Data types within calculations

Last Modified on 13/06/2019 12:32 pm AEST

Introduction to data types in calculations

Every calculation produces a result of a particular data type, such as a numeric result or a particular type of record, for example. Understanding how ReadiScript considers data types can be helpful in understanding how to create and trouble-shoot calculations.

Escape characters

Text data is delimited by a single quote, which means that this character will cause an error if it appears in a literal parameter. To use a quote in literal text it is necessary to employ the escape character before the quote character. The escape character is defined as a backslash.

```
result = len('Roger's new scooter') /* invalid calculation */
result = len('Roger\'s new scooter') /* valid calculation */
```

Result types are based on the calculation structure

ReadiScript determines the result data type of a calculation by considering the calculation itself, and without regard to the specific results that are calculated on a particular run. This approach is often referred to as *static typing*. For example: a calculation that returns all employees who have a status of "On Leave" will have a 'list of Employee record' data type - even if there is only one employee that is on leave.

Calculations are usually made up of smaller sub-calculations, often referred to as *expressions*. Each sub-calculation also has a result type. The result types of these sub-calculations can influence how ReadiScript interprets a calculation.

In the following example, the sub-calculations [Quantity in inventory] and [Quantity received] may both represent number fields. The plus operator will perform a numeric addition if presented with two sub calculations that both have a number result type.

```
[Quantity in inventory] + [Quantity received]
```

Whereas, in the following combination, each sub-calculation has a text result type, so the plus operator will combine the two text values into a single long text value.

```
[First name] + ' ' + [Last Name]
```

List of data types

ReadiScript supports the following data types:

string	Represents text, including formatted text.
int	Represents a whole number.
decimal	Represents a number with a decimal component.
currency	Represents an amount of money.
bool	Represents a yes/no, or true/false, value.
datetime	Represents a date and time, and is time-zone aware.
date	Represents a date only, and is *not* time-zone aware.
time	Represents a time of day only, and is *not* time-zone aware.
record	Represents a record of a particular type.

In addition to the above, a result data type also includes:

- for strings, encoding information (for example if the string is JSON or XML for use in API callouts).
- for decimal, the number of decimal places.
- for records, the type (object) of record that is returned.
- for all of the above, the result type may be a single value or a list of values.

For example, the following calculation has a result type of "List of employee". That is the result type, regardless of whether there are zero, one, or many records returned; and also regardless of whether the actual records returned are exactly 'Employee', or of some other object that inherits from 'Employee' such as 'Manager'.

```
all( Employee )
```

Type conversion

Each function and operator accepts particular data types for input, and also a particular result type for the larger calculation that is formed. When a sub calculation is not of the correct type for use with a function, or when two sub calculations of different types are combined together, an automatic conversion may occur.

In the following calculation, the [Manager] sub expression may return a Manager record. The len function accepts a string, and returns a number (specifically, the length of the string). The manager record will be automatically converted to a string by using the name of the record. The len function, which returns a number is then added to the decimal 1.234. In order for this addition to occur, the whole number is automatically converted to a decimal, and so the overall result type is decimal.

```
len( [Manager] ) + 1.234
```

See Converting between data types for more details.

Record types

If a sub expression returns a **record** result data type, then the result data type also keeps track of which object might be returned. This always includes any inheriting types, but it may be necessary to use the convert function to access fields and relationships that only apply to those inheriting records.

The following calculation may, for example, have a result data type of P**erson record.** But when the calculation runs, a more specific type of inheriting record, such as a Manager record may be returned.

```
[Process Document].[Last Reviewed By Who]
```

Expected result data types

In some ReadiNow features, such as when adding a calculated column to a report, a calculation of any result type may be used.

In other ReadiNow features, there may be specific constraints on what result data type a calculation may have. For example, the calculation for a calculated field may only return a single value, not a list. Attempting to use a calculation that is determined to have a **list** result data type will result in a calculation error.

A challenge can arise when a calculation is designed such that the author/administrator knows there is only one result, but ReadiScript has determined that the calculation returns a list result. In these cases, aggregate functions such as any , count , and first can be used to guide ReadiScript.