1010data Glossary
Glossary
The glossary provides a list of terms and definitions with which you should be familiar when using 1010data.

**action**

An action is any operation applied to a table or worksheet that changes its state. Examples of actions include performing a tabulation, selecting rows, or creating a computed column.

See also:

- *operation* on page 8
- *Actions and Queries* in the 1010data User’s Guide

**aggregation**

An aggregation is the collection of information in a summary form, for purposes such as statistical analysis. A common reason to perform an aggregation is to get more information about particular groups based on specific variables such as age, profession, or income.

In database management, an aggregation is the result of a function that takes the values of multiple rows which are grouped together and calculates a single value of more significant meaning or measurement. Common types of aggregations include: average, count, maximum, median, minimum, and sum.

See also:

- *summarization* on page 13
- *tabulation* on page 13

**analytical database**

An analytical database is typically used to store, manage, and consume data. It is designed and built specifically for use with business intelligence (BI) solutions. An analytical database stores business, market, or project data used in business analysis, projections, and forecasting processes. An analytical database generally provides faster query response times than a relational database and is often more scalable.

**Application Programming Interface (API)**

An application programming interface (API) is a set of routines, protocols, and tools for building software and applications.

The 1010data Application Programming Interface allows a client application running on a user’s machine to access and query data on the 1010data servers. This allows customized applications and interfaces to take advantage of 1010data’s database management services and fast analytics engine.

The API uses HTTP and XML and is compatible with any client application written in a language that supports HTTP transactions (such as Java, Visual Basic, C++, Python, and PERL).

See also:

- *1010data API Reference Manual*
**block**

Blocks in 1010data are self-contained, modular pieces of code that can be reused. Because a block is modular, it can be inserted within a query, at which point the Macro Language code referenced by the block is executed.

Blocks extend their reusability by also allowing for parameterization through the use variables. Variables can be referenced within a block, and their corresponding values will be substituted for the references. This means that the value of a variable can be changed, and that change will propagate throughout the block.

See also:
- *Blocks* in the 1010data Reference Manual

**cache**

The cache is an internal store of computational results as well as the history of actions performed by queries during your 1010data session. Performance is enhanced by the use of the cache since the results of prior actions may be used in subsequent queries, thereby eliminating the need to re-run those actions.

The bigger your cache, the more memory in your workspace is used. Clearing your cache frees up workspace memory, but previously cached operations are no longer saved.

See also:
- *workspace* on page 15

**column**

A column is a set of data values of a particular type arranged in a vertical list. For example, a column in a table may contain a list of last names for all the employees at a company.

Each column name in a particular 1010data table must be unique.

See also:
- *column label* on page 4
- *column name* on page 5
- *row* on page 10

**column label**

The column label is an optional descriptive column title.

The column label may contain any combination of uppercase and lowercase letters, numbers, spaces, and special characters. You can create a multi-line column label by using the backtick character (`) to separate the lines (e.g., "Percentage of Total Sales (%)").

The column label, column name, or a combination of the two may be displayed in the column header at the top of a column in the grid.

See also:
- *column* on page 4
- *column name* on page 5
column name

The column name is the name used to refer to a column in Macro Language code. It can be used in value and selection expressions, as the value of a Macro Language element's attribute, or as the value of a parameter in a 1010data function.

The column name may only contain alphanumeric characters or underscores and must begin with an alphabetic character (e.g., percent_total_sales). It may not contain any spaces or other special characters.

The column name, column label, or a combination of the two may be displayed in the column header at the top of a column in the grid.

See also:
• column on page 4
• column label on page 4

columnar database

A columnar database is a database management system (DBMS) that stores data in columns instead of rows. Columnar databases speed up the time it takes to return a query by efficiently reading and writing data from and to memory and hard disk storage. In a columnar database, all the column 1 values are physically stored together, followed by all the column 2 values, and so forth.

computed column

A computed column is a column that is added to a table in a query. A computed column is determined by a given value expression. The value expression may refer to one or more columns and may include standard arithmetic, relational, and logical operators.

For example, if you had a table of sales data, you could create a computed column for the margin by specifying a value expression that subtracted the cost column from the sales column.

See also:
• Computed Columns in the 1010data Insights Platform Legacy Interface User's Guide

Consumer Insights Platform (CIP)

The Consumer Insights Platform (CIP) is a collection of reports that can be adapted to different datasets. Access to the CIP can be controlled and permissioned.

cross tabulation

A cross tabulation is the result of an operation that allows you compare the relationship between two variables. In 1010data, a cross tabulation summarizes the values in a column based on the values in two or more other columns and displays the result as a matrix.

