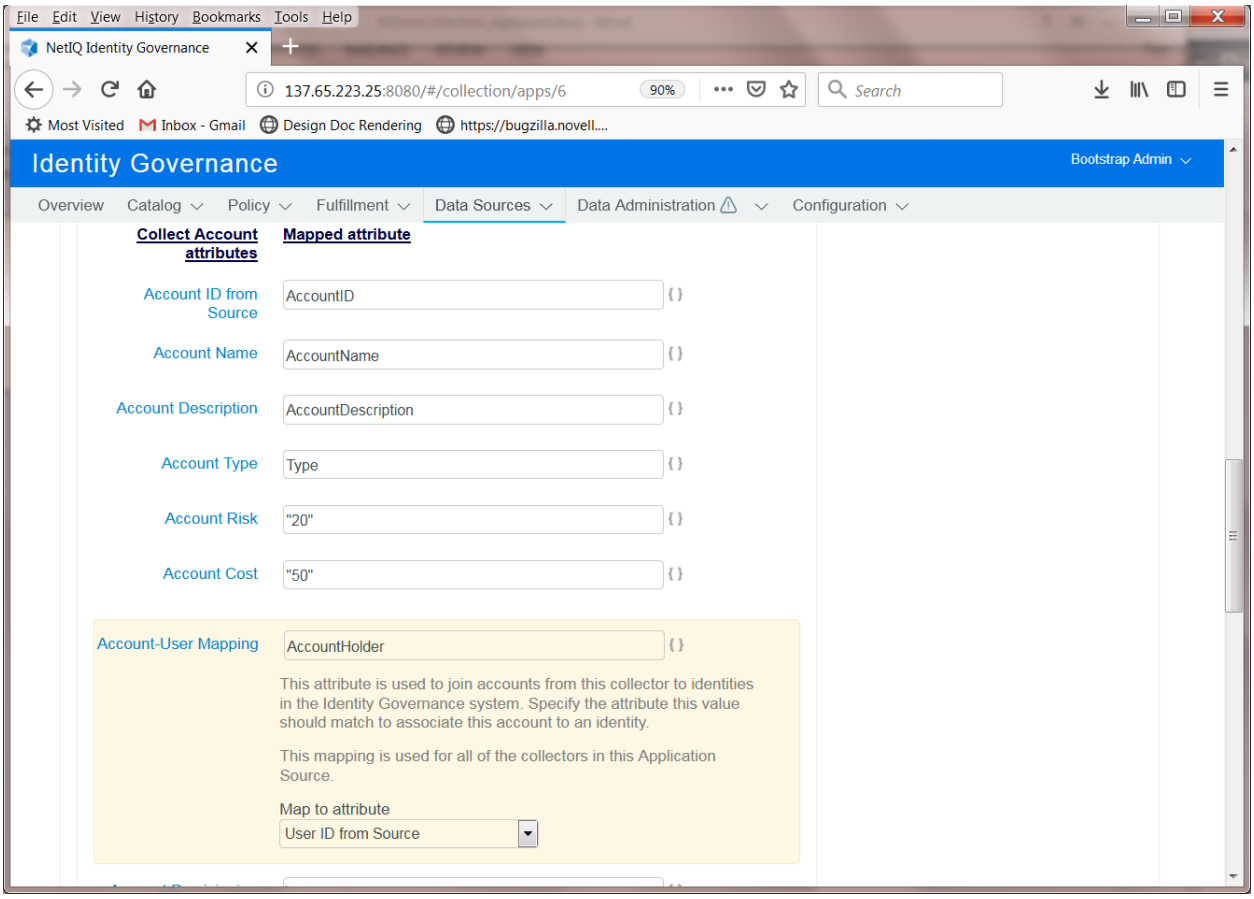
accounts are shared by a group of individuals. As a result, it is very important to collect and review ALL accounts from the data source whether they can be Joined to an Identity or not.

Collection Scenario

In the following scenario, we are dealing with a financial application that does utilize accounts. In setting up the Account collector, we will need to analyze the data source to determine what information is available for us to collect and how it can be joined to our corporate Identities:



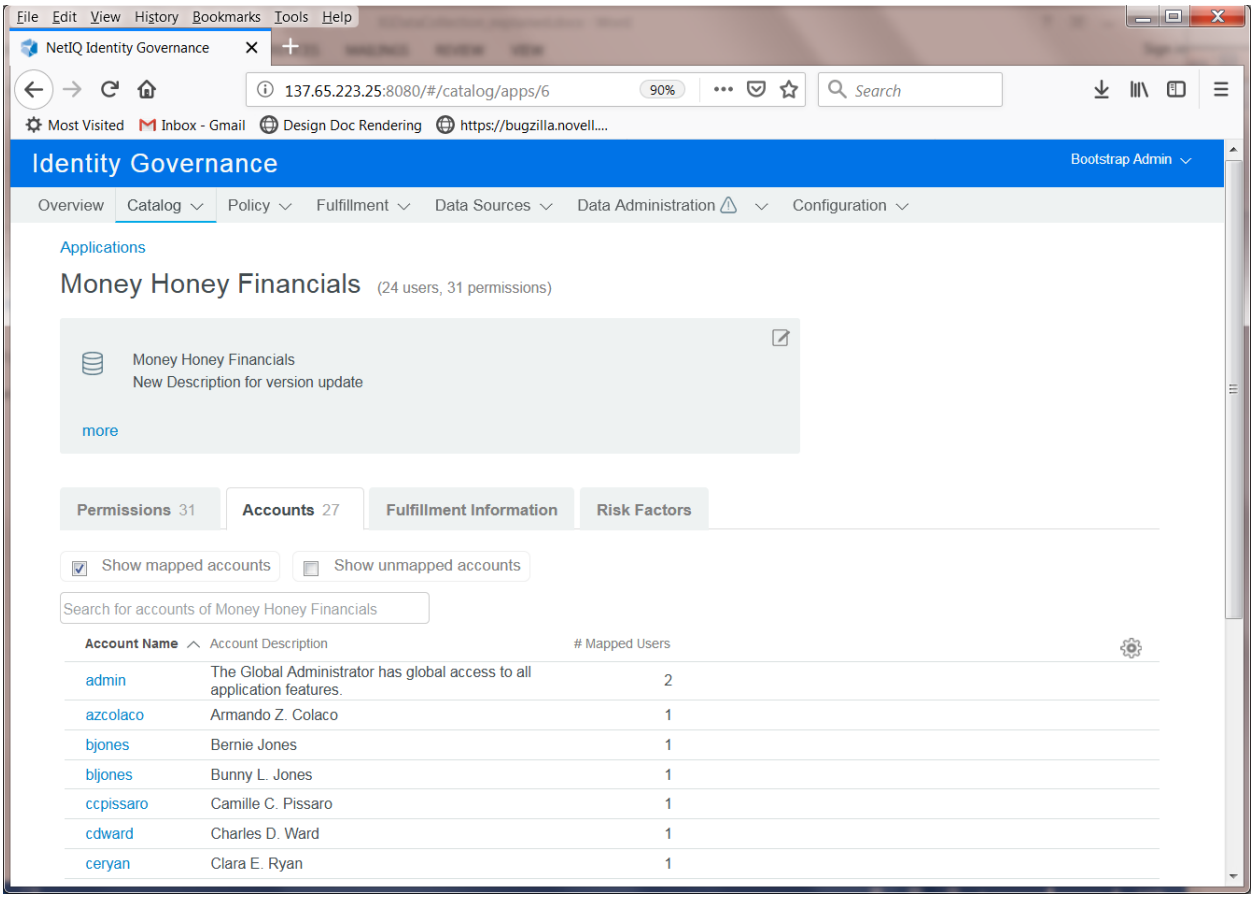
As we can see, we are fortunate with this source because we are provided with Account Identifier information as well as Account Holder information. We may also know that the cost of this application is $50 per seat, and the risk level of each account is a value of 20 (provided by our security team). We can now set up our Application Source and Account collector:



