Terminology

In addition to standard concepts, Wakanda has its own terminology that is used throughout this documentation, as well as in the Wakanda Studio interface and in the various Wakanda JavaScript APIs.

Alias Attribute

An alias attribute builds upon a Relation Attribute. Once an N->1 relation attribute is defined, any of the attributes within the “parent” class can be directly referenced as attributes within the “child” class. The result is what appears to be denormalized data without the overhead of duplicating information. Alias attributes can reference any available attributes further up the relational tree.

Application

A Wakanda application is a running Wakanda Project.

Attribute

An attribute is the smallest storage cell in a relational database. It is sometimes referred to as a column. However, the term “column” is more often associated with the results of a query where it may or may not be the value of a field. In Wakanda the collective term attribute is used to describe fields where information is stored as well as several other items that can return a value.

Bootstrap

A JavaScript file that is automatically executed by Wakanda Server when the project is launched. Bootstrap files can be set individually using the Explorer Area Action Menu or placed in a folder named "bootStraps".

Calculated Attribute

A calculated attribute doesn’t actually store information. Instead, it determines its value based on other values from the same entity or from other entities, attributes or methods. When a calculated attribute is referenced, the underlying “calculation” is evaluated to determine the value. For example, a “fullName” calculated attribute can result from the concatenation of values entered in the “firstName” and “lastName” separate attributes. Calculated attributes may even be assigned values where user-defined code determines what to do during the assignment.

Calculated Attribute Method

A method that substitutes programming behavior in place of standard behavior for an attribute. A method can be associated with each of the four events that calculate values for the attribute: *On Get*, *On Set*, *On Query*, and *On Sort*.

Catalog

A catalog is similar to a **Model**: it contains the collection of datastore classes, attributes and methods that describes the structure of a Wakanda application. A model can contain extra graphical properties, whereas the catalog strictly refers to the datastore structure.

Component

A Component is a widget in which you load a **Web Component** to be displayed on a Page.

Composition Relationship

A special relationship that allows for bundling child entities with parents.

Current Entity Collection

The selection of datastore class entities that a datasource works with by default on the client side. For example, the **orderBy( )** datasource method applies to the current entity collection of the datasource. The current entity collection can contain zero, 1 or X entities from the datastore class. By default, the current entity collection of a datasource contains all the entities of the datastore class. There is one current entity collection per datasource.

Data provider

The client-side proxy of the **Datastore**. The data provider delivers objects, methods, and functions to communicate with the application's datastore and allows user-defined code to easily access back-end information.

Datasource

A sophisticated and convenient object that is instantiated by the WAF and encapsulates many different functions. The datasource acts as a dispatcher for moving information from either entity collections/entities or JavaScript constructs (variables, arrays, or objects) and Wakanda's widgets. A typical datasource includes a list of elements as well as a current element.

Datastore

The Wakanda term datastore is used to reference a single application's set of data defined structurally by a **Model**. More globally, it is the back-end of an application served by Wakanda Server. The datastore is analogous to a database.

Datastore Class

A datastore class is a type of structure that can contain attributes and relation attributes (links between datastore classes) in order to conceptually describe its data and how they all interact with each other as well as methods to interact with the data in your Model. Datastore classes are stored on the server.

Datastore classes on the client are very similar to those on the server, but in some cases they exhibit different properties than under Wakanda Server.

Datastore Class Method

In addition to attributes, each datastore class may also include methods. There are three different scopes to datastore class methods: Class, Collection, and Entity.

Datastore Model

See Model.

Dependent Relation Attribute

A dependent relation attribute is one that is reliant on another relation attribute in the same datastore class.

Derived Datastore Class

A derived datastore class is a new class resulting from the "extension" of a parent datastore class. A parent datastore class, also called **Extended Datastore Class**, can generate one or several derived datastore classes, but a derived class has only one parent class. A derived datastore class can combine inherited attributes (that can be hidden), and added attributes. In the current version of Wakanda, only relation attributes and calculated attributes can be added to derived datastore classes.

ds

On the server, the **ds** property is a reference to the application's default datastore.

On the client side, it is the name of the global object used to reference the data provider.

Element

A general term for an item referenced in a **Datasource**. Since a datasource can have as its origin either an entity collection or an array, this term is used to reference a single item from either.

Entity

In traditional relational databases a record is the logical unit of storage within a table that holds a value for each field in the table. In a SQL database, a row is one of the results of a query, which may not correspond directly to a record. Since a Wakanda datastore entity can represent any level of denormalization without its accompanying shortcomings, it fulfills both of these roles. In this document and elsewhere, datastore entities are often referred to as entities.

On the client side, an entity is a representation of the like-named item from the datastore. As with datastore classes and entity collections, entities function substantially similar to those in Wakanda Server

Entity Collection

An entity collection is a set of entities belonging to the same datastore class. An entity collection can contain any or all entities of the datastore class. An entity collection can also be empty.

