Creating Identity Cubes

In this lesson, we’ll discuss populating IdentityIQ with users and applications.

**First**, we’ll start with an overview of Identity Cubes and how they’re created.

We’ll talk about how we configure authoritative applications, from which we’ll read in the accounts that we want to create new Identity Cubes in IdentityIQ, and then we’ll see that applied through an aggregation task.

A picture containing text, screenshot, font

Description automatically generated

A close-up of a logo

Description automatically generated with medium confidence

Let’s get going with our Identity Cube discussion.

Identity Cubes represent our users - one cube for each access-holding entity or person; a unique identity stored in the repository.

We call it a cube because we try to give a multi-dimensional view of the person – not just what accounts they have but also details about any policy violations, what roles they have, when they are were last certified, their IdentityIQ user rights, and more.

A log of information is stored in the Identity Cube.

There are 4 different ways that data can be added to a cube:

When we aggregate, information is discovered: we are reading in data from some other system, that is, discovering it in that environment, and posting it to the cube.

It can be requested: a user logs in to Lifecycle Manager and requests access to an application or an entitlement; that request will be provisioned, and then that requested information, from that provisioning, will be posted to the cube as access they now have.

It can be assigned: we haven’t gotten into too much detail on this yet, but we can do automatic assignment based on, for example, attributes.

So we might have a role that encapsulates some set of attributes, like a “bank teller role”.

I can have a rule associated with that role that says, if their job title is “bank teller” we will assign them this role.

We don’t even have to request it if that rule exists on that role.

Lastly, data can be calculated.

If you stop the playback and think about it, you already know some of the attributes that are calculated.

Did you say risk score and policy violations? As long as we have our models defined for these, you’re correct.

A screenshot of a computer

Description automatically generated with medium confidence

How do we create Identity Cubes? We have 2 mechanisms.

The primary one is automatically, thru account aggregation – aggregating from the system of record.

This is the source we want to cubes to be created from; those will be the authoritative cubes.

They’ll contain all the data about the user necessary for access management.

We can also manually create Identity Cubes using Lifecycle Manager.

This can be done via the Create Identity Quicklink.

In this case, whoever is creating the cube enters the attributes for the identity manually, as opposed to reading them from a system.

We’ll talk more about this later, too.

A picture containing text, screenshot, font, number

Description automatically generated

Here we see Adam Kennedy’s Identity Cube.

There are 9 tabs, each representing a different perspective about the user.

The default tab when we first view an identity shows us the identity attributes.

The first 5 are standard attributes, and we also see an extended attribute that was added to the page: Department.

These values are typically sourced from account data read from the authoritative application.

So we’re reading in from HR, Saying “what is the user’s first name, last name, email address”, and we’re populating these identity attributes from the account data.

Note that you don’t have to use the authoritative resource – you can also supply the value of the identity attribute using account details from other applications, or through a rule.

You can see details for Adam Kennedy’s accounts on his Application Accounts tab.

There, we see four accounts that Adam has on other systems in this enviroument.

We’ve read them by aggregating from these various applications.

We can expand an account and see the account attributes.

The account attributes are used to populate the identity attributes we just discussed.

A screenshot of a computer

Description automatically generated with medium confidence

A close-up of a logo

Description automatically generated with low confidence

Let’s get into the details of how you can define authoritative applications in IdentityIQ.

As you onboard each source system into IdentityIQ to become part of your governance program, you‘ll define how communication to that system is accomplished and the data required for your governance processes.

Defining applications starts with specifying the name of the application, and whoever the owner is.

Right now we who the administrator as being the owner.

But as you start building out your system, a best practice is to have a workgroup as the owner – We’ll talk about workgroups in the next lesson.

After selecting an owner, we define the application type – that tells IdentityIQ which connector to use to access the data.

For this one I’ve selected delimited file.

The connectors available in IdentityIQ are all documented, But this drop-down will show all your available connectors at a quick glance.

After selecting the connector, the configuration screens will change to include the appropriate settings, which are different for different connectors because they specify how to establish communications between IdentityIQ and that system or data source.

So you might need to provide a file location, a database URL, a service account, host name or IP, or other values, depending on the type of system you’re onboarding.

Over here, this simple checkbox is what makes this an authoritative application.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A picture containing text, font, line, design

Description automatically generated

Checking this box means that IdentityIQ creates authoritative Identity Cubes for each account in this application.

Systems which contain personnel records should be marked as authoritative so we create identity cubes for them.

For any application, part of what we have to define is what data we want to read from the application.

We have to read account data, but which attributes do we read to obtain that account data?

The application may have dozens or hundreds of attributes that are not relevant to identity management.

We only need to read in the data that is particularly important for identity management, for managing those accounts.

So the purpose for our account schema is to define how to identify accounts and which attributes we want to bring into IdentityIQ.

In our example, since we don’t need social security numbers to manage users access, we’re not including ssn in the schema Often, depending on the connector, this information is predefined and filled out.

So for the AD connector, for example, if you were to look at the schema, the attributes are already listed.

You can add more attributes, or delete those you don’t use, but you often don’t have to fill this in.

Out delimited file connector and JDBC connector are very open ended; We don’t know a lot about your delimited files – so for those you’d have to fill this in.

Bottom line, however it’s populated, IdentityIQ requires an account schema for each application.

