Back-End Application Overview

The powerful, data-centric back end of the Palantir Platform provides system administrators and integrators with the means to model business data as objects, import data into Palantir, and export data from it. This section discusses the following key features of the back end:

- Dynamic Ontology
- Support for Multiple Data Sources
- Federated Search Using Raptor
- Change Control with the Revisioning Database
- Access Control on a Granular Level
- Accessible and Extensible Open Platform

Dynamic Ontology

A central feature of the Palantir back end is the dynamic ontology. An ***ontology*** is a way of naming and arranging objects, the relationships among them, and their attributes in such a way that the resulting structures make sense to your organization's workers. An ontology categorizes what appears in your world, representing real-world data in a familiar way without requiring understanding of, or even interaction with, the computer systems that maintain that data.

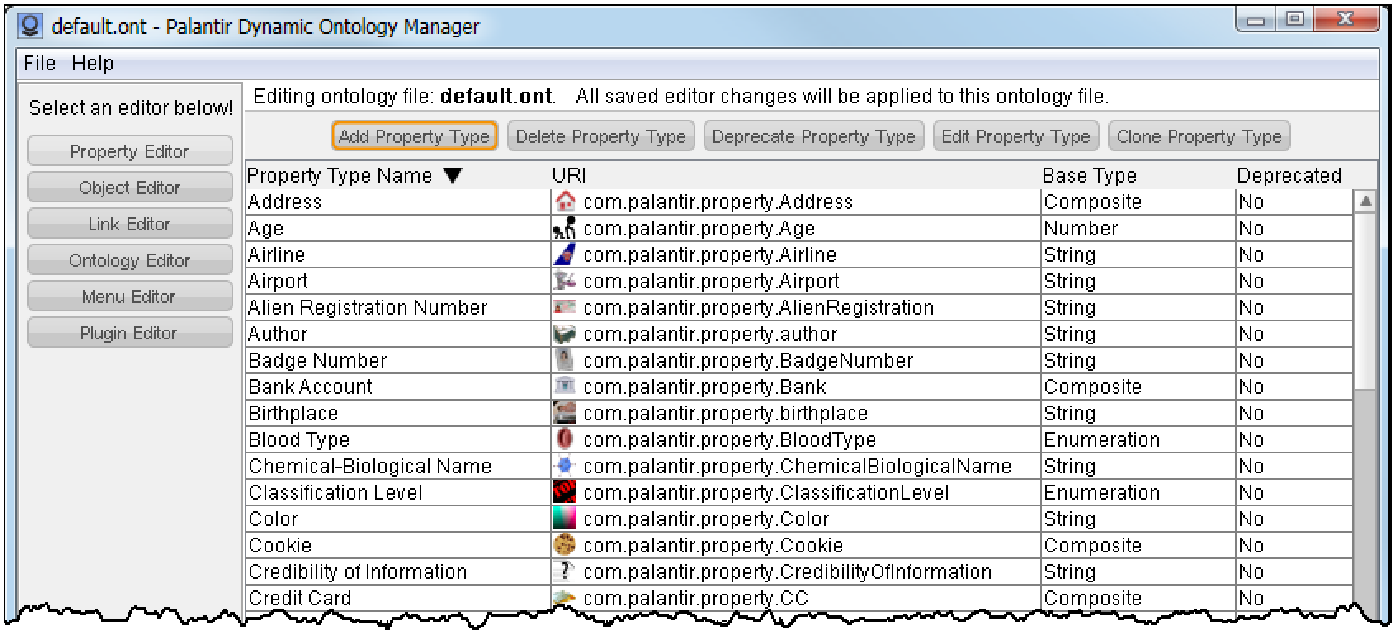
Your ontology is how your organization sees the world. The object model is a translation of that world into data that a software system, in this case Palantir, can manipulate. The Palantir object model uses the following specific building blocks to describe an ontology.

Building Blocks of an Ontology's Object Model

Building Block	Description
object	Representation of a type of physical or living thing in an ontology. Examples of objects are people, places, or documents. An object can also be an event, such as a meeting. In Palantir, the base objects are `Entity`, `Document`, and `Event`.
property	Attribute or characteristic of an object. Each object type has different properties. For example, a `Person` object has an `Eye Color` property and an `Event` object has a `Start Date` property. You cannot swap those properties between the `Person` and `Event` objects, because they are specific to the object type.
link	Relationship between objects. Links describe how two objects are connected. For example, Mike Fikri might be a *friend of* Alice Watson or Alice Watson might be a *member of* the Girl Scouts.