See also:
• Summarizations and Tabulations in the 1010data Insights Platform Legacy Interface User's Guide
• Perform a tabulation in the Getting Started Guide
• Cross tabulations in the Getting Started Guide
**dynamic variable**

A dynamic variable represents a scalar value in a QuickApp. A dynamic variable is declared and can be set to an initial value in the opening `<dynamic>` tag in the Macro Language code for the QuickApp. The dynamic variable can be referenced in a QuickApp using the syntax `@var` (where `var` is the name of the dynamic variable). Dynamic variables may be referenced in both scalar expressions and value assignments.

**environment**

An environment is a single 1010data system installation on a particular server cluster. Users and tables exist within a particular environment. One of the most common environments is accessible from www2.1010data.com.

**Excel Add-in**

The 1010data Excel Add-in is a utility that enables Microsoft Excel to communicate directly with the 1010data server.

Using the 1010data Excel Add-in, you can run a query on 1010data from Excel and have the results directly downloaded into an Excel worksheet. You can also upload data from an Excel worksheet to a 1010data session.

See also:
- 1010data Excel Add-in User's Guide

**expression**

An expression is the composite of any number of values, variables, column names, operators, and functions that evaluates to a certain value. The resultant value can be one of the simple 1010data types, such as integer, decimal, or text; or it can be one of the complex types, such as package, list-value, or model.

An expression is similar to a formula in Microsoft® Excel®.

Expressions may contain various operators (e.g., `+`) and functions (e.g., `min(X;Y)`) and may refer to column names (e.g., `price`) as well as explicit values (e.g., 1.01). They may also contain certain predefined variables (e.g., `i_`).

See also:
- scalar expression on page 10
- selection expression on page 11
- value expression on page 14

**function**

In 1010data, a function is a computational tool used in an expression that computes the result for a given set of arguments, which provide input. Functions can perform mathematical operations on numerical values (e.g., calculate the sum of a certain group of values), be applied to date values (e.g., find the number of days between two dates), or manipulate string values (e.g., return the result of concatenating two strings together).
1010data offers a broad collection of over 300 computational functions for everything from string manipulation to complex statistical modeling. The Function Reference in the 1010data Reference Manual contains documentation for these functions and is organized into categories, such as mathematical, time and date, and string. One of the most powerful categories of functions is the Group Functions (or g_functions), which perform operations on groups of values.

See also:
- Function Reference in the 1010data Reference Manual

**g_function**

Group functions (g_functions) are used to perform various kinds of calculations across a particular set of rows, grouping by unique values, within one or more columns in a table.

Oftentimes, especially with basic g_functions, the operation will perform some calculation on one set of data while grouping by another. For instance, you can use a g_function to calculate total sales by store or average temperature by city. In both these instances, the data that comes after the word “by” (store and city) is a group. G_functions can provide similar functionality as tabulations, but are often faster and do not result in the loss in granularity of the data on which they operate.

See also:
- Group Functions in the 1010data Reference Manual

**linking**

Linking two tables or worksheets together combines the columns from both into a single, larger worksheet. The results apply only to your session; the original tables or worksheets are not affected. Linking in 1010data is similar to a VLOOKUP in Excel and various types of SQL joins.

**Note:** Linking differs from merging. Linking combines the columns of two tables or worksheets together whereas merging combines the rows of two or more tables or worksheets together. Links align worksheets side by side, while merges combine worksheets vertically.

See also:
- merging on page 8

**link header**

The link header is the text that is prepended to column headings from a foreign table after a link operation. It can be specified using the Label field within any link-related dialog in the Trillion-Row Spreadsheet (e.g., Link in Another Worksheet), the label attribute in the opening tag of the <link> operation, or in the <link> element in the XML Table Tree when using the API.

**list-value**

A list-value is a variable that contains multiple values called elements. Similar to lists in other high-level languages, list-values in 1010data provide a useful way to collect scalar values into a single variable and refer to them by index.

See also:
- package on page 9
- List and Package Types in the 1010data Reference Manual
Macro Language

The 1010data Macro Language is the XML-based compositional language in which 1010data queries are written. Macro Language consists of a set of elements. These include data transformation operations, block code, and application development elements. Each of these elements has a set of attributes that are used to provide additional information to the element.

