



This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

## Composer Help

DB Input Block

# DB Input Block

## Contents

- **1 DB Input Block**
  - 1.1 DB Input Block Exception Events
  - 1.2 Name Property
  - 1.3 Block Notes Property
  - 1.4 Data Source Property
  - 1.5 Data Source Type Property
  - 1.6 Data Source Variables Property
  - 1.7 Exceptions Property
  - 1.8 Language Property
  - 1.9 Clear Buffer Property
  - 1.10 Interruptible Property
  - 1.11 Prompts Property
  - 1.12 Timeout Property
  - 1.13 Security Property
  - 1.14 Output Result Property
  - 1.15 Get Shadow Variables Property
  - 1.16 Number of Retries Allowed Property
  - 1.17 Retry Prompts Property
  - 1.18 Use Last Reprompt Indefinitely Property
  - 1.19 Use Original Prompts Property
  - 1.20 Use Single Counter for Nomatch And Noinput Property
  - 1.21 Condition Property
  - 1.22 Logging Details Property
  - 1.23 Log Level Property
  - 1.24 Enable Status Property

The DB Input block accepts a **DB Data** block as its data source and acts as an input field that accepts input based on a grammar created from the results returned from the database. It accepts DTMF or speech input. This block differs from the **Menu block** in that it enables taking input that might not belong to a simple choice list (as for the Menu block). The DB Input block can be used to collect numerical data; for example, phone numbers, account numbers, amounts, or speech data, such as a Stock name. It uses speech or DTMF grammars to define the allowable input values for the user responses. Built-in system grammars are available for data, such as dates and amount. The user input result will be stored in a block name variable in the VXML application. Note: If the DB Input block uses a **DB Data** block as its data source, it uses only the first column from returned results to generate the grammar. The DB Input block can also use a variable as a data source instead of a DB Data block. In this case, grammar for the input is generated based on data in the array. The variable should represent a JSON array similar to the sample below:  
`myVariable=[["Google"],["Apple"],["Motorola"],["Samsung"],["Nokia"]]` The DB Input block has the following properties:

## DB Input Block Exception Events

The DB Input block has four exception events as described in **Exception Event Descriptions**:

- error
- error.noresource
- noinput
- nomatch

## Name Property

Please find this property's details under **Common Properties**.

## Block Notes Property

Can be used for both callflow and workflow blocks to add comments.

## Data Source Property

The Data Source property allows you to select the DB Data block that contains a previously-defined database query. This is used when DBDataBlock is selected as the Data Source Type property value. The results of this database query will be used to create the input field. To select the data source (a DB Data block):

1. Select the **Data Source** row in the block's property table.

2. In the **Value** field, select the appropriate DB Data block from the drop-down list.

## Data Source Type Property

The Data Source Type property allows you to select whether your data source is the contents of a DB Data block or a variable. To select the data source type:

1. Select the **Data Source Type** row in the block's property table.
2. In the **Value** field, select DBDataBlock or Variable from the drop-down list.

## Data Source Variables Property

The Data Source Variables property allows you to select the contents of a **variable** as your data source. This is used when Variable is selected as the Data Source Type property value. To select the variable that serves as your data source:

1. Select the Data Source Variable row in the block's property table.
2. In the Value field, select one of the available **variables** from the drop-down list. This can also be a custom variable you assigned in the Entry block.

## Exceptions Property

Find this property's details under **Common Properties**.

## Language Property

The language set by this property overrides any language set by the **Set Language** block, the Project preferences, or the incoming call parameters. The property takes effect only for the duration of this block, and the language setting reverts back to its previous state after the block is done. In the case of the DB Input block, this property affects the language of grammars of TTS output:

1. Click under Value to display a down arrow.
2. Click the down arrow and select English - United States (en-US) or the variable that contains the language.

## Clear Buffer Property

Use the Clear Buffer property for clearing the DTMF digits in the key-ahead buffer. If it is not set to

true, the DTMF digits entered are carried forward to the next block. It is commonly used for applications that the caller is familiar with. For example, the caller hears a welcome prompt but knows the next prompt will solicit the caller's input or menu selection. The caller may start inputting with DTMF while the welcome prompt plays and expect the input to carry forward. To assign a value to the Clear Buffer property:

1. Select the **Clear Buffer** row in the block's property table.
2. In the **Value** field, select true or false from the drop-down list.

## Interruptible Property

This property specifies whether the caller can interrupt the prompt before it has finished playing. To assign a value to the Interruptible property:

1. Select the **Interruptible** row in the block's property table.
2. In the Value field, select true,false,or DTMF (for DTMF barge-in mode support) from the drop-down list.

