

1

Overview of Active > 360 APIs

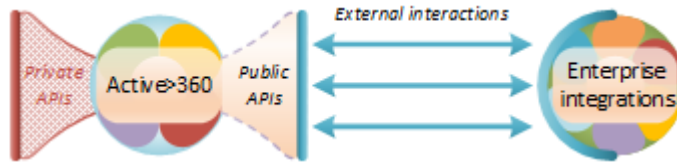
This chapter provides an overview of the Active>360 APIs. The Active>360 APIs enable Active>360 developers and administrators to customize Active>360 deployments to suit particular business goals and enterprise integration scenarios.

This chapter covers the following topics:

- [Introduction](#)
- [General features of the Active>360 APIs](#)
- [Benefits of the Active>360 APIs](#)
- [Summary of the Active>360 APIs](#)
- [Active>360 API upgrade model](#)
- [Getting started with the Active>360 APIs](#)
- [For more information](#)

Introduction

All external interactions with Active>360 must be implemented through the Active>360 public APIs.



Basic terminology

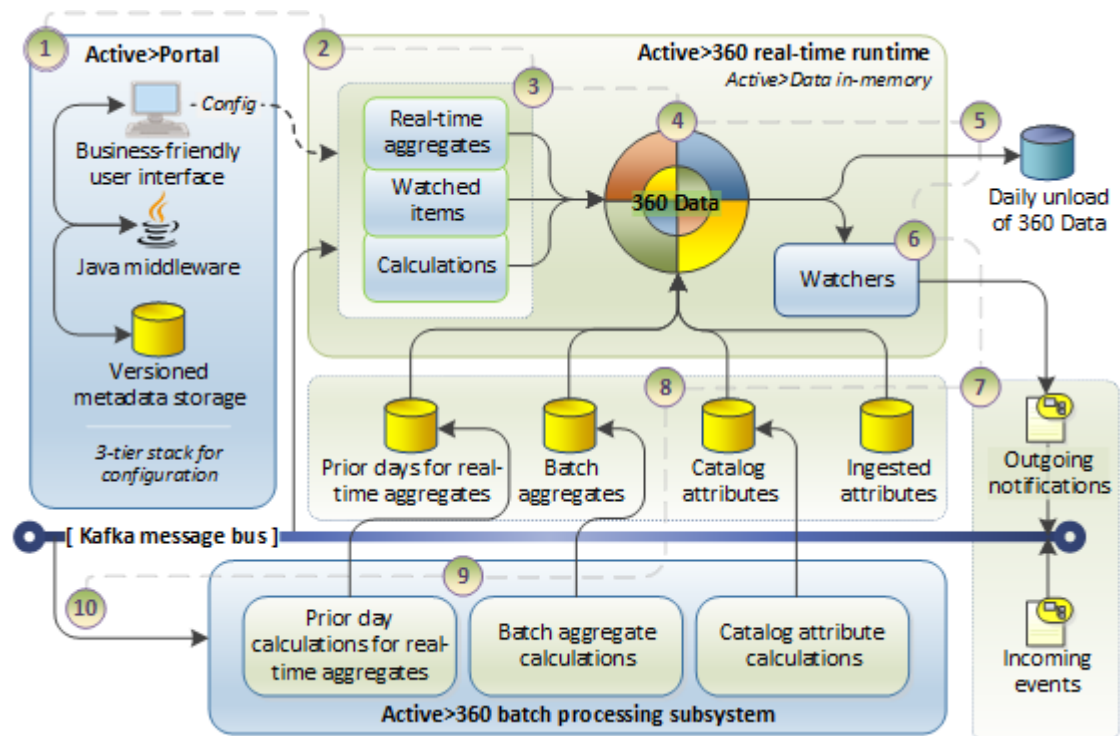
Before proceeding, it is useful to review some basic terminology used in this guide.

Term	Description
External interactions	<i>External interactions</i> are all exchanges of messages and data (typically networked) between Active>360 and anything that is <i>not</i> Active>360, such as enterprise applications, data integrations, and third-party services.
Active>360	<i>Active>360</i> refers to both the Active>360 runtime subsystem and the Active>Portal web application.
Public APIs	<i>Public APIs</i> are the set of Active>360 APIs that are exposed to enterprise applications and services outside of Active>360. These public APIs are the only Active>360 APIs that can — and must — be used for all external interactions between Active>360 and enterprise infrastructure components.
Private APIs	Private APIs are the set of Active>360 APIs that are used by Active>360 for internal interactions within and between the Active>360 runtime and Active>Portal application. These private APIs are not exposed to enterprise applications and services, and are not intended for use outside of Active>360 itself.

NOTE: This *API Guide* describes the Active>360 public APIs only. This guide does not describe the Active>360 private APIs.

General features of the Active>360 APIs

The Active>360 APIs support real-time and batch interactions, using some combination of HTTP or Kafka protocols, as summarized in "Summary of the Active>360 APIs". The primary interactions supported by the APIs are summarized in the following illustration.