For example, in the 1010data Macro Language, there is a `<sort>` element that is used to sort the data in a table. The `<sort>` element has a `col` attribute, which specifies the column whose values will be used to order the rows, and a `dir` attribute, which specifies the direction in which to sort. In the Macro Language code, this example would be written: `<sort col="date" dir="up"/>

merging

Merging two or more tables or worksheets combines their rows together into a single, larger worksheet. The rows from the foreign table(s) are appended to the end of the base table in the order in which they are specified. The results apply only to your session; the original tables or worksheets are not affected. Linking in 1010data is similar to a various types of SQL unions.

Note: Merging differs from linking. Merging combines the rows of two or more tables or worksheets together whereas linking combines the columns of two tables or worksheets together. Links align worksheets side by side, while merges combine worksheets vertically.

You can perform a merge using the `<merge>` operation, which can be performed within a 1010data query, or the merge API transaction, which can be used in a client application or QuickApp.

See also:
- linking on page 7
- `<merge>` in the 1010data Reference Manual
- merge (Merge table) in the 1010data API Reference Manual

Open Database Connectivity (ODBC)

Open Database Connectivity (ODBC) is an open standard application programming interface (API) for accessing a database.

The 1010data ODBC driver is a software utility that serves two primary functions. First, it provides a standard interface so that applications can connect to 1010data directly. Second, it allows the 1010data system to understand SQL. The 1010data ODBC driver conforms to the ODBC 3.0 specification.

See also:
- 1010data ODBC Driver User’s Guide

operation

An operation performs a specific action on data in a table or worksheet. At the core of 1010data are five basic operations: Select, Link, Tabulate, Create Computed Column, and Merge. These operations, as well as the other data transformation operations, take one or more tables as input and produce a resultant table as their output.

See also:
- action on page 3
package

Packages are compound variables containing key/value pairs. Packages use keys to reference their values, making them similar to a dictionary or associative array in other languages. Referencing package values is very similar to accessing values in lists, except the values in the package are referenced by keys, not index values.

See also:
- list-value on page 7
- List and Package Types in the 1010data Reference Manual

PowerLoader

PowerLoader is a browser-based user interface that allows you to load your data into the 1010data platform. PowerLoader is useful for loading flat text files such as comma-separated value (.csv) and tab-delimited (.txt) files.

See also:
- PowerLoader User's Guide

prelink

A prelink is a precalculated linkage between two tables on a specific set of columns. Each prelink is saved with the base table on which it is applied. During an analysis, the link does not need to be calculated, only read. A prelink improves the speed of linking larger tables together.

query

A query is a list of the actions performed on a table to achieve a particular result or to perform a specific analysis. In other words, a query is the sequence of transformations that are performed on a particular base table.

See also:
- Actions and Queries in the 1010data Insights Platform Legacy Interface User's Guide

Quick Query

A Quick Query is a saved query that can be rerun at anytime. Quick Queries are used to save your work so that you can use it later or share it with others. Quick Queries allow simple parameterization, the ability to choose different inputs when running the saved query.

See also:
- Quick Queries in the 1010data Insights Platform Legacy Interface User's Guide

QuickApp

A QuickApp™ is an interactive application that provides a custom front-end interface to the 1010data analytical platform. Depending on the functionality built into a QuickApp, a user can interact with data, provide input, and view the results of queries in tabular or graphical form.
QuickApps are built using a set of tags in 1010data's Macro Language: `<dynamic>`, `<widget>`, and `<layout>`. These tags work in conjunction with block code and the other data transformation operations to specify how the QuickApp should accept user inputs and display query data.

See also:
- `widget` on page 15
- `Intro to QuickApps` tutorial

**row**

A row consists of a set of related values from all the columns in a particular table. The data within a row may appear in any order by rearranging the columns in a table without changing the underlying structure of the table.

See also:
- `column` on page 4
- `Rows` in the 1010data Insights Platform Legacy Interface User's Guide

**SAM Pool**

Shared Access Management (SAM) pools enables a single set of credentials to be shared between client side threads to leverage multiple threads of parallelism on the 1010data Insights Platform.

**scalar expression**

A scalar expression is an expression that results in a scalar value. In Macro Language code, a scalar expression is surrounded by braces (e.g. `{sqr(5)}`). Unlike value expressions, which operate on and result in vectors, scalar expressions use scalar values and variables to evaluate to scalars.

See also:
- `expression` on page 6
- `selection expression` on page 11
- `value expression` on page 14
- `Scalar Expressions in 1010data` in the 1010data Reference Manual

**scalar value**

A scalar value is either an individual value such as an integer or string, or a value containing multiple components which can be referenced individually or as a whole such as a package or list-value. A scalar value is different than a vector.

