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.

Genesys Administrator Extension Developer's Guide

JavaScript Usage

5/8/2025

JavaScript Usage

Within SPD files, JavaScript performs additional processing and manipulation of data. This includes the use of built-in JavaScript functions, as well as the ability to define and call your own JavaScript functions. All facilities of the JavaScript language are available to the SPD writer.

All <input> elements have a **name** attribute. The specified name defines a JavaScript variable that contains the input value that is selected by the user. If the **type** that is specified in the <input> element is `selectOne` or `selectMultiple`, then the variable contains the JSON representation of a single configuration object (in the case of `selectOne`) or a JavaScript array, where each array element is the JSON representation of a configuration object (in the case of `selectMultiple`).

All action elements (for example: `cfg:*`, `fileio:*`, `os:*` and `wfm:*`) have a **result** attribute that specifies the name of a JavaScript variable that stores the result of the action. Further details about the content of the variable can be found in the details section of each action element.

Within the SPD file, the <script> tag can contain valid JavaScript. The JavaScript variables that are defined by the <input> elements and action elements can be accessed and manipulated in these sections.

Some SPD XML elements have attributes named **cond** or **expr**. These attributes must contain JavaScript expressions. A JavaScript expression could be a JavaScript variable, a function call that returns a value, or an actual inline JavaScript expression. If the value evaluates to a "falsy" condition (for example: `0`, `""`, `false`, `undefined`, `null`, or `NaN`), then this element is not shown or executed.

In the following conditions, use of JavaScript (expressions or variables) is possible by placing "`${}`" around the JavaScript expression or variable:

1. attributes that are not named **cond** or **expr**
2. within XML elements (besides the <script> tag)

The following is an example: <deploymentkey>${TENANT}</deploymentkey>, where TENANT is a JavaScript variable.