# Scaffold Decision Tree Data Structure

Copyright (c) 2020. Ascendant Design and Training, LLC.  
Last updated by Daniel Patterson, Saturday, October 24, 2020

The decision tree view is a node editor used for creating branching logic scenarios that illustrate a series of states, actions, or events.

Each block on the display acts as a node on the chain, the main text is currently referred to as the question, the connecting points are referred to as input and output sockets, and each output socket can be associated with text, which is currently referred to as an answer.

## Object Model

Following is the general object model of this file. See each section for detailed descriptions.

* NodeFile.
  + Node Collection.
    - Node Object.
      * Node Properties.
      * Socket Collection.
        + Socket Object.

Socket Properties.

* + Resource Collection.
    - Resource Object.
      * Resource Properties.

The data underlying the visual layout is strictly structured, adhering to the following specification.

## NodeFile Object

The entire file is a single object with the following properties.

* Node Collection. General structure of nodes, sockets, connections, and properties.
* Resource Collection. Single-instance storage of loaded media and other resource assets. Each resource entry is identified with a GUID Ticket property, a reference link, and optional Data URI for support of embedded data.

## Node Collection

The entire file content is a collection of node objects. There are no properties on this collection.

## Node Object

The node object is the first layer at which details are defined. The node object is comprised of the following properties and collections.

### Node Properties

Each of the node has a number of well-defined properties. Notice that in addition to the base level of properties listed for the node, there are permanent property definitions defined for the same node in the Properties collection. The difference between node-level properties and permanent properties defined in the Properties collection has to do with whether the properties have been defined on the programming object named NodeItem.

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| Delay | float | Time to delay, in seconds, when the node is set to NodeType.Delay. |
| Height | float | The height of the node, in pixels. |
| NodeColor | color | Background color of the node. |
| NodeTextColor | color | Text color of the node. |
| NodeType | string | Type of Node. At present, following are the defined node types:   * Start. The node has output sockets, but no input sockets. * Fork. The node has input and output sockets. * Delay. Pause for a specified delay then continue. * Termination. The node has only an input socket. |
| Properties | Property Collection | Collection of Name/Value pairs describing an infinite possible number of attributes of the node. |
| Sockets | Socket Collection | Collection of socket objects. |
| Ticket | guid | Globally unique identification of this node. |
| TitleHeight | float | Height of the title section of the node. |
| TitleProperty | string | The property name to be used as the title of the node. |
| Width | float | The width of this node, in pixels. |
| X | float | The horizontal coordinate of the left side of the node from the left side of the editor canvas. |
| Y | float | The vertical coordinate of the top side of the node from the top side of the editor canvas. |
| ZOrder | int | The placement of this node in front of or behind other nodes in the editor. |

## Socket Collection

The socket collection is represented as an array of Socket Objects.

## Socket Object

The socket object defines the details that describe a single incoming or outgoing aspect of a node.

### Socket Properties

Each socket has a number of well-defined properties. Notice in addition to there being base properties on the Socket object, you will also find permanently defined properties on the Socket.Properties collection. The difference between whether a property is considered to be base or extended, in essence, is directly determined by whether or not a corresponding property is represented in the programming object named SocketItem.

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| Connections | guid[] | Array of socket tickets to which this socket is connected. |
| Height | float | Height of the socket symbol, in pixels. |
| Properties | Property Collection | Collection of Name/Value pairs describing an infinite possible number of attributes of the socket. |
| SocketMode | string | The operational mode of the socket. In this version, the choices are Input, Output, and None. |
| TextHeight | float | Height of the text, in pixels. |
| TextWidth | float | Width of the text, in pixels. |
| TextX | float | Horizontal coordinate of the text for this socket, relative to the parent node. |
| TextY | float | Vertical coordinate of the text for this socket, relative to the parent node. |
| Ticket | guid | The globally unique identification of this socket. |
| TitleProperty | string | The local property used as the display text for this socket. |
| Width | float | Width of the socket symbol, in pixels. |
| X | float | The horizontal coordinate of the displayed location of the socket symbol, relative to the parent node. |
| Y | float | The vertical coordinate of the displayed location of the socket symbol, relative to the parent node. |

## Property Collection

The property collection contains a list of Name / Value pairs that can describe any kind of configuration.

A number of permanent properties are defined for both nodes and sockets, but any number of additional non-permanent properties can be defined for end-user applications.

## Property Object

The property object has two properties, Name and Value. The Name indicates the name of the entity, while the Value provides its value.

## Permanent Node Properties

The permanently defined properties on a node are the following.