Along with properties and links, you can also attach notes or media to an object in the Workspace. Notes are simply free-form text you can attach to an object. For example, you might attach a note to a person that says, "likes to frequent coffee bars." Media can be images, sound, or video.

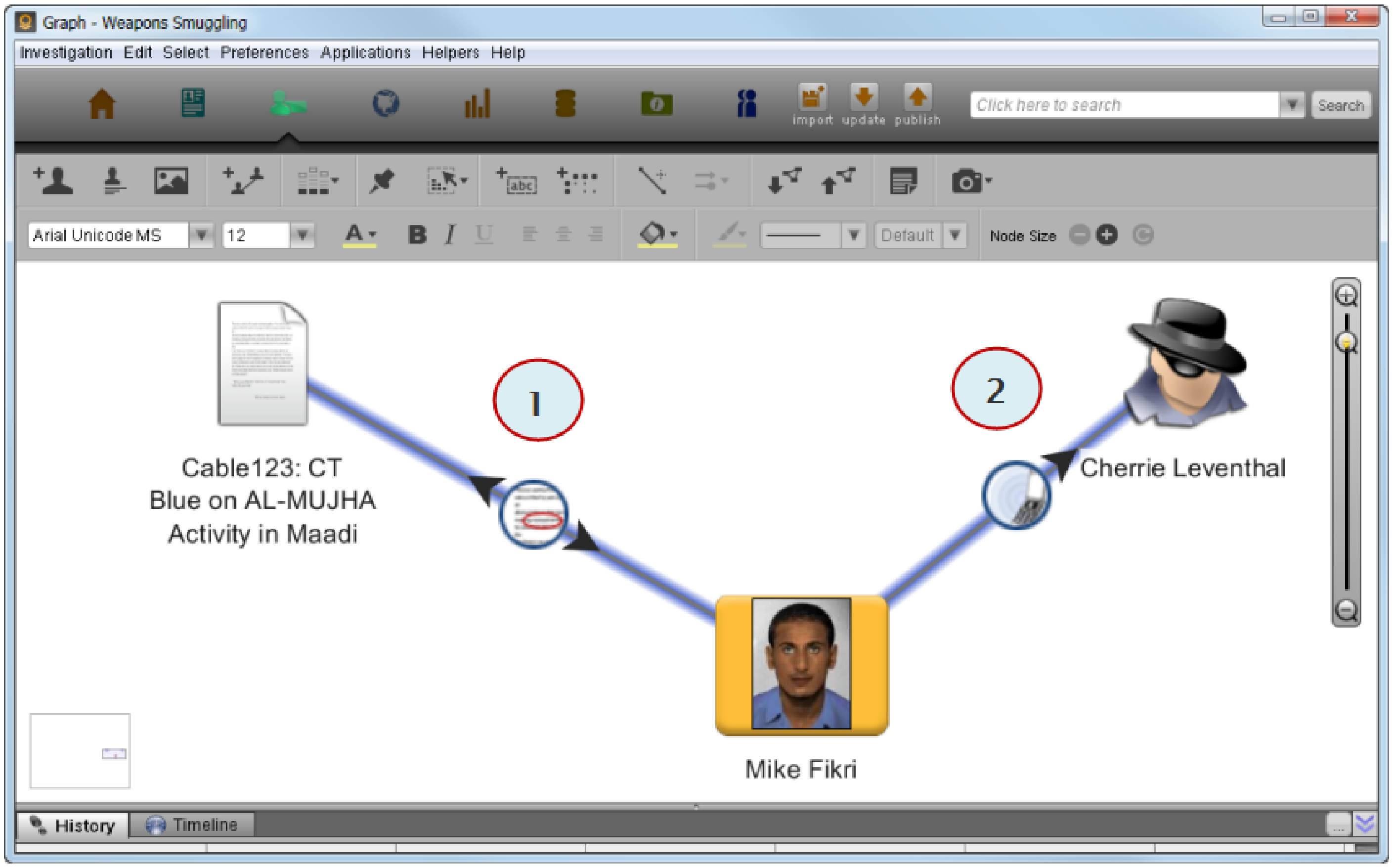
The dynamic aspect of the Palantir ontology is that each Palantir installation can have its own unique ontology, and it can evolve over time. Each Palantir installation has a default ontology. For example, a police unit would have objects such as suspect, defense attorney, arrest, and police report. A military unit might have objects such as combatant, battle, topographical map, and enemy camp. Palantir administrators can customize the default ontology or develop totally new ontologies using the Palantir Dynamic Ontology Manager.



