eServices Manager Administrator's Guide

Using Formulas in Field Codes

6/13/2026

Designer User's Gui

Business H

Contents

- 

- Administrator

View formula usage in field codes.

Related documentation:

-

In addition to system variables such as `Contact.FirstName`, field codes may contain formulas. This section provides an outline of formula usage. Details on many of these topics are provided in the Genesys eServices Field Codes Reference Manual.

You must always delimit field codes by using `.` If you type a field code directly into the body of a standard response, then you must enter the delimiters yourself. If you select from the list of field codes in eServices Manager, then the delimiters are added automatically.

The text that appears inside the delimiters is a formula. Field code formulas are very similar to formulas in other applications, such as Microsoft Excel.

A *formula* is a sequence of one or more operands (such as numbers and text strings), separated by operators (such as + and `-`).

For example, in the following formula, 2 and 3 are operands and + is an operator:

Operands can be values that do not change (constants), or values that vary based on the context. In the previous formula, all the operands are constants, so the formula always evaluates to `5`. The next formula, on the other hand, evaluates to a different value for each agent who uses it:

Field Code Syntax

To summarize field code syntax:

- A field code must be delimited by `.`
- Alphabetic strings, whether constants in formulas or elsewhere in a field code, must be enclosed in double quotes.
- Numeric constants require no special treatment.
- You must use special characters for some purposes. For example, for your field code to render with a line break, you cannot simply type a carriage return. Instead, you must insert the expression `\n`. A list of these special characters is available.

HTML in Field Codes

With special configuration, field codes can contain HTML markup; for example, you could have a field code defined as

```
Sam Agent

Acme Products

29 Exterior Blvd

Springfield, CX 09090
```

To enable this, you must use the Java property `-Dsrl-field-code-allow-html=true`, in one of the following ways:

- Add it to the `JavaArgs` section of `ContactServerDriver.ini`
- Add it as an argument to the startup command line in `contactServer.sh`.

Operator Precedence

If you use more than one operator in a formula, the order in which they are evaluated depends on their relative *precedence* (higher precedence operators are evaluated first). For example, multiplication (`*`) has a higher precedence than addition (`+`), so that the formula below evaluates to `14`, not `20`:

You can use parentheses to override the default precedence. The formula below evaluates to `20`:

For a complete list of operators and their relative precedence, see "Operator Precedence" in the Genesys eServices Field Codes Reference Manual.

Data Types

Operands of several different types may appear in formulas:

- Number
- String (text)
- Date/time
- Boolean (true/false)
- Object (Contact, Interaction, and Agent)

Each data type behaves differently in formulas, and the operators have different meanings when you

use them with different data types. For example, the + operator means “add” when used with numbers, but “concatenate” (paste together) when used with strings. This formula evaluates to *Uncle Sam Wants You*:

In addition, some operators cannot be used with some data types at all. For example, you cannot use the multiplication (*) operator on two strings.

All formulas, regardless of their final data type, are converted to strings before being merged into your standard response. This conversion follows a set of default rules that depend on the data type. For example, the default rules for numbers round them off to integers. This formula causes 2 to be inserted into your standard response, even though the real result is `2.25`:

You can use the `Text` function (see below) or `format operator`:) to override the default formatting. Either of the following formulas inserts `2.25` into your standard response:

For a detailed list of data types and how you can use them, see "Data Types” in the Genesys eServices Field Codes Reference Manual..

Functions

When composing formulas, you can use many built-in functions. *Functions* are predefined formulas that perform calculations using values, called *arguments,* which you supply. To use a function, write its name, followed by an opening parenthesis, the arguments for the function separated by commas, and a closing parenthesis.

Function arguments may be of any data type, although individual functions may place restrictions on their arguments. Function arguments may be constants or formulas. The `Length` function, for example, takes a single string argument and returns its length in characters. This formula evaluates to `13`:

As another example, the `Date` function takes individual date components (year, month, day, and so on), and constructs a date/time value. The formula below evaluates to `1965-11-23 09:03:10`:

Functions may act as arguments to other functions. The `WeekdayName` function takes a single date/time argument and returns the day of the week as a string. The formula below evaluates to `Tuesday`:

This formula evaluates to `7`:

)

For detailed descriptions of all available functions, see "Functions" in the Genesys eServices Field Codes Reference Manual.

Using Objects

All object/property pairs are also available in the `Variables` drop-down menu in the Knowledge Manager `Field Code Editor`.

Object properties can be of any data type. `Agent.FullName`, for example, is a string, but `Interaction.DateCreated` is a date/time.

The data type of an object property can even be another object. For example, `Contact.EmailAddresses` yields another object called a `ContactEmailAddressList`. In cases such as this, you can access the properties of the resulting object by entering a period (.), followed by the property name, just as before. For example, the formula below evaluates to the number of e-mail addresses assigned to the contact:

Some object properties require arguments just as functions do. For these properties, write the arguments, enclosed in parentheses after the property name, just as before.