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| Question | string | The default property used for the main text title for the node. |
| StoryColorFill | color | The color used to fill the text shape for this node on the storyboard. |
| StoryColorOutline | color | The color used to outline the text shape for this node on the storyboard. |
| StoryColorText | color | The text color used when printing the text for this node on the storyboard. |
| StoryFontName | string | Name of the font used to print node text on the storyboard. |
| StoryFontSize | float | Font size, in points, for node text printed on the storyboard. |
| StoryPageHorizontalPlacement | string | Horizontal justification from page edge. Choices are from 'Left' and from 'Right'. Default is 'Left'. |
| StoryPageNumber | int | The page number of the corresponding storyboard slide. |
| StoryPageVerticalPlacement | string | Vertical justification from page edge. Choices are from 'Top' and from 'Bottom'. Default is 'Top'. |
| StoryPageWidth | float | Maximum width of the caption container on the storyboard. Default = 512. |
| StoryPageX | float | The horizontal location of this node on the storyboard. |
| StoryPageY | float | The vertical location of this node on the storyboard. |
| StoryShapeType | string | Name of the shape to use while displaying this node on the storyboard. The following types are currently defined.   * Caption. The node text is displayed as a caption bubble. * Rectangle. The node text is displayed as a normal rectangle. |

## Permanent Socket Properties

Following are the permanently defined socket properties.

| **Name** | **Type** | **Description** |
| --- | --- | --- |
| Answer | string | The default property used for text about this socket. Similar in use to the node's Question property. |
| Index | string | Sorting index of this socket, relative to all other sockets on the node. |
| StoryColorFill | color | The color used to fill the text shape for this socket on the storyboard. |
| StoryColorOutline | color | The color used to outline the text shape for this socket on the storyboard. |
| StoryColorText | color | The text color used when printing the text for this socket on the storyboard. |
| StoryFontName | string | Name of the font used to print socket text on the storyboard. |
| StoryFontSize | float | Font size, in points, for socket text printed on the storyboard. |
| StoryPageHorizontalPlacement | string | Horizontal justification from page edge. Choices are from 'Left' and from 'Right'. Default is 'Right'. |
| StoryPageNumber | int | The page number of the corresponding storyboard slide. |
| StoryPageVerticalPlacement | string | Vertical justification from page edge. Choices are from 'Top' and from 'Bottom'. Default is 'Top'. |
| StoryPageWidth | float | Maximum width of the caption container for this socket. Default = 256. |
| StoryPageX | float | The horizontal location of this socket on the storyboard. |
| StoryPageY | float | The vertical location of this socket on the storyboard. |
| StoryShapeType | string | Name of the shape to use while displaying this socket on the storyboard. The following types are currently defined.   * Caption. The node text is displayed as a caption bubble. * Rectangle. The node text is displayed as a normal rectangle. |

## Resource Collection

The Resource collection of a node file contains zero or more Resource objects.

## Resource Object

The Resource object is used to describe a single instance resource asset that has been loaded for use of any other object in the file. As the term implies, there is only one copy per file defined of a single instance resource, yet any number of other objects in the file can reference or use it.

In this version, any single instance object can be represented by a link to the resource data or can be loaded with base64 data to provide Data URI capabilities for data embedded directly in the file.

When using linked information, the object Uri follows the HTML5 HREF syntax.

### Resource Properties

Similarly to the node and socket objects, the Resource object is extensible through the presence of an abstract Properties collection that allows an indefinite number of properties to be defined by the end user for any reason.

The resource also has a number of built-in properties on the object itself. Following is the listing of those built-in properties.

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| AbsoluteFilename | string | The full path and filename of the resource file as it was originally loaded into the file. |
| Properties | Property Collection | Collection of abstract properties for this resource. |
| RelativeFilename | string | The relative filename to the loaded node data file, if feasible. Otherwise, blank. |
| Ticket | string | The globally unique identification of this resource. |
| Uri | string | The active URI. If Data URI, the digital content of the data will be embedded in this field. Otherwise, if this resource is accessed by a link, the standard HTML HREF syntax will be used. |

## Node And Socket Media Support

Nodes and sockets both have support for multiple types of media and support for basic interaction. In general, nodes and each of a node's sockets have support for the following elements.

These additions allow the node and its sockets to present a self-contained card-like personality during runtime that has multiple aspects available for interactive purposes. In industry terms, this effect produces that of the Hero Card.

|  |  |  |
| --- | --- | --- |
| **General Element** | **Property Name** | **Description** |
| Audio | MediaAudio | On nodes, this is a URI referring to an audio to play when the node is loaded. On sockets, the URI is played when the option is selected and before the connection is followed to the next node. |
| Image | MediaImage | A URI referring to an image to display in the card. base64 data URI is allowed. |
| Link | MediaLink | A URL the user can click to view more information about the current context. Standard href link syntax. |
| Video | MediaVideo | On nodes, this is a URI referring to a video to play when the node is loaded. On sockets, the URI is played when the option is selected, and before the connection is followed to the next node. |

Notice that in this version, if audio and video URIs are both specified, the video URI will take precedence, and the audio will only be utilized on non-video systems that provide audio support, such as a telephone system or radio, etc.