



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

Callflow Blocks

# Callflow Blocks

## Contents

- [1 Callflow Blocks](#)
  - [1.1 VXML Properties](#)
  - [1.2 Main Versus Subcallflow](#)
  - [1.3 Subcallflows](#)

A block is the fundamental element of a callflow. Each block defines specific properties and how to handle specific events. You use the **Link tools** to connect these blocks in the order that the application should follow. A single VXML application is generated per callflow. Each block in a callflow becomes a form in the generated VXML document.

## VXML Properties

Each block has custom VoiceXML properties. These properties appear within a Properties view at the bottom of the Composer window when you right-click the block and then select Show Properties View from the shortcut menu. For each block, specific properties determine how events are handled. There are several categories of properties depending on the specific block. The blocks build a callflow or subcallflow. Generate code either from the Toolbar or from the Diagram menu. Static VXML pages (pure VXML code) are generated in the src-gen folder.

## Main Versus Subcallflow

There are two types of callflows:

- **Main Callflow:** This is the starting callflow for any application.
- **Subcallflow:** This is a component callflow that can be called from the main callflow or another subcallflow.

Each main callflow or subcallflow application should have at least three blocks:

- The **Entry block** to start the application. This block also specifies the relative file locations of the **audio files** for the generated application code and default exception handling.
- At least one other block to perform specific functions such as passing a call to an agent, creating a log of an activity, requesting caller input, playing a prompt, and so on.
- The Exit block to end the application, or, for example, the **GoTo block** to direct the application to another application.

## Subcallflows

Subcallflows are used for modularizing applications and for writing components that can be reused by multiple applications (such as a credit card validation subcallflow). The usage of subcallflows within a main callflow is very similar to a function call in a programming language. One or more input parameters can be passed to a subcallflow. Similarly, the subcallflow can return one or more output parameters. Therefore, a subcallflow can be designed to behave differently depending on the input parameter(s) passed.