For example, the ContactEmailAddressList object has a property named `Exists`, which you can use to test whether a particular e-mail address is assigned to a contact. The data type of this property is Boolean (true/false), and it takes one argument, the e-mail address to test. For example:

For detailed descriptions of all objects and their properties, see "Objects" in the Genesys eServices Field Codes Reference Manual.

Contents

-
- Administrator

Set up and manage the hours of operation for your business.

Related documentation:

-

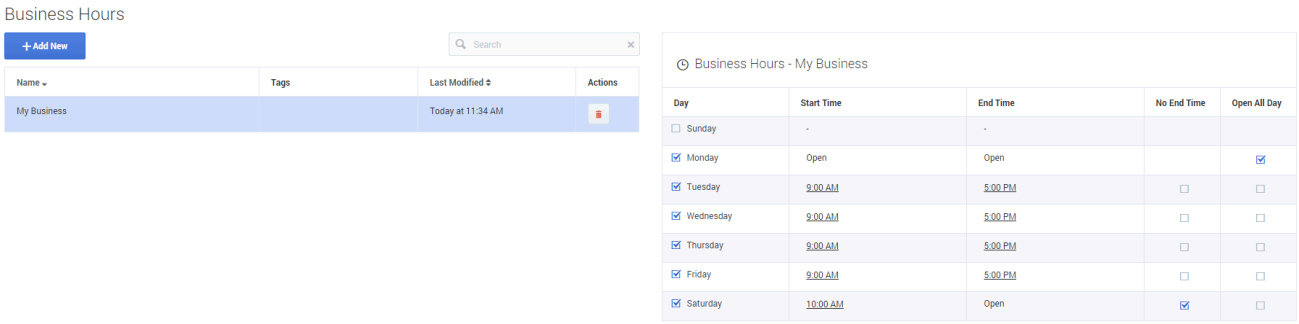
You can use the Business Hours feature to create various sets of business hours for use in your applications.

You must use a unique name for each set of business hours. You might want to use a combination of company names and departments. For example, you could use `CompanySales` and `CompanyService`.

It is also recommended to use tags to help organize your business hours. Once Business Hours are set, you can use them in your applications with a Business Hours block.

Specify open and closed times

Use the check boxes to indicate which days your business is open. Click the **Start Time** and **End Time** values to specify the opening and closing times, or use the **No End Time** or **Open All Day** options.



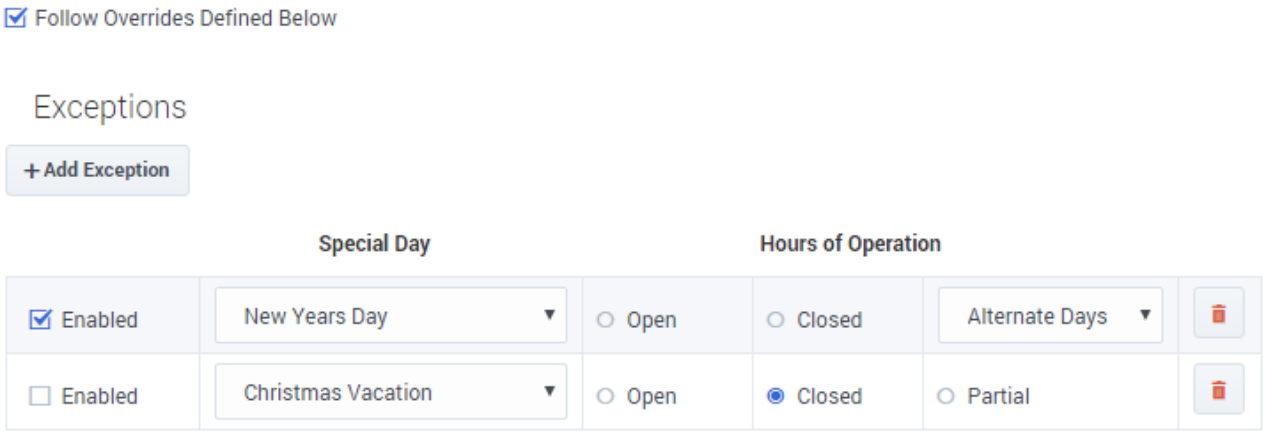
Important information for Callback

If you are using this business hours entry for Callback (i.e. in the Callback Settings Data Table), you must also specify the **Time Zone** setting. See Time Zone, below, for more information.

Specify exceptions

You might want to specify alternate business hours that differ from the normal Special Day hours. To specify alternate business hours for Special Days, select **Follow Overrides Defined Below** and add an exception.

Select a Special Day and choose the **Hours of Operation**.



When you add an exception (or override), it is enabled by default. You can clear the check box to disable it.

> **Important**
>
> - You must select a **Special Day** for each exception. Otherwise, the exceptions will not take effect.
> - Even if a **Special Day** exception is enabled, it won't take effect unless the **Follow Overrides Defined Below** option is selected.
> - If the current day is a Special Day and there are multiple matching exceptions defined for it, Designer uses only the first exception that matches the specified conditions and ignores any others.