A scalar variable can be referenced in a query using the syntax `@var` (where `var` is the name of the scalar variable). Scalar variables may be referenced in both scalar expressions and value assignments.

See also:
- `scalar expression` on page 10
- `vector` on page 15
**segby**

Segby is a specific segmentation where the values in the segby columns govern the way the rows are split. If a table is segby a given column, no unique value of the column can be found in more than one segment. This allows for quick computation of aggregate/grouping functions (e.g., sums, averages, etc.) on that column since it is only necessary to look in one file for each unique value of that column. To achieve this, the rows of the table frequently need to be reordered.

See also:
- *segmentation* on page 11
- *sortseg* on page 12

**segmentation**

Segmentation is the process of partitioning/splitting a table horizontally in the underlying file structure so that not all rows live in the same file.

For example, a 45-row table can be split so that each "segment" (i.e., file) contains ten rows; this would yield five segments, where the first four segments contain ten rows and the last segment contains five.

The 1010data Insights Platform provides two specialized forms of segmentation: *segby* and *sortseg*. If a table is segby a given column, no unique value of the column can be found in more than one segment. If a table is sortseg on a particular column, not only are unique column values not allowed to be found in different segments, the segments themselves are internally sorted on the sortseg column. These specialized forms of segmentation allow for optimized performance when aggregating (or using g_functions) on the segmented column.

See also:
- *segby* on page 11
- *sortseg* on page 12

**selection expression**

A selection expression is an expression used for selecting a subset of rows from a table or worksheet, usually based on some comparison criteria.

A selection expression is used in the `value` attribute of the `<sel>` operation. A selection expression generally resolves to a 0 or 1. Rows for which the expression evaluates to 1 remain in the resultant worksheet; rows where the expression evaluates to 0 are omitted. However, there are exceptions such as when you are using the `expand` or `sample` attributes.

See also:
- *expression* on page 6
- *scalar expression* on page 10
- *value expression* on page 14
- `<sel>` in the *1010data Reference Manual*
- *Writing Expressions* in the *1010data Reference Manual*

**session**

A session is an instance of a particular user ID logged into 1010data within a certain environment. You can only have one active session at a time for a single user ID across all environments. In addition, a session
can only have one transaction occurring at a time; two queries cannot be run simultaneously in the same session.

**shifting**

Shifting allows you to move rows of data according to a defined interval. The interval could be based on time, a count of rows, or a relative relationship between two points in a vector. For example, to compare sales data of the current month to sales data for the same month one year prior, you can shift the rows in your worksheet so that the same month from the two different years are in adjacent columns. Once the aggregations are in the same row, calculations such as the difference between the two or percent of change is simple.

Performing a time comparison analysis allows you to examine how your data changes from one time period to another.

See also:
- *time series* on page 14

**Software Development Kit (SDK)**

A software development kit (SDK) is a set of tools or functions that provide for the creation of applications for a certain software package, software framework, hardware platform, computer system, operating system, or similar development platforms. With any of the 1010data SDKs, developers can create native and web-based applications that use the 1010data analytics engine.

1010data offers SDKs supporting many popular programming languages including C, C++, .NET, Java®, Python, and Visual Basic® for Applications (VBA).

**sortseg**

Sortseg, like segby, is a specific segmentation governed by the sortseg columns. It has an even stronger restriction than segby. Not only are unique column values not allowed to be found in different segments, the segments themselves are internally sorted on the sortseg column. The table itself is not guaranteed to be globally sorted, but the segments are guaranteed to be disjoint on the sortseg column.

See also:
- *segmentation* on page 11
- *segby* on page 11

**sticky**

Sticky refers to a value, such as a row number or the number of rows in a worksheet, that remains persistent despite changes to the current worksheet.

For instance, a sticky row number for a particular row is determined from the current worksheet at a given evaluation point in the query. A sticky row number will not change regardless of whether the number of rows or relative position of that row changes due to operations performed on that worksheet after the evaluation point. There are three sticky system values: $i_0$, $ii_0$, and $n_0$.

As an example, consider a row that was the 100th row in a base table but which is the 5th row after a transformation. The sticky row number for that row would still be 100.

See also:
- *System Variables* in the 1010data Reference Manual
**summarization**

Data summarization is the calculation of certain statistics and the display of those results in the form of tables, graphs, or charts. In other words, a summarization is information gathered and displayed in summary form. For example, a simple summarization could calculate the total sales in dollars for every store in a retail chain during a given period of time. In 1010data, the results of the summarization would be a worksheet with two columns and one row for each store in the original table. The first column lists the store identifier, and the second contains the total sales in dollars for that store.