On the client side, an entity collection is a representation of the like-named item from the datastore. Client-side entity collections behave substantially similar to those on Wakanda Server.

Event Method

A method triggered by either a datastore class event or an attribute event. These methods are sometimes referred to as simply events.

Events

Any of several built-in trigger points in a Wakanda datastore model. Events are available for both attributes and datastore classes. Wakanda supports different events, such as *On Init* or *On Load*. For more information, refer to the section **Description of events**.

Extended Datastore Class

An extended datastore class is one that has generated at least one **Derived Datastore Class**. Datastore classes can be extended on several levels, so derived classes can also be extended.

GUI Designer

The GUI Designer (GUI stands for *Graphical User Interface*) is a Wakanda Studio built-in editor where you can design all the pages of your Web application. Each page can have several interfaces, that will be used depending on the client (desktop, tablet, mobile...).

Inherited Attribute

Similar to the "inheritance" concept found in object oriented programming, a datastore class can inherit from another. The newly defined (derived) class may add additional properties or hide inherited attributes. Datastore class inheritance may be defined by a restricting query stored in a project’s datastore model.

jQuery

A JavaScript framework that is used to reference and manipulate a web page. jQuery is included in WAF.

JSON

An information format. REST calls to Wakanda Server are formatted in JSON.

Method

A collective term for a discreet portion of programming in Wakanda. Calculated attribute methods, datastore class methods and event methods are all examples. A method is usually associated to a single JavaScript function.

Model

A complete set of all datastore classes for a single Wakanda project is referred to as a datastore model and is roughly equivalent to the relational concept of a single database structure. Datastore model is the top-level term for a Wakanda application’s datastore and incorporates all the datastore classes for the project/application. The datastore model is sometimes referred to as the model.

options

The *options* parameter is used in many WAF calls, usually to designate an object containing callback functions for asynchronous calls as well as additional options.

Parent Datastore Class

See Extended Datastore Class.

Primary Relation Attribute

A primary relation attribute is not dependent upon another relation attribute. Primary relation attributes are created by adding an attribute to a datastore class that refers to another class and either creates a new relationship between the classes or reverses the path of an existing N->1 primary attribute. Only primary relation attributes can be designated as a composition relationship.

Project

A Wakanda project is a design time concept that houses a single datastore model along with the other files that make up a single Wakanda application.

Property

A general term for either an attribute or a method associated with a datastore class.

Query

A query in a traditional database typically returns a row set, which can be thought of as a grid of values (rows and columns). A Wakanda query returns an entity collection, which is a bundle of entities, each of which “remembers” where it exists in the data.

Relation Attribute

Relation attribute is a collective term that exposes a relationship between two datastore classes and can be either N->1 or 1->N. Since they are attributes, they can be named and therefore become available as properties of the corresponding class. A relation attribute can result in an entity or an entity collection.

Relationship

Two datastore classes are associated via a relationship, which is an artifact of a datastore model. A relationship’s existence results in either one or two relation attributes, one in each of the related classes. Relation attributes are named and become a property of their respective classes.

REST

A protocol for communicating with a Web server. The WAF uses REST calls to Wakanda Server to send and retrieve information.

Restricting Query

A restricting query is a query that is automatically executed whenever all the entities in a datastore class are accessed, whatever the way they are accessed. A restricting query is ususally (but not always) used in the context of a **Derived Datastore Class**.

Scalar Attribute

A scalar attribute is a collective term to denote any attribute that returns a single value. All storage and most calculated attributes are scalar.

Solution

A Wakanda solution is a collection of projects, each of which contains a single datastore model. A solution can contain any number of projects, all of which can be accessible at the same time.

sources

The name of the global object used client-side to reference all datasources.

Storage Attribute

A storage attribute (sometimes referred to as a scalar attribute) is the most basic type of attribute in a Wakanda datastore class and most directly corresponds to a field in a relational database. A storage attribute holds a single value for each entity in the class.

userData

The *userData* parameter is typically inert and simply passes user-defined information to called methods.

WAF

Wakanda Application Framework – A collection of JavaScript objects and code that interacts with Wakanda widgets and datasources, thus providing access to Wakanda Server data.

Wakanda Loader

A term used for the portion of the WAF responsible for preparing the Page to be viewed.

Web Component

A Web Component is a package, containing HTML, JSON, CSS, and JavaScript files, that allows you to build complex and independent areas for your application. You can create Web Components to manage specific portions of a Page, like a login area, navigation menus, dialogs, etc. You then load the Web Component into a Component widget that is on your Page.

Widgets

Form controls that interact with the user. Wakanda's widgets are instantiated, maintained, and managed by the WAF.