

Calculation Names and Identifiers

Last Modified on 06/04/2020 1:58 pm AEST

Names are used in a calculations to refer to various data values such as field names, variable names and so on.

Overview

Names, also called *identifiers*, are the text labels within a calculation that refer to different sources of information.

Names may refer to:

- Field names
- Relationship names
- Calculation Variable names
- Workflow Variable names
- Object names
- Record names

Representing names in calculations

Simple names may be used directly in calculations. More complex names that contain spaces or certain special characters need to be enclosed in square brackets.

For example, consider the calculation:

```
[Last Name] + ' ' + Status
```

In this example, **Status** is a simple name and may be used directly, whereas **Last Name** must be surrounded by square brackets because it contains a space. Note that square brackets may also be used around simple names, so [Status] is also valid.

More details about simple and complex are described later on this page.

Case Insensitive

Names are *not* case sensitivity, which is to say that you do not need to have correct or consistent upper and lower case capitalization.

However, it is recommended that you try to use capitalization consistently because it will make your calculation easier to read.

Relationship Names

Relationships between records can be following from either record to the other. However, typically the natural name of a relationship in one direction is different to the name of the relationship in the other.

For example, the relationship from an employee to their manager might be **Employee's manager**, or simply **Manager**. Conversely, the same relationship from the manager back to their team members would more appropriately be **Direct Reports**.

ReadiNow calculations require that the correct name be used for the direction that is being followed.

See [Relationships](#) to learn more about relationship names.

Tip: When you create a new relationship, the default reverse name of the relationship is the name of the object that the relationship comes from. For example, if you create a relationship from the Incident object to the Employee object, the default reverse name is Incident. In many cases this is a suitable name - but it should be reviewed to make sure it does not cause later confusion.

Changing Name - Script Names

Fields and relationships have a setting (located in the field or relationship properties window within the form builder) called Script Name. The Script Name is the name that must be used in a calculation. Relationships have one Script Name for each direction.

When a field or relationship is first created, the Script Name is initially set to be the same as the name that is entered. However, if the name is later modified to something else then the script name does not automatically change. This allows fields and relationships to be renamed without risking problems for any existing calculations, workflows or APIs.

However, only the script name may be used in calculations.

If you wish to change the Script Name for a field or a relationship, then you may do so, however you will need to update the name in any calculations, workflows that are referring to it.

Non-Unique Names

Precedence for different types of names

If the same name could refer to a field name, variable name, or workflow variable name, then the following precedence rules are applied:

1. **Calculation Variable name** - highest precedence, so that if a new field (conflicting) field is added to an object or workflow then existing calculations are not impacted. Calculation Variables only have meaning within the calculation so can be easily changed you wish to refer to something else.
2. **Workflow Variable name** - next highest precedence, so that if a new field (conflicting) field is added to an object then existing workflows are not impacted. Workflows Variables meaning within the workflow so can be changed if you need to refer to a field or relationship, but need to be changed throughout the workflow.
3. **Field or Relationship name** - lowest precedence.

Explicitly referring to a particular type of name

Consider a calculation, such as the following, that has a calculation variable called status, within a workflow that also has a variable called status, and the calculation is working with an object that also has a field called status

```
let status = 'x'  
select all([Employee] where Status = 'Available')
```

In this example, the second usage of the name Status is ambiguous, but the precedent rules would mean that it would refer to the calculation variable defined on the line above. If we want to access the status choice field, then the most natural option is to change the variable name:

```
let somethingElse = 'x'  
select all([Employee] where Status = 'Available')
```

However, another option is to use the **context** function to select the record, which then makes it clear that we intend to access the choice field:

```
let status = 'x'  
select all([Employee] where context().Status = 'Available')
```

Or, if there is a workflow parameter called `status`, then we can prefix the name (and any square brackets) with the `@` symbol to explicitly specify that a workflow variable is being referenced:

```
let status = 'x'  
select all([Employee] where @Status = 'Available')    -- explicitly refer to  
a workflow variable called 'Status'
```

Field and Relationship names must be unique

When referring to fields, relationships, objects and records, the name must be uniquely identify. If a name is used in a calculation that does not uniquely identify the item (in the context that the name is being used) then the calculation builder will show an saying that the name is not valid.

For example, if an object named `ChildObject` has a field named `Status`, but the object inherits another object, `ParentObject`, that also has a field named `Status`, then a calculation in a `ChildObject` report cannot refer to the `status` field because it is ambiguous.

However, if the report is based on the `ParentObject`, then the calculation can refer to the `status` field because the fields of the child object are not available in that context (unless the calculation uses the **convert** function to treat the record as a `ChildObject`).

Name conflicts can be resolved by giving the two fields different Script Names. See below for details.

Simple and Complex Names in detail

Simple Names

A name is a simple name and can be used directly in a calculation if:

- it starts with a letter
- and it only contains letters and numbers - i.e. with no spaces or other characters

If a name is the same as a calculation keyword, then - depending on the scenario - it may not be possible to use it as a simple name.

Complex Names

If a name contains spaces or special characters then it must be enclosed within square brackets within the calculation.

Once a square bracket is encountered, all characters are then treated as part of the name until a closing square bracket is found.

Examples:

```
[Name with spaces]
```

```
[This & that character.]
```

Escape Sequence

Most special characters are allowed within square brackets. Very rarely you may need to refer to a field name that actually contains a square bracket in the name itself. In this scenario, the square bracket must be represented (*escaped*) by typing the square bracket twice.

For example, to represent the name **A [slightly] complicated name** use:

```
[A [slightly]] complicated name]
```