Editing ontology file: **default.ont**. All saved editor changes will be applied to this ontology file.

Property Type Name ▼	URI	Base Type	Deprecated
Address	com.palantir.property.Address	Composite	No
Age	com.palantir.property.Age	Number	No
Airline	com.palantir.property.Airline	String	No
Airport	com.palantir.property.Airport	String	No
Alien Registration Number	com.palantir.property.AlienRegistration	String	No
Author	com.palantir.property.author	String	No
Badge Number	com.palantir.property.BadgeNumber	String	No
Bank Account	com.palantir.property.Bank	Composite	No
Birthplace	com.palantir.property.birthplace	String	No
Blood Type	com.palantir.property.BloodType	Enumeration	No
Chemical-Biological Name	com.palantir.property.ChemicalBiologicalName	String	No
Classification Level	com.palantir.property.ClassificationLevel	Enumeration	No
Color	com.palantir.property.Color	String	No
Cookie	com.palantir.property.Cookie	Composite	No
Credibility of Information	com.palantir.property.CredibilityOfInformation	String	No
Credit Card	com.palantir.property.CC	Composite	No

The objects, relationships, and links that users encounter when working in the Workspace all belong to your organization's ontology.


Graph - Weapons Smuggling
Investigation Edit Select Preferences Applications Helpers Help
import update publish
Click here to search
Search
Arial Unicode MS
12
Default
Node Size
Cable123: CT Blue on AL-MUJHA Activity in Maadi
1
2
Cherrie Leventhal
Mike Fikri
History
Timeline

Using the Palantir Dynamic Ontology Manager, a Palantir administrator can modify the ontology quickly and deploy it. After deploying the modified ontology, the changes are reflected in every Workspace and are available to every user.

