Dialogs in The Sims have the following parameters and features:

- **Dialog Type**
  - **Message**

    The Message Dialog has a icon, tile, text, and one button, defaulting to "OK".

    The tree node has one return branch: "OK" returns true.

  - **Yes-No**

    The Yes-No Dialog has a icon, title, text, and two buttons, defaulting to "Yes" and "No".

    The tree node has two return branches: "Yes" returns true, "No" returns false.

  - **Yes-No-Cancel**

    The Yes-No-Cancel Dialog has a icon, title, text, and three button, defaulting to "Yes", "No" and "Cancel".

    The tree node has two return branches and returns a value in temp 0: "Yes" returns true, "No" returns false and 0 in temp 0, "Cancel" return false and 1 in temp 0.

  - **Text Entry**

    The Text Entry Dialog has a icon, title, text, single line input field, and one button, defaulting to "OK".

    The tree node has one return branch: "OK" returns true.

    The entered string is stashed away in a global variable, which is used to initialize the first name of the next new baby Sim who's conceived. Unfortunately text entry dialogs aren't useful for much else than naming new Sims. But they can also be used as a placibo, if you want to have the user enter a string that's ignored (like a suggestion box).

  - **Tutorial**

    The Tutorial Dialog has a icon, tile, text, and one button, defaulting to "OK".

The tree node has one return branch: "OK" returns true.

- **Behavior**
  - **Engage and Block Sim**
  - **Return and Block Sim**
  - **Engage and Continue Sim**
  - **Return and Continue Sim**
- **Icon**
  - **Automatic**

    Automatically generates icon based on the object id in the stack object.

  - **None**

    No icon.

  - **Neighbor**

    Automatically generates icon based on the neighbor id in the stack object.

  - **Indexed**

    Uses indexed bitmap based on resource id 5000 plus index parameter.

  - **Named**

    Uses icon named in icon name of dialog strings.

- **Local Reference**

  Selects the stack local variable to be used for local references during dialog string substitution (described below).

- **Text**

  Each dialog has a set of strings, which use defaults if not explicitly assigned.

  - **Message Text**

    The Message Text is displayed in the dialog as a message to the user.

- **Yes Button Text**

  The Yes Button Text is displayed in the dialog Yes button.

- **No Button Text**

  The No Button Text is displayed in the dialog No button.

- **Title Text**

  The Title Text is displayed in the dialog title.

- **Cancel Button Text**

  The Cancel Button Text is displayed in the dialog Yes button.

- **Icon Text**

  The Icon Text is the name of the icon to display.

The dialog strings undergo parameter substitution before being shown to the user. Expressions start with a dollar sign followed by a keyword. Some of the expressions take one or two numeric parameters. Parameters are numbers that follow the keyword and are separated by colons. The following expressions are supported.

- **$FamilyAssets:localIndex**

  Replaced by the number of simoleans owned by the family whose family id is given in the localIndex specified in the keyword parameter.

- **$Family**

  Replaced by the last name of the Sim whose neighbor id is given by the stack object.

  Seems to return the family number in stack local variable 1. Note: This does not work with UserEvent, which requires the stack object to be an object id instead of a neighbor id.

- **$Neighbor**

  Replaced by the first (catalog) name of the Sim whose neighbor id is given by the stack object.

Seems to return the neighbor id in stack local variable 0. Note: This does not work with UserEvent, which requires the stack object to be an object id instead of a neighbor id.

- **$Me**

  Replaced by the first (catalog) name of me (the selected person).

  Note: This does work with UserEvent!

- **$Object**

  Replaced by the first (catalog) name of the stack object id.

  Note: This does work with UserEvent!

- **$Local:localIndex**

  Replaced by numeric value given in the localIndex specified in the keyword parameter.

- **$Grade:localIndex**

  Replaced by grade given in the localIndex specified in the keyword parameter.

- **$TimeLocal:localIndex**

  Replaced by the formatted time (in hours) in the localIndex specified in the keyword parameter.

- **$JobOffer:local1Index:local2Index**

  Replaced by the job offer string. If two localIndexes are given, uses the career id and job number from the given locals. If two localIndexes aren't given, uses the current object's career id and job number.

- **$JobDesc:local1Index:local2Index**

  Replaced by the job description string. If two localIndexes are given, uses the career id and job number from the given locals. If two localIndexes aren't given, uses the current object's career id and job number.

- **$Job:local1Index:local2Index**

  Replaced by the job name string. If two localIndexes are given, uses the career id and job number from the given locals. If two localIndexes aren't given, uses the current object's career id and job number.

- **Buttons**