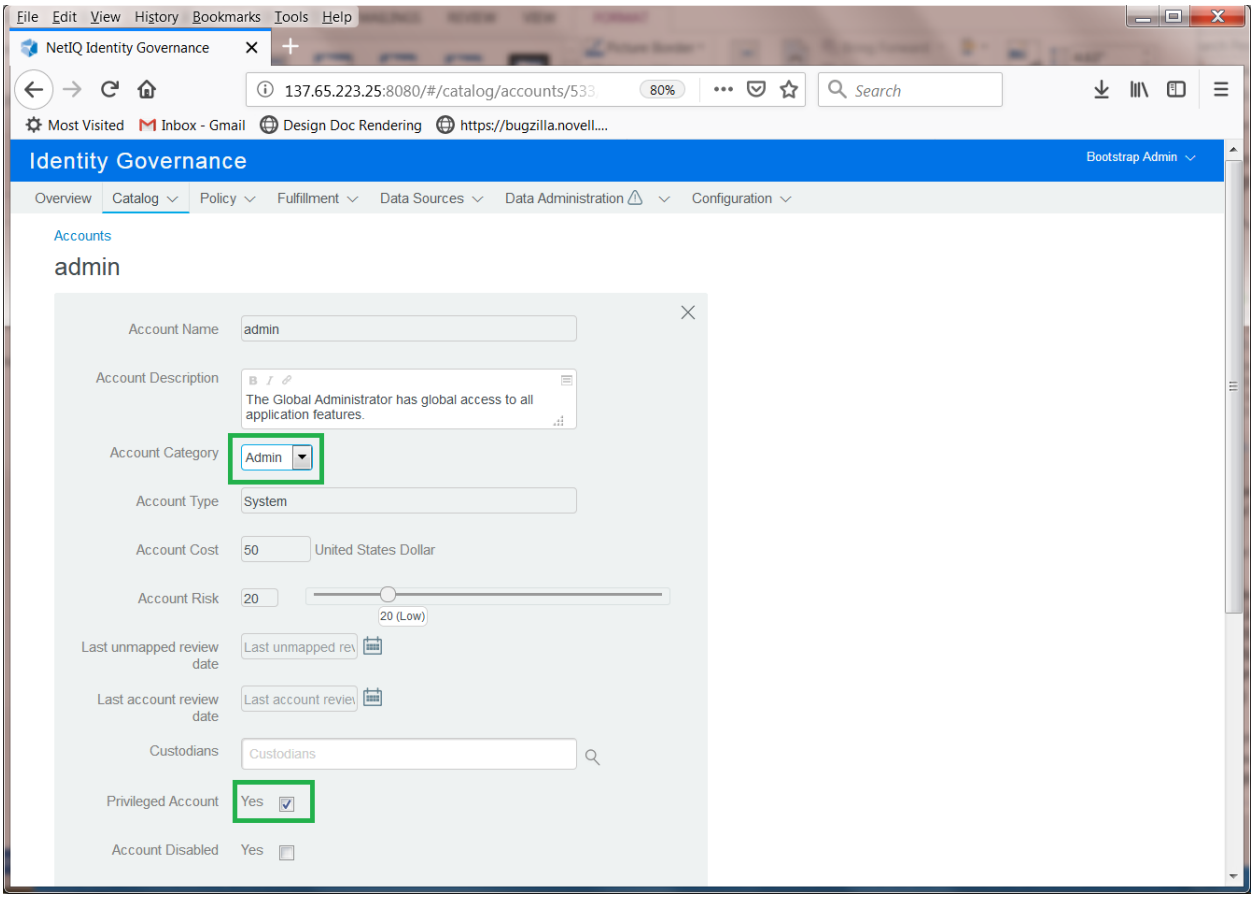
The Account is mapped to an Identity using the *Account-User Mapping* attribute. Note the mapping of the *AccountHolde*r field from the CSV source and the “Map to attribute” value specifying the *User ID from Source* field of our previously collected Identities. Also note the two “static value” mappings of the cost and risk associated with the collected accounts. Most attributes may be mapped to a static value simply by enclosing the desired value within double-quotes. These static values will be applied to each record collected.

Since each Application source contains data that is independent of other Application sources, it is possible to publish your data immediately after collection – in other words, there is no match/merge procedure between Application sources.

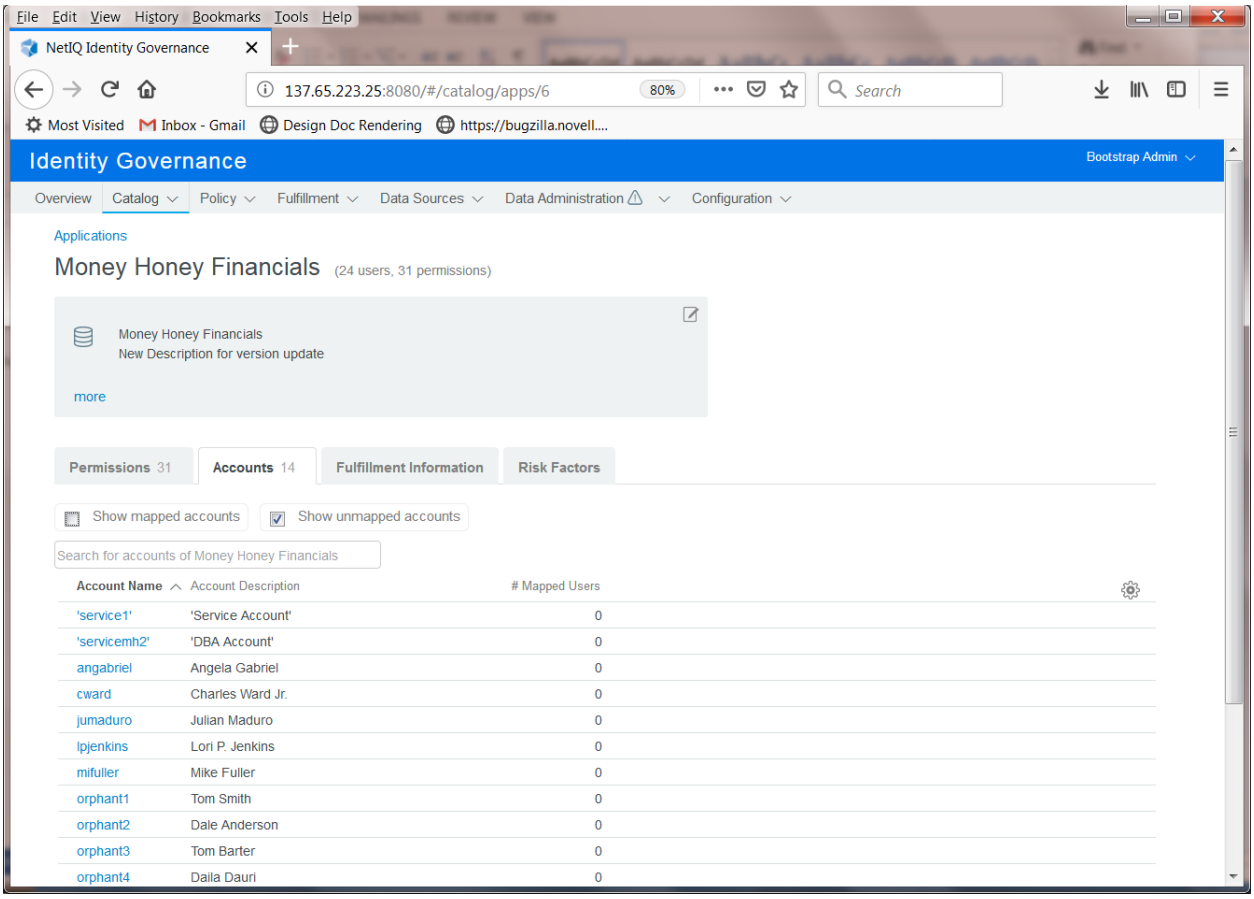
After collecting and publishing our data, we may examine our collected accounts in the Identity Governance Catalog:



In this view, we have selected the “Mapped Accounts” checkbox to show all of the accounts that have been successfully mapped to an Identity. If you click on the account names, you are linked to the Account details page where you may see which Identity or Identities have been mapped to the account. In this sample, the “admin” account was mapped to multiple Identities due to the multi-value mapping in the CSV source. If desired, you may also Edit/Curate the data from the details page to add or modify information that could be helpful in reviews of the accounts, such as a Category and Privileged status of the account:



It is often the case that an application source contains System, Administrative, or Orphaned accounts that are not associated with actual Identities. In many cases, these accounts have some of the greatest privileges within the application and are thus the most important to review. To see the accounts that have not been mapped, select the “Unmapped Accounts” checkbox on the Application details page in the Catalog:

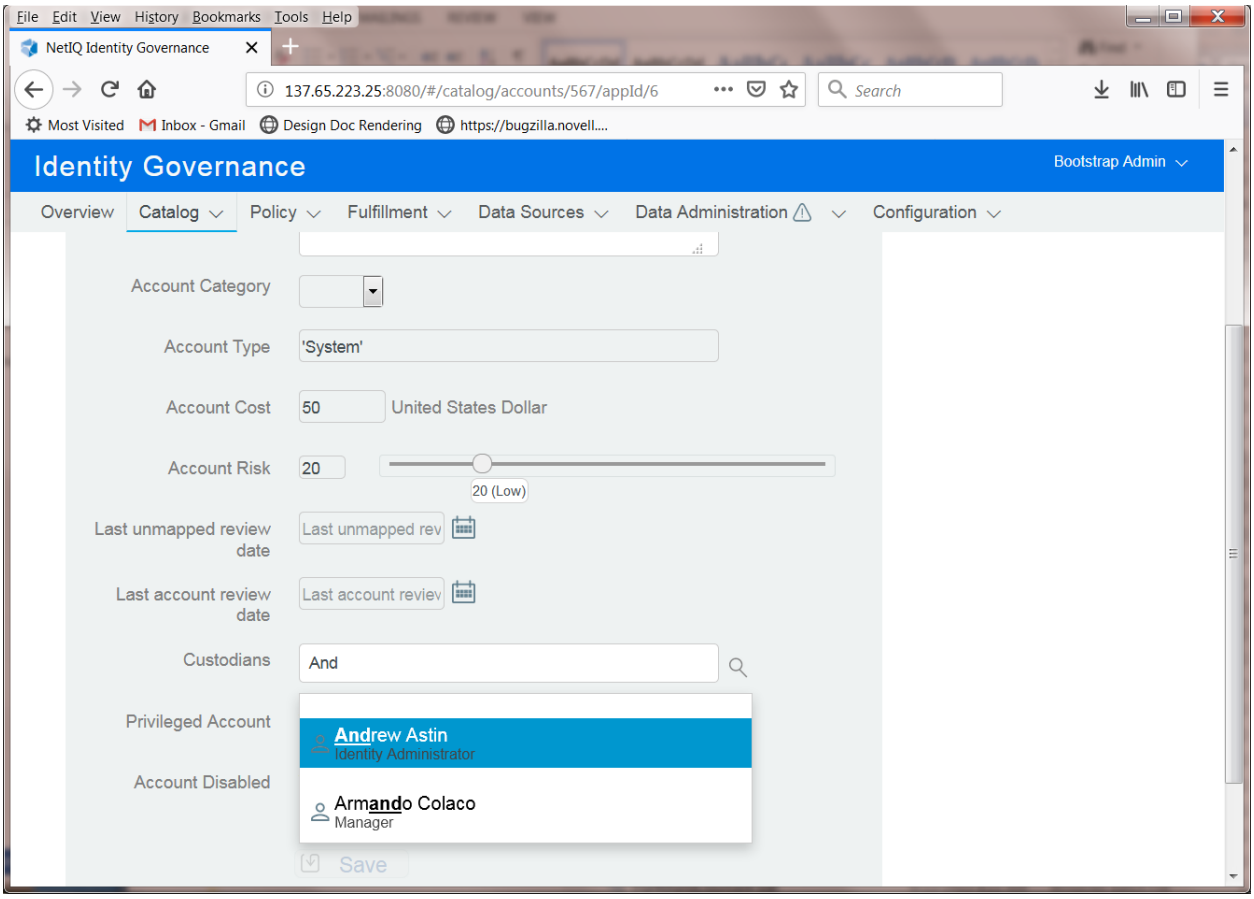


Both mapped and unmapped accounts may be included in reviews.

Account Custodian Mapping

In many customer scenarios, there is a desire to assign an owner or custodian to an account. Typically, the custodian is needed for System or Administrative accounts that are not directly linked to an Identity. Some Application sources (eg: eDirectory and Active Directory) support the concept of an owner which can be collected directly from the system. If the data can be collected, the *Account Custodian* attribute is mapped in the Account collector configuration and is joined to an Identity in a manner similar to the *Account-User Mapping* attribute described previously. If the data cannot be collected, or if collection of the source value is not desirable, one or more custodians may be added by editing the account and selecting Identities that exist in the catalog as shown below.

NOTE: At this time, only Identities may be assigned as Account Custodians.



Permission Collectors

Overview

Permission collectors are one of the two types of collectors that can be created for an Application source. As described in the *Terminology* > *Permissions* section of this document, permissions come in a variety of forms. From a data collection perspective, the purpose of a permission collector is to:

- Collect the set of permissions and descriptive attributes for each permission type
- Collect the hierarchical relationship (if any) between permissions
- Collect the data that will allow us to join permission assignments to Identities or Accounts

There are a great number of varieties in the way that permissions and the various relationships are described within different Application sources. In order to accommodate this variety, Identity Governance provides a large number of ways to configure Permission collectors by using views:

- **Collect Permission** view – Used to collect the available permission values and descriptive information about the permission. If the permission schema contains the information needed to establish permission hierarchy and/or join permission values to Identities or Accounts, this view may be utilized to perform those functions also – with the benefit of configuration simplicity and better performance. Some examples of applications that utilize this type of combined view are the Active Directory and eDirectory “Group” permission collectors.

- **Collect Holder to Permissions Mapping** view – This view is utilized when the information about assigned permissions is contained within a source that has the holder-to-permission relationship defined on the holder (Account of Identity) records. Some examples of applications that use this method exclusively are Salesforce.com and SAP. An example of this relationship would be the “memberOf” attribute on Active Directory User objects.
- **Collect Permission to Holders Mapping** view – This view is utilized when the information about assigned permissions is contained within a source that has the holder-to-permission relationship defined on the permission records. An example of this relationship would be the User “members” attribute on eDirectory Group objects.
- **Collect Permission hierarchy based on parent to child** view – This view is utilized to collect top-down permission relationships. An example of this relationship would be the Group “members” attribute on eDirectory Group objects.
- **Collect Permission hierarchy based on child to parent** view – This view is utilized to collect bottom-up permission relationships. An example of this relationship would be the Group “memberOf” attribute on eDirectory Group objects.