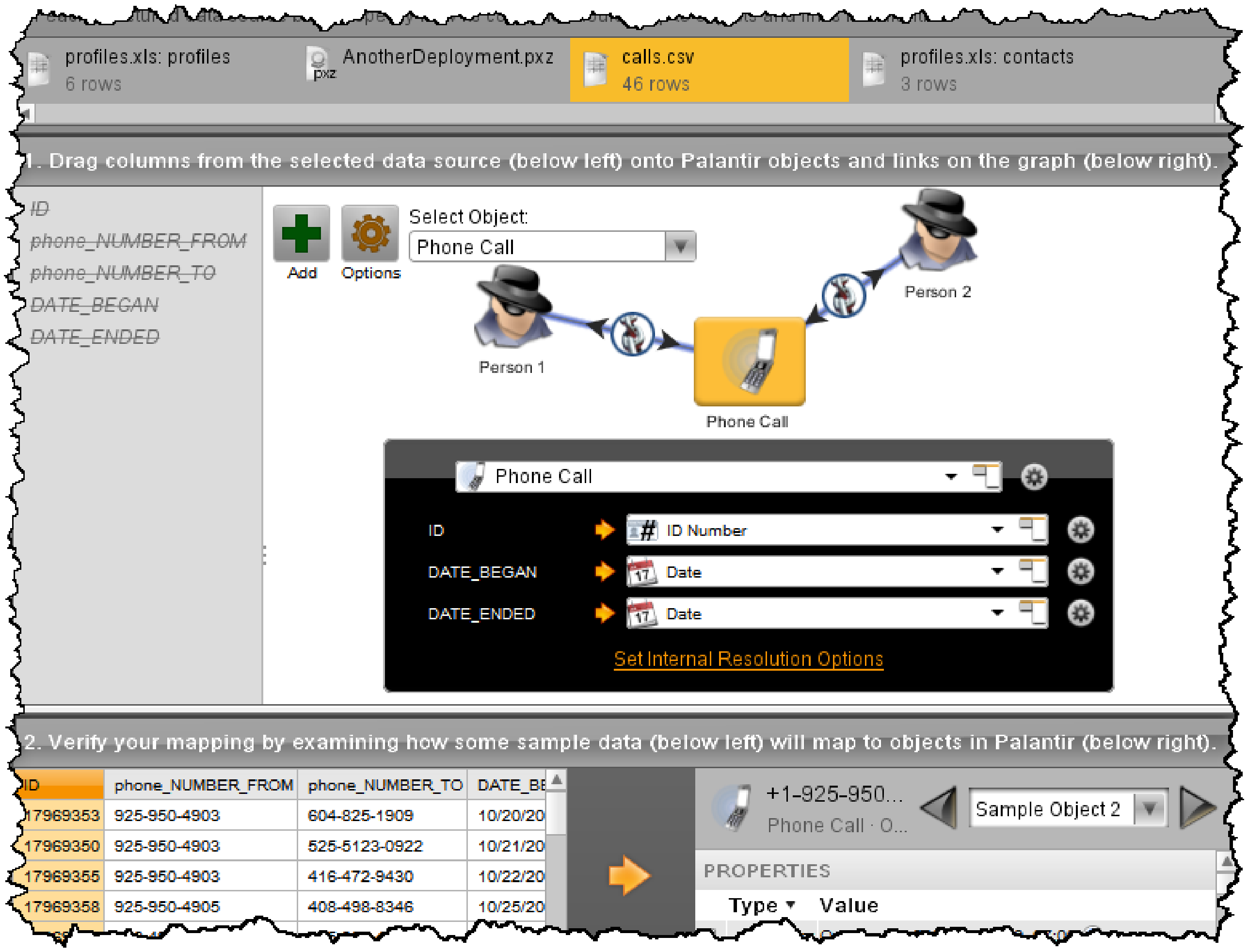
Support for Multiple Data Sources

Analysts associate their data with ***data sources***. Analysts can typically import data in small batches, such as a handful of files. Palantir administrators typically import data in bulk, such as an entire file system or database. Palantir categorizes data as originating from:

- Structured data sources, such as a database, a `.csv` file, or any tab-delimited or fixed-width file.
- Semi-structured sources, such as an email server, or forms such as a suspicious activity report (SAR) or currency transaction report (CTR).
- Unstructured data sources, typically encoded files such as PDF, sound, and image files.

Structured data sources are matched against the dynamic ontology and imported as objects (people, places, things), events, properties, and links (relationships). Semi-structured and unstructured sources are typically imported as documents. Some formats, such as email, are imported as events.

This image shows an example of using the Palantir Data Importer to import data from a structured data source.



profiles.xls: profiles 6 rows | AnotherDeployment.pxz | calls.csv 46 rows | profiles.xls: contacts 3 rows

1. Drag columns from the selected data source (below left) onto Palantir objects and links on the graph (below right).

~~ID~~
~~phone_NUMBER_FROM~~
~~phone_NUMBER_TO~~
~~DATE_BEGAN~~
~~DATE_ENDED~~

Select Object: Phone Call

Phone Call

ID	ID Number
DATE_BEGAN	Date
DATE_ENDED	Date

Set Internal Resolution Options

2. Verify your mapping by examining how some sample data (below left) will map to objects in Palantir (below right).

ID	phone_NUMBER_FROM	phone_NUMBER_TO	DATE_BE
17969353	925-950-4903	604-825-1909	10/20/20
17969350	925-950-4903	525-5123-0922	10/21/20
17969355	925-950-4903	416-472-9430	10/22/20
17969358	925-950-4905	408-498-8346	10/25/20

+1-925-950...
Phone Call · O...
Sample Object 2

PROPERTIES

Type	Value

Your installation might also include the Palantir Extraction Integration Server, in conjunction with a third-party entity extraction tool. This server supports document imports. The server takes output from a third-party extraction tool and imports it into Palantir. The server extracts objects and their properties from these preprocessed document data sources and creates corresponding Palantir objects.

Palantir provides an Entity Extractor SDK that includes a common interface to all major extractors. You can access features through a command-line interface or a simple object access (SOA) web service.