See also:
- [aggregation](#) on page 3
- [tabulation](#) on page 13
- *Summarizations and Tabulations* in the 1010data Insights Platform Legacy Interface User's Guide

**table**

A table is a collection of data that is stored as rows and columns. In 1010data, a table is the permanent, unchanging version of the data that is saved on the server. 1010data uses a columnar database to store its tables, which differs from a relational database.

See also:
- [columnar database](#) on page 5
- [worksheet](#) on page 15

**tabulation**

A tabulation allows you to group the values in a column (or columns) based on the values in another column (or columns) and summarize the data for each group. For example, a table containing demographic information of employees in a company could be used to determine the total number of employees by age group and gender. In this example, a tabulation could group the employee records by gender and then summarize the total number of employees in each age group.

See also:
- [aggregation](#) on page 3
- [summarization](#) on page 13
- *Summarizations and Tabulations* in the 1010data Insights Platform Legacy Interface User's Guide

**TenDo**

TenDo™ is a command-line interface for executing and automating queries in 1010data.

See also:
- *TenDo User's Guide*

**TenUp**

TenUp™ is a command-line interface for extracting data from ODBC-compliant databases and loading it into 1010data.
time series

A time series is a sequence of data points ordered by time. In 1010data, many functions operate on table data as a function of time. Most notably, many g_functions accept an order argument which is often a column of chronological values (e.g., date or time).

See also:
- shifting on page 12
- window on page 15

transformation

A transformation is the result of a set of operations or actions that have been applied to a table or worksheet.

Trillion-Row Spreadsheet (TRS)

The 1010data Trillion-Row Spreadsheet® (TRS) is a graphical user interface (GUI) that allows you to visually interact with your data on the 1010data platform. The 1010data GUI is a browser-based interface that is used mostly for ad hoc data analysis.

Universal Calculation Library (UCL)

The Universal Calculation Library (UCL) is a 1010data library of blocks that provides a collection of commonly performed calculations for mortgage-backed security (MBS) data. Numerous MBS data sets are supported by the UCL, including eMBS and CoreLogic.

user ID

A user ID is associated with an individual account in 1010data. A user ID can only be logged in to one session of 1010data in a particular environment at a time.

value expression

A value expression is a mathematical formula that is used to determine the value of a computed column or as a qualifier when selecting rows.

See also:
- expression on page 6
- scalar expression on page 10
- selection expression on page 11
  - Writing Expressions in the 1010data Reference Manual
vector

A vector is an entity having multiple values of the same type that can be operated on as a single unit. In 1010data, an example of a vector is a column, in which there is one value for each row in the column. Operations and certain functions are applied to the column as a whole, essentially performing the calculation or action on all of the values in the column.

widget

A widget is an individual component of a QuickApp that can be used for displaying data, accepting user input, or both.

A widget can be used to visually represent the results of a block of Macro Language code, such as in a spreadsheet-like grid or a bar chart. It typically contains a single 1010data query that defines the data that it displays. Widgets can also be used as a means for user input, such as an input field or drop-down list.

See also:
• `<widget>` in the 1010data Reference Manual
• Intro to QuickApps tutorial

window

A window is an interval period specified in certain 1010data functions related to moving and shifting calculations. These functions allow you to examine how your data changes from one interval to another. For example, an ongoing analysis that compares store sales of the current week to store sales for the previous week uses a moving seven-day window that is one week prior to the current date. The size of an interval can be determined by a number of factors such as a number of rows, a given time period, or any other ordered sequence.

See also:
• shifting on page 12
• time series on page 14
• Shifting and Windows in the 1010data Reference Manual
• Moving Calculations in the 1010data Reference Manual

worksheet

A worksheet represents the temporary state of the data after one or more operations have been applied to a base table in 1010data. Any operations performed on a worksheet are temporary and do not impact the state of the original table.

See also:
• table on page 13
• Tables and Worksheets in the 1010data Insights Platform Legacy Interface User's Guide

workspace

The workspace is the amount of virtual memory that has been allocated to your 1010data session. As you perform an analysis, 1010data stores computational results as well as the history of actions performed by queries in the cache. The workspace indicates the amount of memory you have used. Clearing the cache
cleans out your workspace so that you have the maximum amount of virtual memory available to your 1010data session.

See also:

- *cache* on page 4
- *Workspace* in the *1010data User's Guide*