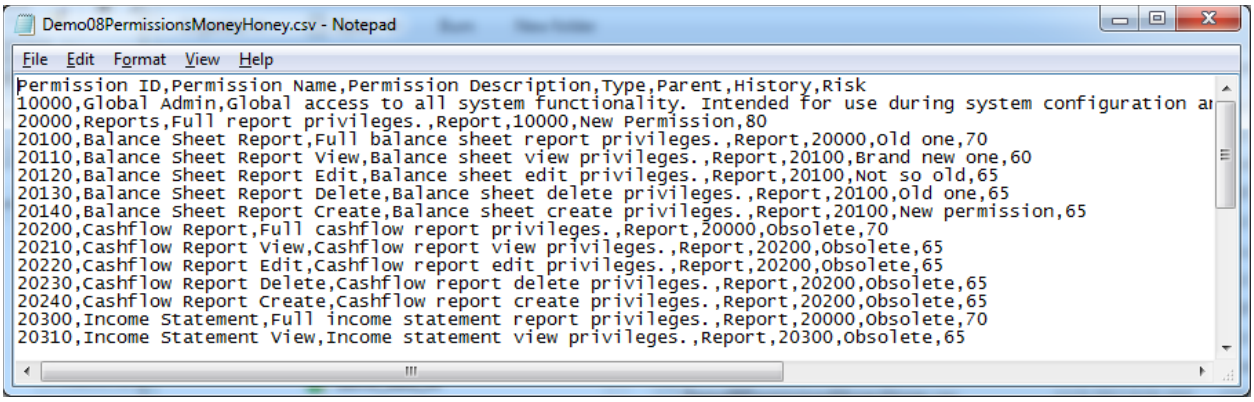
Permission Collection and Joins

The primary purpose of these views is to get the detailed information about the permission objects or values of a selected permission type. Identity Governance provides templates for the various application types it supports, and these templates contain sample Permission collectors for the traditional permission types for those applications. (This is very similar to the concept of “Entitlement” objects in Identity Manager). However, it is possible to describe virtually anything as a permission. A parking pass, a keycard, a laptop, a mobile device, etc. could all be described as a permission, and thus Identity Governance allows you to extend or modify the application templates as needed for the use-case.

Regardless of the Permission type, there is an assumption that there exists a finite set of permission values that may be assigned to Application users or accounts. If the permission type is “Groups”, then there may be a container of group objects in LDAP or a CSV file of all existing groups that may be collected. If the permission is type “laptop”, then there may be a database table that contains the list of all laptops within the organization. The job of the Permission collectors is to consume this source data and create a catalog of permissions for the Application source.

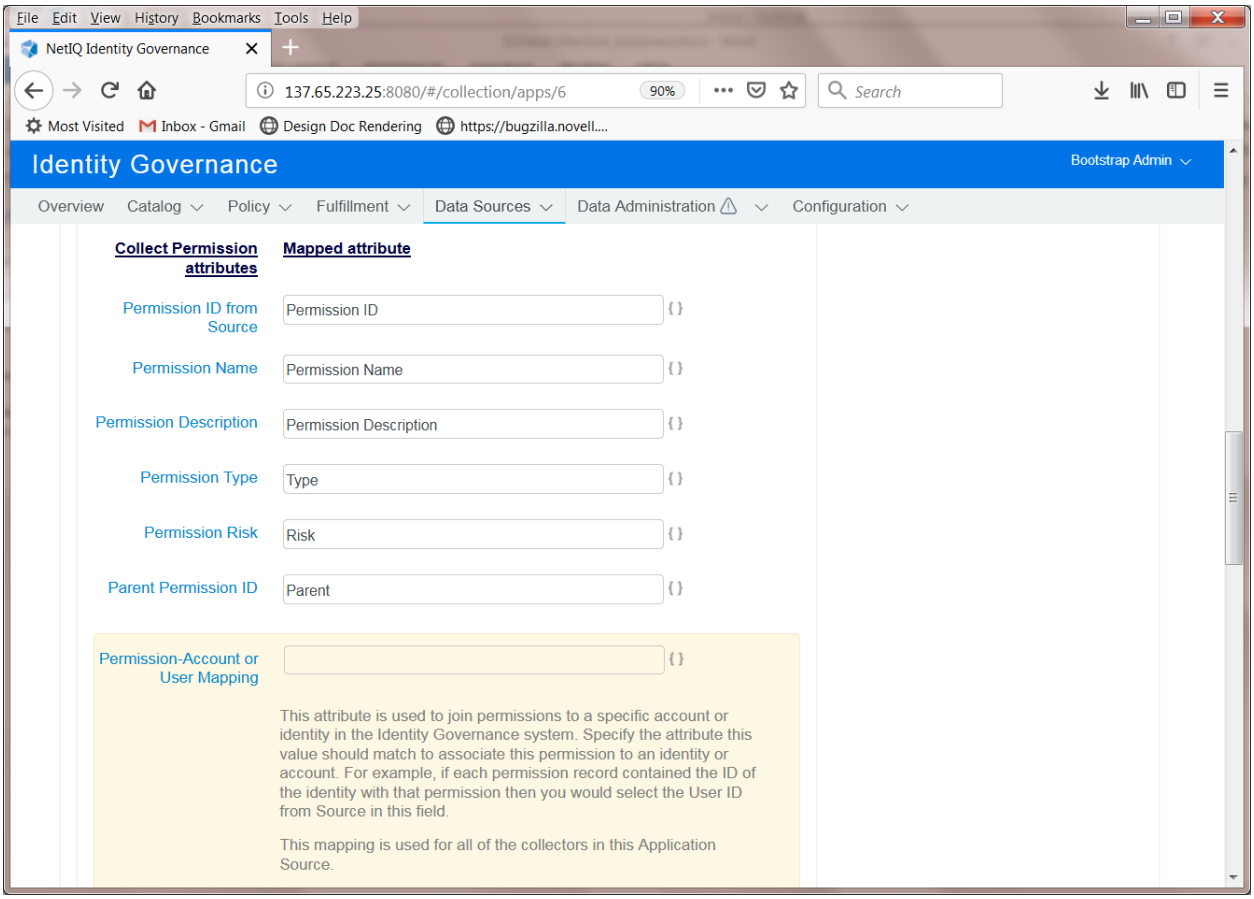
NOTE: ***The source type of Application connectors does NOT have to be the same!*** For example, there may be an attribute on User accounts in Active Directory (eg: myAppRoles) that is used to assign permissions for an enterprise application. Although the Accounts are collected from Active Directory, the Permission catalog may be collected from a JDBC database, a CSV file, or some other source.

For the first Permission collector example, we will utilize the Money Honey Financials application source from the chapter on Account collectors. The permission catalog for this source is maintained in a CSV file:

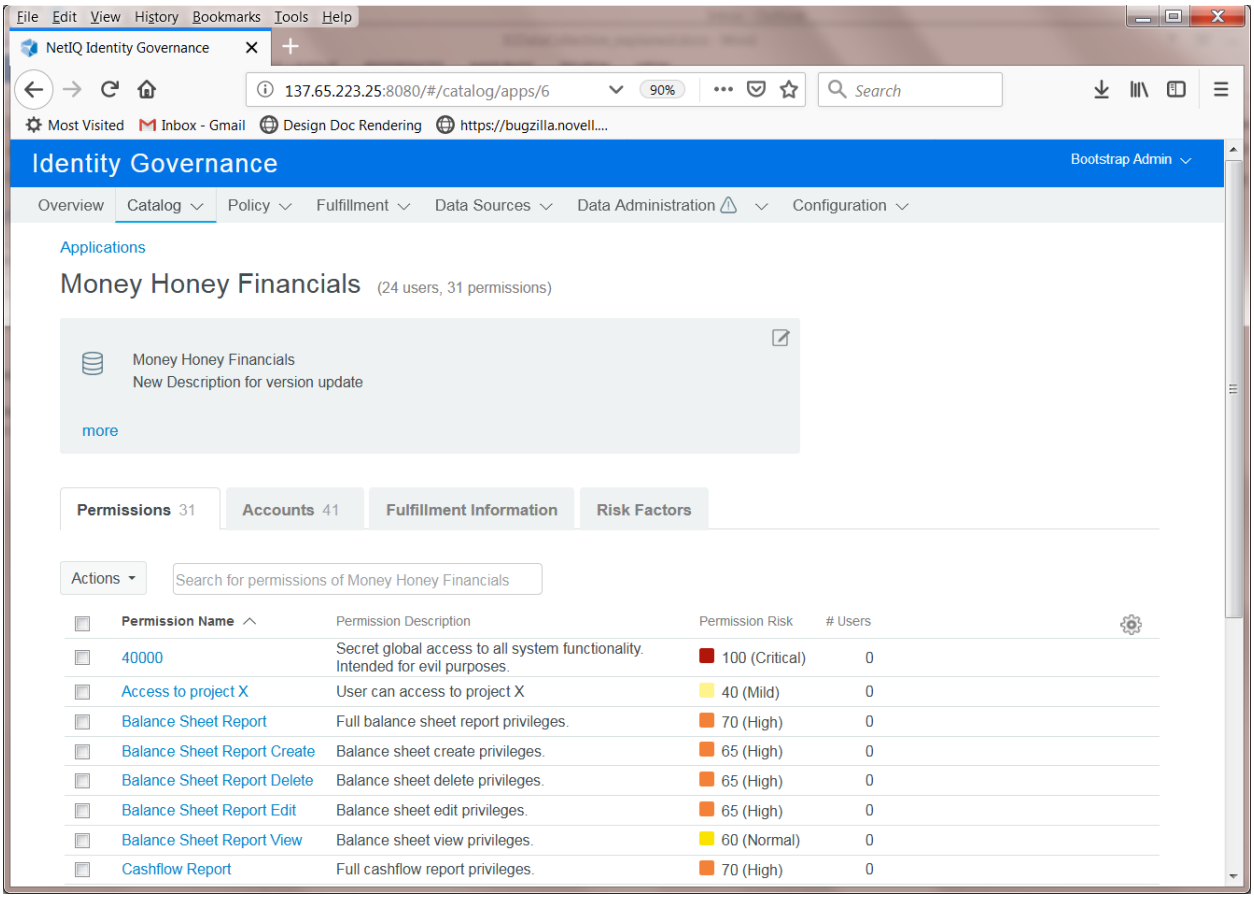


This is a very well curated source of permission data that contains a great deal of descriptive information about each possible permission that may be assigned to a Money Honey account holder. Additionally, it contains a link to a Parent permission that may be used to build a permission hierarchy in Identity Governance.

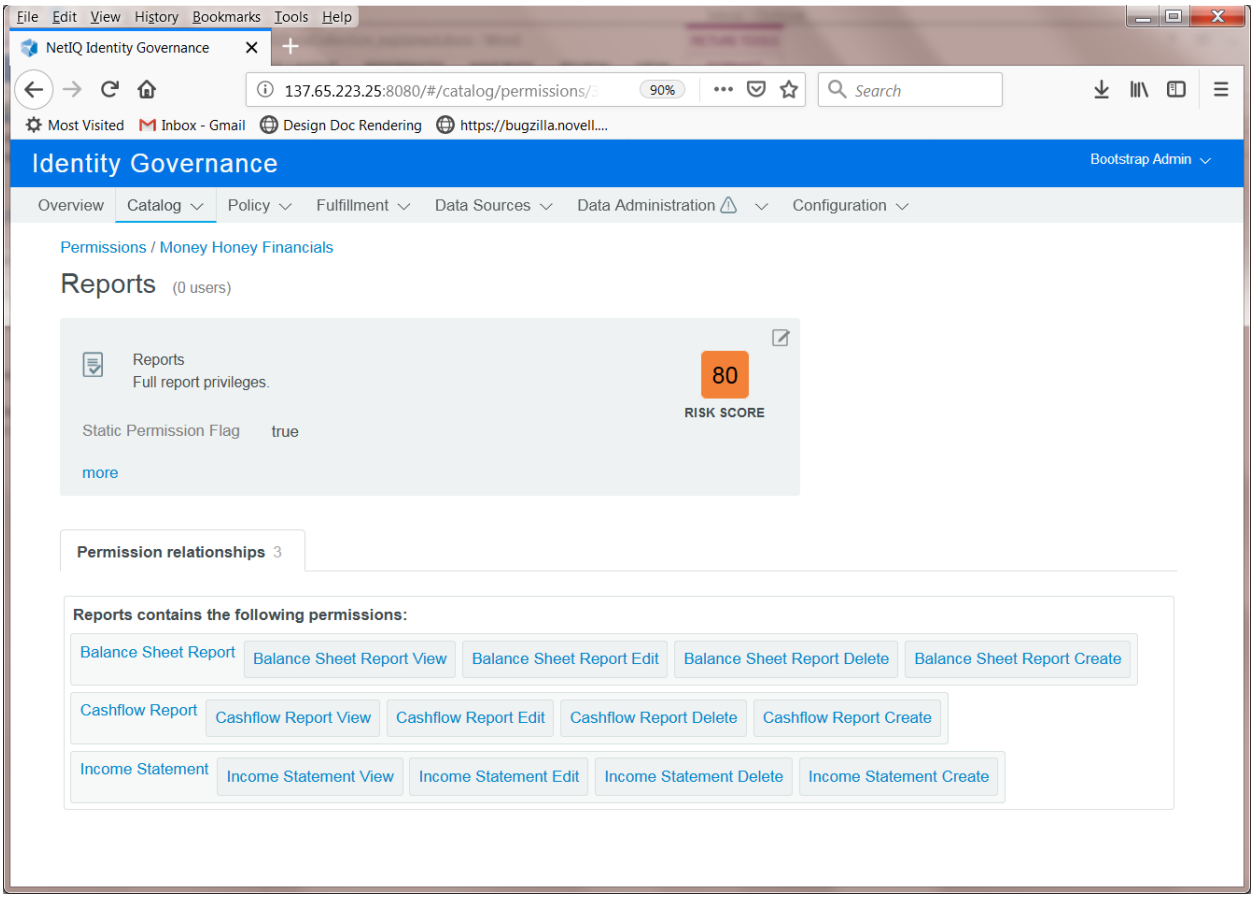
A Permission collector for this permission type would look like this:



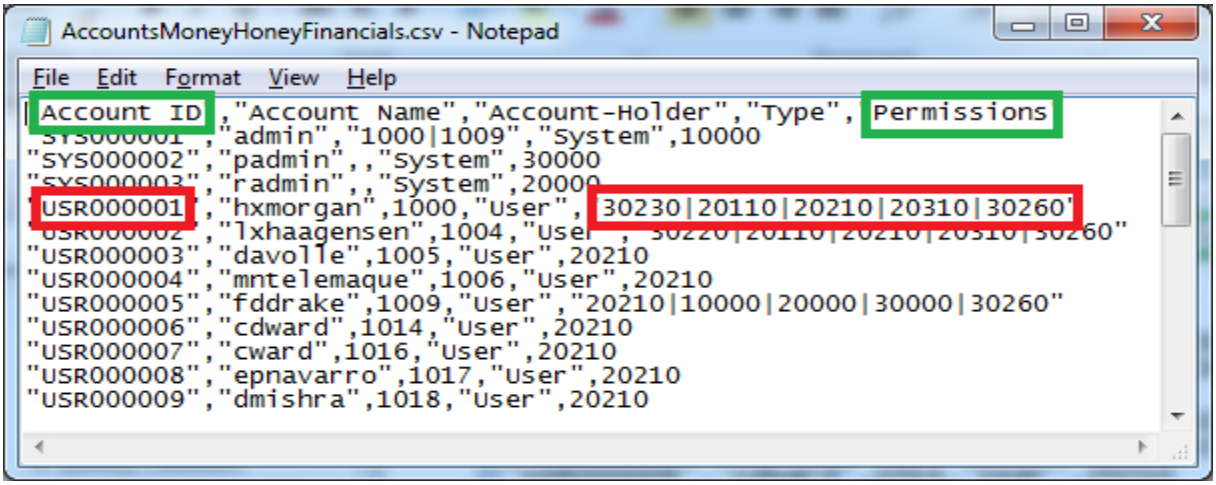
In this collector, the relevant columns from the CSV were mapped. After collection and publication, the set of collected permissions can be seen under the Money Honey Financials application view in the Catalog:



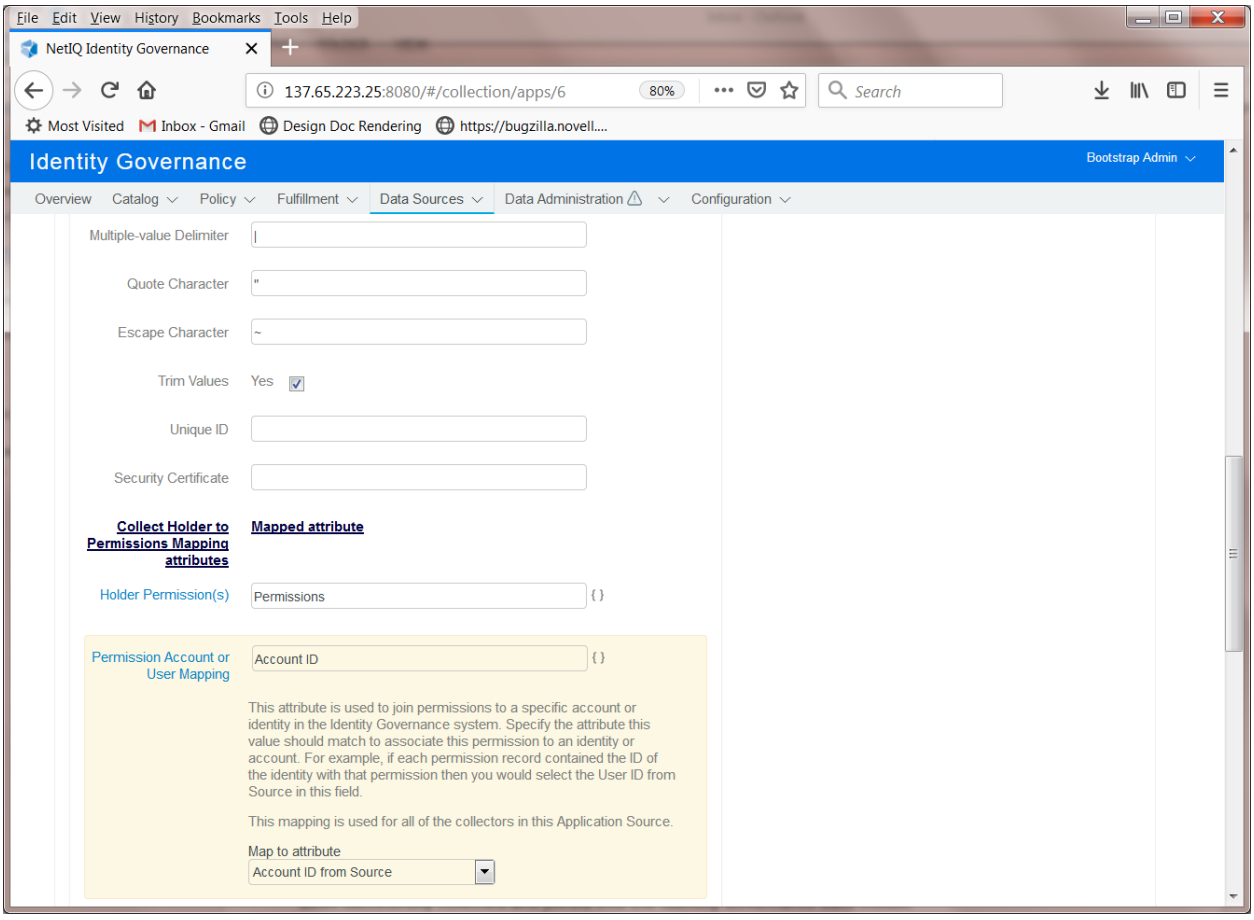
Since a child-to-parent permission relationship was also collected, if we drill down on a top-level permission we will be able to see the collected relationships in the details:



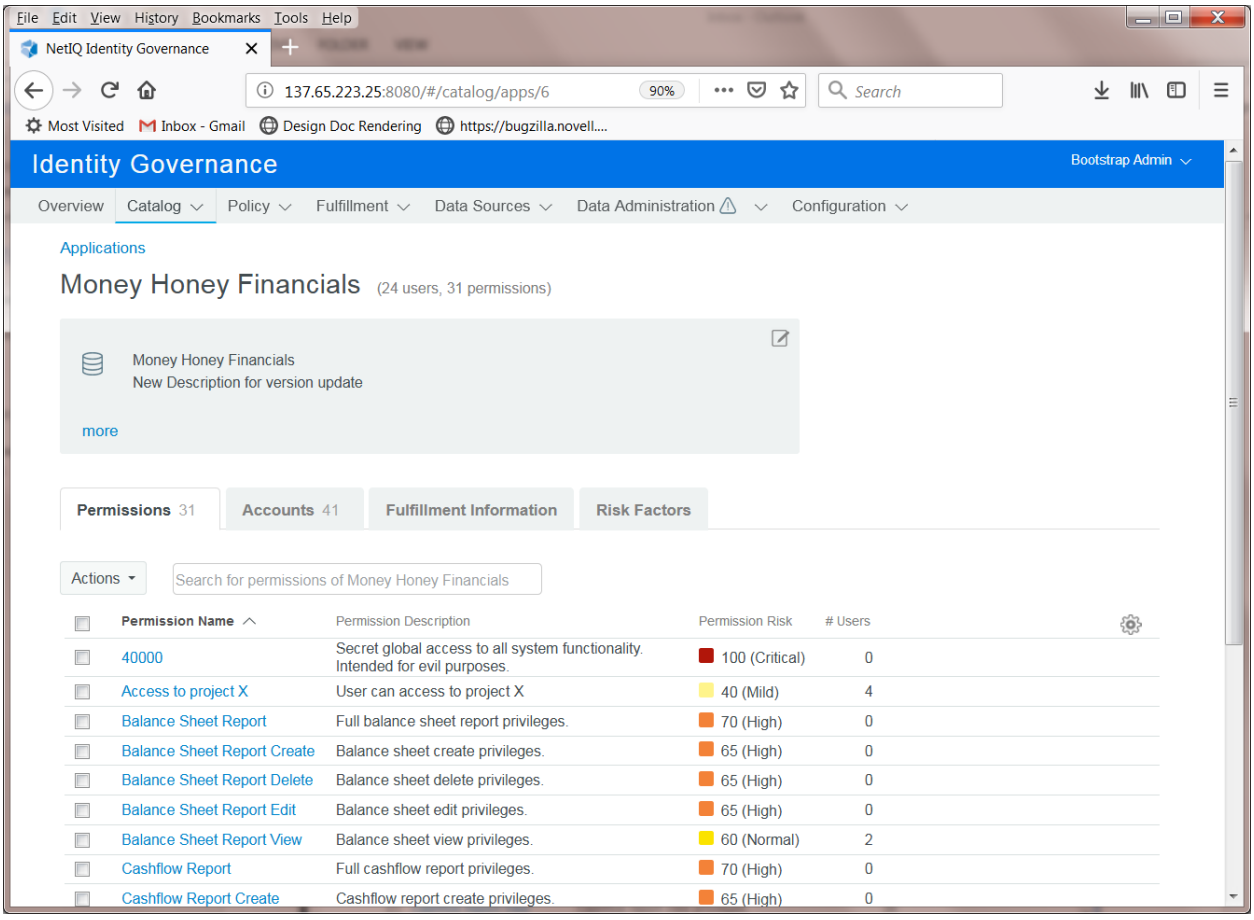
Now that we have collected Money Honey Financials Accounts and Permissions, we need to build the relationship between them that will complete our picture of the application. If we re-examine the Account information CSV for this application, we see that the relationship between the account holder and their permissions is maintained in that file:



The mapping attributes are “Account ID” (holder) and “Permissions” (permission). The example record shown in red indicates how multiple values may be assigned on a single row if this is desired. Since the relationship goes from the holder to the permissions in a 1:n fashion, we will enable and configure a *Collect Holder to Permissions Mapping* view on our Permission collector for our application, and also ensure we use the “|” multi-value delimiter in the source configuration. Since the permissions will be joined to the Account ID, not directly to the Identity, we select the “Account ID from Source” join attribute for the holder:



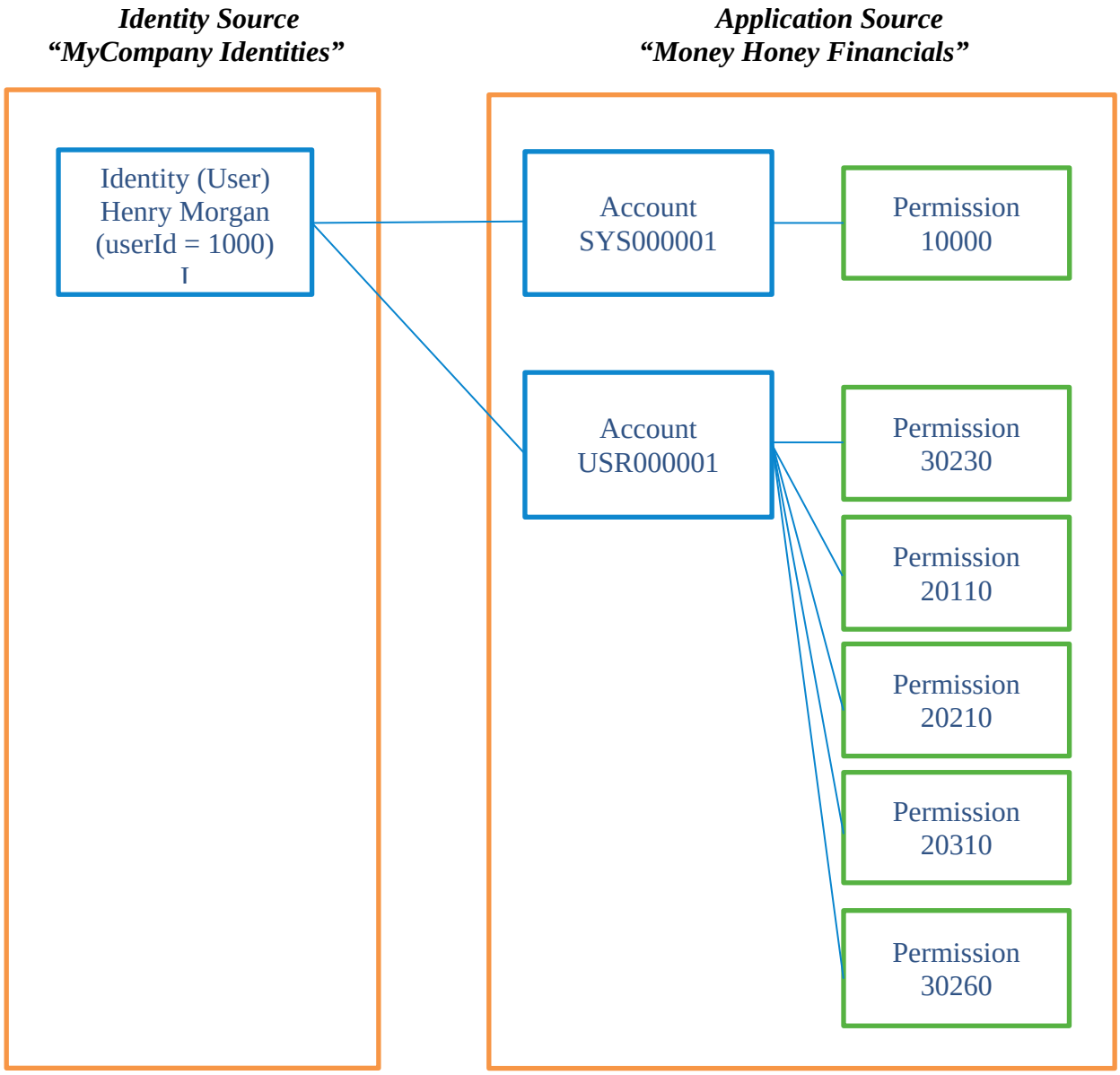
Once we collect and publish the application, we can revisit the catalog and see that all of the different entities have been successfully collected and joined into the Identity Governance data model:



The Money Honey Financials Account collector collected account information and mapped the accounts to the MyCompany Identities, and the Money Honey Financials Permission collector collected the permission catalog and mapped the assigned permissions to the Accounts. The Catalog always presents this information as a mapping between the Identities and the Permissions.

If we examine one of the more complex assignments for this application, we can see the power of the mapping capabilities of Identity Governance. The Account USR000001 (hmorgan) is linked to Identity 1000 (Henry X Morgan). The USR000001 account is directly assigned 5 permissions (30230, 20110, 20210, 20310 and 30260). However, the Henry X Morgan Identity is also assigned to the SYS000001 Account (admin), which is directly assigned to Permission 10000. During Publication of this application, Identity Governance will associate both accounts USR000001 and SYS000001 to Identity 1000, and join the permissions assigned to those account to the Identity.

An illustration for this example would look like this:



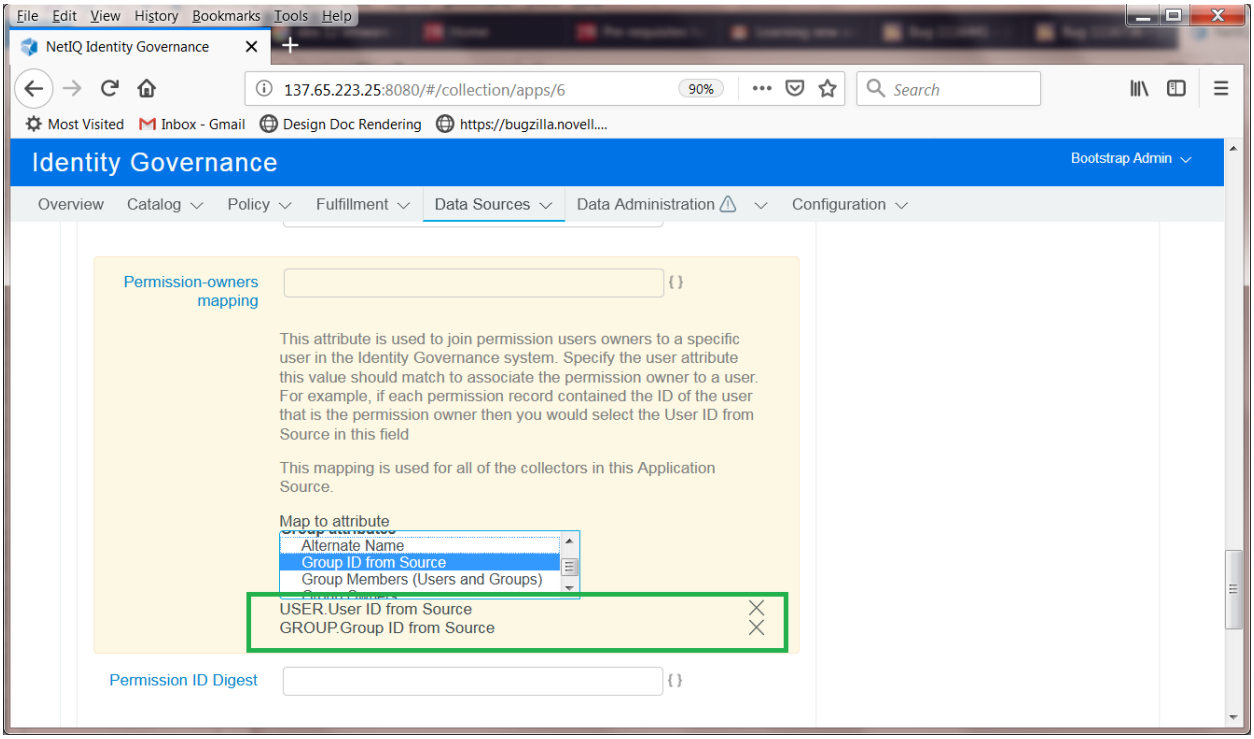
NOTE: The Permission assignments of collected accounts that do not have mappings to Identities are not displayed in the Permission view of the Catalog.

Permission Owner Mapping

In many customer scenarios, there is a desire to assign an owner to a Permission. The owner(s) may be needed to review access to the permission, provide a business-friendly description or category, or evaluate the risk. As with account custodians, some Application sources (eg: eDirectory and Active Directory) support the concept of an owner which can be collected directly from the system. If the data can be collected, the *Permission-owners mapping* attribute is mapped in the Permission collector configuration and is mapped to an Identity or Account Holder in a manner similar to the *Permission-Account or User Mapping* attribute described previously. If the data cannot be collected, or if collection of the source value is not desirable, one or more custodians may be added by editing the permission and selecting Identities or Groups that exist in the catalog.

There is an additional feature for the *Permission-owners mapping* that is significantly different from all other mappings in the Identity Governance mapping functionality – It is possible to map Identities AND Groups as permission owners. In the configuration UI, this difference is seen in the default mapping of both *User ID from Source* and *Group ID from Source*. Either, or both, of these mappings may be removed by clicking on the 'X' next to the mapping.

Only one mapping attribute may be selected from *each* of the User and Group attributes in the mapping selector. You must hold the <Ctrl> key down and then click on your choice to make a selection for both User and Group attributes.

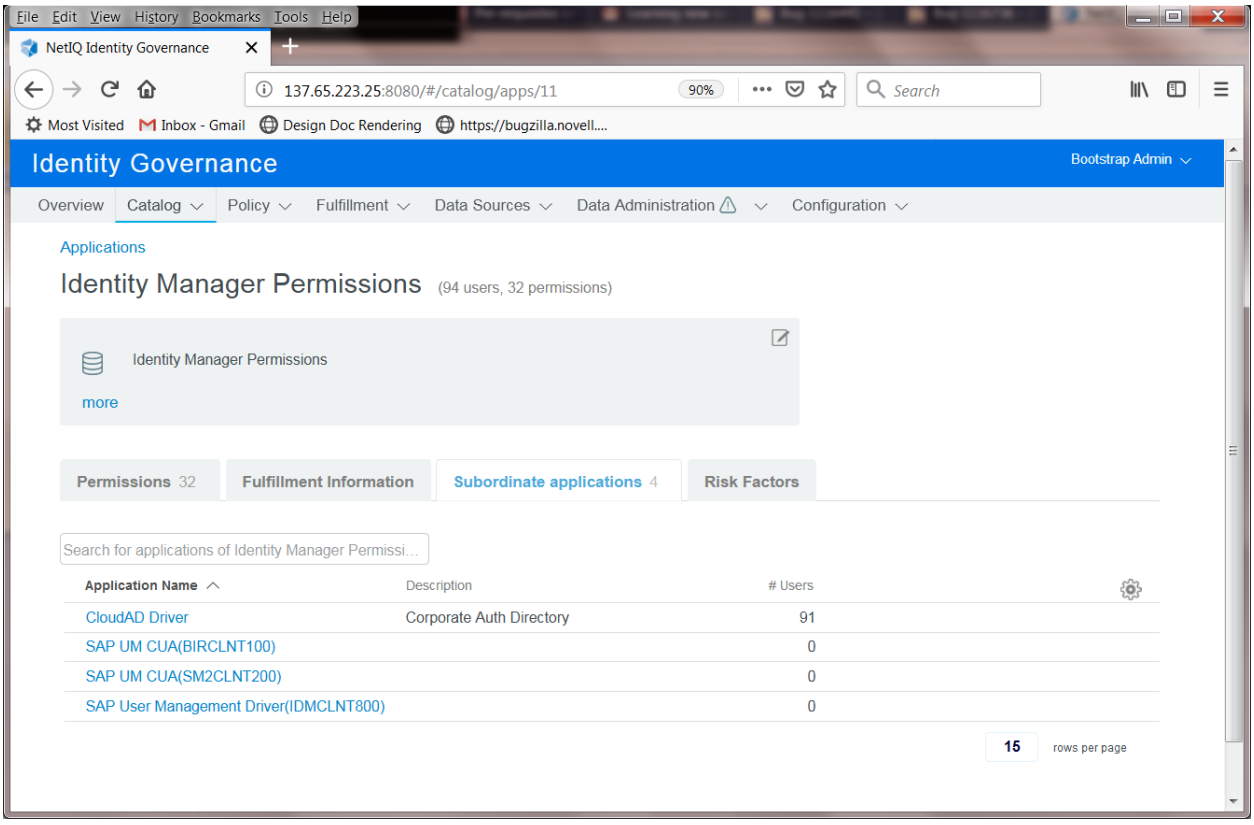


Identity Manager AE Permission Collector

Identity Manager Accounts

The Identity Manager AE Permission collector is a unique type of Application Source collector. In addition to creating the base Identity Manager application in Identity Governance, it also automatically generates subordinate applications that represent IDM Drivers that support IDM Entitlements. Any User record collected from the IDM application that is associated with a subordinate application (via the DirXML-Association attribute on the User) will also receive an Account assignment for that application and be automatically mapped to the IDM User. There is no other Application source Permission collector that provides automatic generation of subordinate applications or accounts. Due to the complexity of the relationships managed by this collector, the template intentionally blocks customization to ensure the integrity of the data collections.

The screen shot below shows the collected Identity Manager Permissions application and associated driver subordinate Applications:



In this graphic, you see that the base application is the ***Identity Manager Permissions*** application created by the Identity Governance Data Administrator. There are 4 subordinate applications automatically created due to the presence of these drivers which support entitlements:

- CloudAD Driver
- SAP UM CUA (logical system BIRCLNT100)
- SAP UM CUA (logical system SM2CLNT200)
- SAP User Management Driver (logical system IDMCLNT800)

Legal Notices

For information about legal notices, trademarks, disclaimers, warranties, export and other use restrictions, U.S. Government rights, patent policy, and FIPS compliance, see http://www.microfocus.com/about/legal/.