Time Zone (for Callback only)

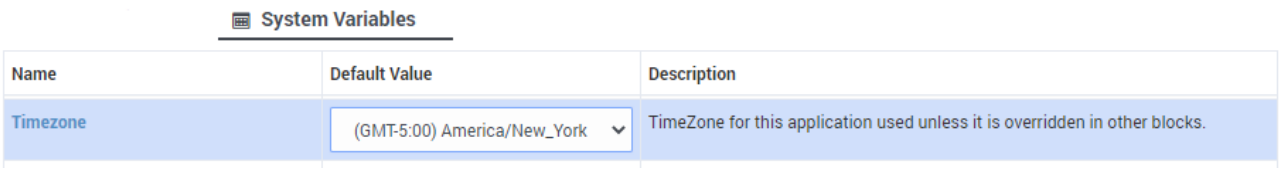
If you are using Callback, select the appropriate **Time Zone** to use for booking scheduled callbacks. This setting ensures that customers who request callbacks are offered time slots that correspond to the time zone of the business hours that are being used for callbacks, and not the time zone of the Designer application (as set in the system variables).

The **Time Zone** setting only applies when determining the available time slots for booking scheduled callbacks. It does not affect any other callback features or settings, such as determining if the business is open or closed for offering callbacks or when to initiate **Immediate Blackout**. These calculations are always based on the time zone that is specified in the System Variables of the

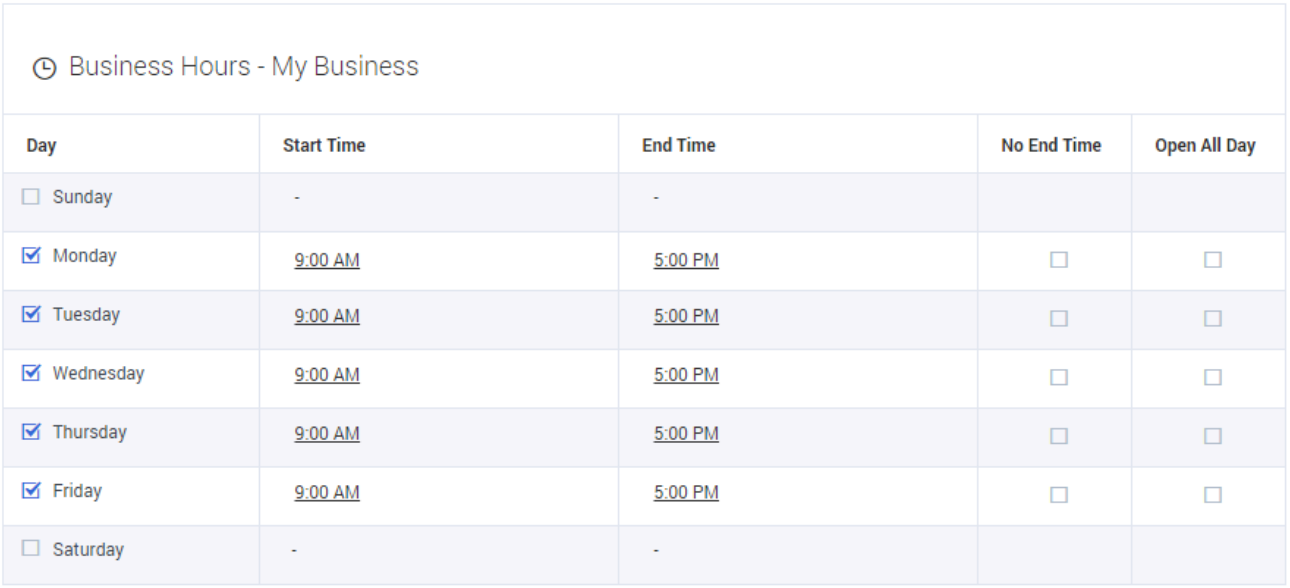
Designer application.

Example

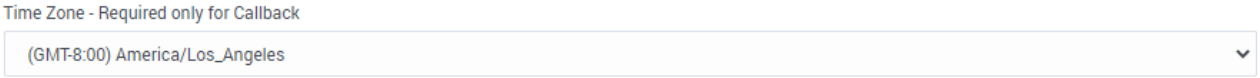
Let's say our application is running in the Eastern Daylight Time (EDT) zone (i.e. **America/New_York**), as specified in the **System Variables**:



However, our contact center is located in California and is open from Monday to Friday, 9 am to 5 pm (Pacific Daylight Time). In this case, we would specify the business hours accordingly:



Then, for the **Time Zone**, we would select **America/Los_Angeles** to use for booking scheduled callbacks. This setting will override the time zone that is currently set in the Designer application and be used for determining the available callback times:



If a customer calls and requests a callback, they are offered an available time slot that is based on the **America/Los_Angeles** time zone. All other features and settings continue to use the **America/New_York** time zone that was set for the application in the **System Variables**.

Important

When you make changes to Business Hours or Special Days, you must click **Save** for the changes to take effect. Click **Reset** to discard your changes without saving them or **Delete** to remove a saved entry.