In general terms, the Active>360 public APIs support the following kinds of interactions:

- Getting entities, getting events, and receiving notifications
- Handling real-time queries and responses through the Active>Portal
- Performing daily unloads of 360 Data
- Maintaining long-term aggregations
- Handling event enrichment messages through Kafka
- Accessing entity metadata and schema in read-only mode
- Handling project-based logging and ingested attributes
- Accessing the observability subsystem database through SQL
- Accessing event histories through SFTP

Benefits of the Active > 360 APIs

Using a single set of Active>360 public APIs for all external interactions serves several purposes, summarized in the following table.

Purpose	Summary
Security	Exposing a single set of public interfaces for all external interactions between Active>360 and the enterprise makes it easier for system architects, developers, administrators, and enterprise IT teams to implement consistent security policies across all Active>360 deployments.
Flexibility	Using a single set of public interfaces makes it possible to cleanly segregate enterprise integration components from Active>360 subsystems, which in turn makes it possible for Active>360 developers and enterprise developers to independently update their respective product features and implementations without affecting each other.
Accountability	Segregating enterprise integration components from the Active>360 subsystems also makes it possible to maintain lines of ownership and responsibility for feature requests, product enhancements, custom implementations, bug fixes, and so forth.
Testing	Providing a single, consistent set of public interfaces makes it easier to design testing protocols and test suites on both the Active>360 side and the enterprise side.

DRAFT

Summary of the Active>360 APIs

The following table summarizes the Active>360 runtime APIs, their real-time or batch-mode interactions, and whether they use HTTP or Kafka protocols. For more information about these and other Active>360 APIs, see ["Active>360 API reference"](#).

API	Description	Runtime mode	Direction	Protocol	Mini-RTS example
Ping	Ping the Active>360 runtime to check whether it is running.	Real-time	Get from Active>360	HTTP	Check Active>360 runtime status.
Get Entity	Get an individual 360 profile, optionally filtered by subset.	Real-time	Get from Active>360	HTTP	Enrich NBOS parameters NBOS as part of a Get Offer Request.
Get Batch 360 Unload	Interact with the Active>360 batch subsystem to perform batch unloads of 360 Data.	Batch	Get from Active>360	HTTP, file	Daily unload of 360 Data to disk.
Submit Event (synchronous)	Submit an event to Active>360 for synchronous processing.	Real-time	Push to Active>360	HTTP	Ingest CIP events into Active>360.
Event Submission (asynchronous)	Submit an event to Active>360 for asynchronous processing.	Real-time	Push to Active>360	Kafka	Ingest CIP events into Active>360.
Get Global	Get the value of a global variable.	Real-time, batch	Get from Active>360	HTTP, Kafka	
Set Global	Set the value of a global variable.	Real-time, batch	Push to Active>360	HTTP, Kafka	
Entity Status	Get the status of the runtime for a specific entity.	Real-time	Get from Active>360	HTTP	
Notification Subscription	Enables Active>360 to continuously subscribe to events in an external enterprise Kafka topic.	Real-time	Get from Active>360	Kafka	Subscribe to incoming events.

Active>360 API upgrade model

In order to make application maintenance and upgrades as practical as possible for both the enterprise and Active>360 developers, the Active>360 APIs are designed to be backwards-compatible with previous versions of the APIs. Specific implementation details and guidelines will be available in the first GA release of the Active>360 product.

DRAFT

Getting started with the Active > 360 APIs

The Active>360 APIs are included in the Active>360 product bundle as an EME Technical Repository project. By default, the Active>360 installer loads the Active>360 API project to a repository project directory named **Projects/abinitio/act/act_api**.

The installer also creates an API project sandbox named **\$act.root/sand/abinitio/act/act_api** and a main Active>360 project sandbox named **\$act.root/sand/abinitio/act/main**, where **\$act.root** is the root directory for the Active>360 installation. By default, the **act_api** project is included as common sandbox in the **main** project sandbox.

In order to use the features of the APIs in your private projects, you must include the **\$act.root/sand/abinitio/act/act_api** as a common sandbox in your private project sandbox. The remainder of this guide provides more detailed API concepts and usage instructions.

DRAFT

For more information

For more information about the Active>360 APIs, you may find the following resources helpful:

- **Active>360 Reference Application** — The Active>360 reference application, called the 360 Console, is installed by default with Active>360. The 360 Console provides API implementation exemplars and sample data against which you can experiment with various API calls and responses.
- **Active>Data Dashboard** — Active>360 makes extensive use of Active>Data for real-time operations. The Active>Data Dashboard application is available for all Active>Data installations, and the specific URL to access the Dashboard varies by installation. For the Dashboard URL for Active>360 installations, refer to the *Active>360 Installation Guide*.
- **Swagger utility pages** — The Swagger utility pages provide detailed information about every Active>360 API. You can access the Swagger utility pages from the Active>Portal application. To access the Swagger pages, choose, **Tools > Swagger** from the Active>Portal **Navigation** menu.