Federated Search Using Raptor

Federated search allows you to search multiple sources with a single query from a single user interface. The Raptor Server is the component of Palantir that supports federated searches on external data. Using Raptor, you identify data sources outside of Palantir, such as an archive of documents. Raptor indexes these archived documents. The same access-control features you use with repository data apply to the data searchable through Raptor.

Analysts use the SearchAround feature to perform federated searches. The search runs transparently against all data sources available to the system, including the ***Data Repository*** and the raptored data.

Raptor is an optional feature of Palantir. You need not install it, but many customers do.

Change Control with the Revisioning Database

As user add, edit, and remove data, the system keeps track of changes to the data. Palantir stores a record of these changes in the Revisioning Database, which:

- Stores an online history of each change to an object.

  Analysts can review their changes over time by viewing a history of changes they made in their investigations.
- Maintains separate "investigational spaces" or realms.

  A single ***data repository***, or base realm, forms a complete view of all the objects available in the system. Each investigation has its own virtual view of these objects in an ***investigative realm***. Analysts use investigative realms to explore competing hypotheses about the data and to edit that data without affecting other analysts' investigations.

- Shares changes granularly.

  Changes developed in one investigational realm must be shared with other analysts. This feature of the Revisioning Database allows the system to share changes on a per-object basis. Analysts can import changes from the larger data repository into their investigations by choosing to update an entity (object). Alternatively, an analyst can publish changes made in their investigation's entities to the data repository.

Access Control on a Granular Level

You can control access to data in these ways:

- According to user and group definitions, similar to the way Linux controls user and group permissions.
- According to classification markings. You can also use such classification markings simply to label data in the system.

You assign permissions to a user or group of users to form an ***access control list (ACL)***. Permissions can be one of the following values, in increasing order:

`none`	Users have no access at all to the data.
`discovery`	Users can discover through a search that data exists in an investigation, but they cannot read, change, or delete the data.
`read`	Users can discover and read data, but not change or delete it. Users cannot modify or create tags or notes.
`write`	Users can discover, read, change, and delete data, as well as create and modify tags and notes.
`owner`	Users can discover, read, change, and delete data, create and modify tags and notes, as well as change the permissions of the associated data source.

A higher permission level implies the lower levels. For example, giving a user `write` permissions automatically gives them `read` and `discovery` as well. At least one user in the group must have owner permissions.

Access control in Palantir is configured for individual data sources. Users have access to data sources, not to individual properties or sets of properties. In Palantir, sensitive data is located on an object's properties. There is no data on an object itself.

The sensitive data on an object's properties can come from multiple data sources. For example, the Mike Fikri person entity might have a phone number that comes from one data source and an email address from another. A data source record associates each object property back to a data source.

A particular user's access to object data in Palantir is determined by the union of his or her permissions granted by the data source records underlying that data. For example, looking at Mike Fikri, a user might be able to see his phone number and not his email address, if the two pieces of data come from different data sources with different ACLs.

This is a high level introduction to the Palantir access control model. For an in-depth discussion of access control on the Palantir Platform, see the Palantir Access Control video.

Accessible and Extensible Open Platform

Palantir is an open platform. It has a rich set of APIs that you can use to customize Palantir features and to integrate them with third-party software. The following list identifies some features that you can work with in this manner:

- Authentication

  Palantir works with your existing authentication and authorization sources. For example, you can use your existing LDAP services with Palantir or make use of public key credentials. You can also have multiple authentication sources. The Palantir authentication web service provides a common interface to these multiple sources.
- XML

  Palantir XML (pXML) provides a human-readable, serialized form of the Palantir object model. A special format, DocXML, supports the specialized needs of document formats. Through this XML API, you can pull data into Palantir from other sources or translate Palantir objects for use in other parts of your infrastructure.
- Palantir Ontology APIs

  Your developers can use the Palantir ontology APIs to extend the functionality of object properties. These APIs allow you to extract, transform, validate, and load a wide range of data types into your Palantir ontology.
- Client Connection API

  Your developers can write applications using the same API used by the Palantir Workspace. The API provides abstraction for the object model, the Revisioning Database, and the access-control model. Using this simple Java API you might, for example, quickly write a web-based view application for Palantir.

Palantir Technologies