## Prompts Property

Find this property's details under [Common Properties](#). Note: When Type is set to Value and Interpret-As is set to Audio, you can specify an HTTP or RTSP URL. When Type is set to Variable and Interpret-As is set to Audio, you can specify a variable that contains an HTTP or RTSP URL.

## Timeout Property

The Timeout property defines the length of the pause between when the voice application plays the last data in the list, and when it moves to the next block. To provide a timeout value:

1. Select the **Timeout** row in the block's property table.
2. In the **Value** field, type a timeout value, in seconds.

## Security Property

When the Security property is set to true, data for this block is treated as private. GVP will consider the data associated with this block as sensitive and will suppress it in platform logs and metrics. To assign a value to the Security property:

1. Select the Security row in the block's property table.
2. In the Value field, select true or false from the drop-down list.

### Output Result Property

You must use the Output Result property to assign the collected data to a user-defined **variable** for further processing.

- Note! This property is mandatory. You must select a variable for the output result even if you do not plan on using the variable. If this is not done, a validation error will be generated in the Problems view.
1. Select the Output Result row in the block's property table.
  2. In the Value field, click the down arrow and select a variable.

For more information, see [Upgrading Projects/Diagrams](#).

### Get Shadow Variables Property

Shadow variables provide a way to retrieve further information regarding the value of an input item. By setting this property to true, it will expose the block's shadow variable within the callflow. When enabled, the shadow variable will be included in the list of available variables. (For example, the Log block's Logging Details will show DBInput1\$.) A shadow variable is referenced as `blockname$.shadowVariable`, where `blockname` is the value of the input item's name attribute, and `shadowVariable` is the name of a specific shadow variable, for example: `DBInput1$.duration`. To assign a value to the Get Shadow Variables property:

1. Select the **Get Shadow Variables** row in the block's property table.
2. In the Value field, select true or false from the drop-down list.

### Number of Retries Allowed Property

The Number Of Retries Allowed property determines how many opportunities the user will be provided to re-enter the value. If Use Last Prompt Indefinitely is set to true, this property has no effect; otherwise, the `error.com.genesyslab.composer.toomanynomatches` or `error.com.genesyslab.composer.toomanynoinputs` errors will be raised on reaching the maximum retry limit. To provide a value for the number of retries allowed:

1. Select the Number Of Retries Allowed row in the block's property table.
2. In the Value field, type a value for the number of retries that will be allowed.

### Retry Prompts Property

Find this property's details under [Common Properties](#).

## Use Last Reprompt Indefinitely Property

If you set the Use Last Reprompt Indefinitely property to true, the application uses your last reprompt as the prompt for all further retries. In this case the NoMatch and NoInput exception handlers will never get executed, as the retry loop keeps executing forever. To assign a value to the Use Last Reprompt Indefinitely property:

1. Select the **Use Last Reprompt** Indefinitely row in the block's property table.
2. In the Value field, select true or false from the drop-down list.

## Use Original Prompts Property

If you set the Use Original Prompts property to true, in the event of an error requiring a retry, the application first plays back the retry error prompt, and then plays back the original prompt for the block (as specified in the Prompts property). To assign a value to the Use Original Prompts property:

1. Select the **Use Original Prompts** row in the block's property table.
2. In the Value field, select **true** or **false** from the drop-down list.

## Use Single Counter for Nomatch And Noinput Property

If you set the Use Single Counter For Nomatch And Noinput property to true, the application maintains a single combined counter for the nomatch and noinput errors. For example, if the block has three nomatch retry messages and three noinput retry messages, the user gets three retry attempts. If you do not select this option, the application generates a total of six retries; and the user gets up to six retry attempts while not exceeding three of each type -- noinput or nomatch. Note: This property not available on the Record block. To assign a value to the Use Single Counter For Nomatch And Noinput property:

1. Select the Use Single Counter For Nomatch And Noinput row in the block's property table.
2. In the Value field, select true or false from the drop-down list.

## Condition Property

Find this property's details under [Common Properties for Callflow Blocks](#) or [Common Properties for Workflow Blocks](#).

## Logging Details Property

Find this property's details under [Common Properties for Callflow Blocks](#) or [Common Properties for](#)

Workflow Blocks.

## Log Level Property

Find this property's details under [Common Properties for Callflow Blocks](#) or [Common Properties for Workflow Blocks](#).

## Enable Status Property

Find this property's details under [Common Properties for Callflow Blocks](#) or [Common Properties for Workflow Blocks](#).