A screenshot of a computer

Description automatically generated with low confidence

A close-up of a logo

Description automatically generated with medium confidence

Now we’ll take our first look at the Rapids Setup features and how they can aid in your configuration and Implementation Process.

Once you’ve setup the application connectivity and schema details, the Rapid Setup feature can be used for defining data-centric configurations.

The options in Rapid Setup include aggregation-focused settings that tell IdentityIQ how to use and interpret the account data in the correlation process and for categorizing the account or identity.

A screenshot of a computer

Description automatically generated

Identity Correlation is how we define which identity object each account belongs to.

For your authoritative applications, that is usually pretty straightforward.

It should match the logic used for setting the identity name in the first place.

This Rapid Setup option is one of the several ways you can define correlation logic, which we will explore in more detail in a later lesson, which we talk about aggregating from other systems in your environment.

Manager Correlation is how IdentityIQ should match identities to each other to match your reporting hierarchy.

This identity-to-identity relationship is used in operations like manager certifications and manager approval, or to let managers make certain types of access requests for their team members.

Both of these configurations can alternatively be specified in the Application Definition itself, But with Rapid Setup, you can more easily divide access and responsibilities across your team.

The connectivity configurations can still be done by your technical implementers while these data-centric settings get assigned to a business analyst to configure.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

The other configurations here help IdentityIQ interpret account attributes to categorize the account or identity.

The Disable and Locked Account configurations indicate how to reflect the state of the account.

For many of the systems IdentityIQ connects to, the attributes which define these status are well-defined and already known by the connector.

In that case, there’s no need to manually define them and they won’t appear on this page at all.

This applies to any application that is configured to support changing of these attributes for IdentityIQ.

When IdentityIQ doesn’t control these, though, this lets you specify how to recognize locked and disabled accounts when you are reading data form these systems.

The other two categorization options in Rapid Setup let you specify how IdentityIQ can recognize accounts as mapping to special type of identities – namely, Service or Robotic Process (Bot) Identities.

These will be used to set the identity type of the identity created for, or connected to, these accounts.

A screenshot of a computer

Description automatically generated

A close-up of a sign

Description automatically generated with low confidence

Once you’ve defined these configurations for you authoritative applications, you’re ready to read that data into IdentityIQ through an Account Aggregation.

For each application, you will configure an aggregation task to read in its account data.

You do this based on the Account Aggregation template task.

The aggregation task template includes many configuration parameters, but only one is required: the application this task will aggregate.

The most commonly selected additional option is Detect Deleted Accounts.

This tells IdentityIQ that if an account no longer exists on the application’s aggregated data, it should also be deleted from the identity cube.

This makes sense, as we want IdentityIQ to match the real-world situation.

If the account doesn’t exist on the application, it shouldn’t exist in IdentityIQ!

The aggregation task will then use all of the details you’ve configured in the application definition to access and read the account data form the application.

**A close-up of a document

Description automatically generated with low confidence**

Now let’s walk through the Identity Cube creation process.

We start with our authoritative resource – this is the application where, for each user account, we want to create an identity cube within IdentityIQ.

The application has account information needed by IdentityIQ for identity governance and access management.

Here, we’re looking at the account attributes for Megan Cahill.

In IdentityIQ we create an application definition that represents the authoritative resource – what account information to read, and how to access it – the connector.

If we need to, we can use rules to add our own business logic.

Using the Rapid Setup functionality, we can add logic for classifying accounts and identities.

This application includes an account Schema, which describes that data fields of the application that we want to read into IdentityIQ.

These are our account attributes. Notice that we are not reading in all fields; for example, in this case, we’re not reading in United States social security numbers.

We then create and run an aggregation task that reads the accounts from the authoritative resource as specified in the application definition.

The task references the details from the application definition.

It uses the connector to read the accounts one by one, and process them through any rules that have been defined starting with connector rules and on to application rules.

Finally, IdentityIQ creates the new authoritative cube, or updates an existing cube.

The user’s account attributes are stored on the Application Accounts tab of the cube.

A screenshot of a software creation process

Description automatically generated with medium confidence

A screenshot of a application account

Description automatically generated

Here we see Admin Kennedy’s Identity Cube again.

When we aggregate, we read account information and attributes from the applications into the Application Accounts area of the identity cube.

When we re-aggregate, we update the account data held in IdentityIQ.

So far, we’ve only aggregated from the authoritative HR Employees system, but later, we’ll see more accounts added here as we aggregate from other applications.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A white circle with blue text

Description automatically generated with medium confidence

Next, you’ll define your authoritative applications within IdentityIQ.

In the first exercise, you configured IdentityIQ.

Now, still in Section 1, you ‘ll continue with Exercise 2. In this exercise, you will onboard the systems of record.

A picture containing text, screenshot, font, design

Description automatically generated

In this exercise, you will onboard the systems of record.

We have two delimited files provided to us by the data owners.

One contains the employee accounts, and the other contains the contractor accounts.

These are our authoritative sources for our exercises.

In this exercise, you will define the applications and the aggregation tasks, load the authoritative data and populate the Identity Cubes.

You’ll also add a creation rule to set default passwords for our users to authenticate into IdentityIQ.

A picture containing text, screenshot, font, line

Description automatically generated