

SRP_JsonX

Adds elements to the current document based on the current path.

Syntax

```
// Adding a member to the current object
SRP_JsonX(MemberName, Value, Hint)

// Adding a value to the current array
SRP_JsonX(Value, Hint)

// Starting a new object or array in the current object
SRP_JsonX(MemberName, Opener)

// Starting a new object or array in the current array
SRP_JsonX(Opener)

// If closing an object or array
SRP_JsonX(Closer)
```

Returns

1 if successful, 0 if there was an error.

Parameters

Parameter	Description
MemberName	The name of the new object member.
Value	The new value. <i>Omit to pass null.</i>
Hint	Forces the value into a json type. Can be "Bool", "Null", "String", or "Array". Optional.
Opener	Starts a new object or array. Can be "{" or "[".
Closer	Closes the current element. Can be "}" or "]".

Remarks

SRP_JsonX is a dynamic routine that interprets its parameter based on the current state of the document, and calling it can even change the state. The actual parameter names are strange, so the SRP Editor tooltip won't always be of use. There are three general use cases:

Use Case	Syntax	Behavior
Adding Member to Object	SRP_JsonX (MemberName, Value, Hint)	Adds or replaces the member of the current object with the given value. If you omit the value, then you set the member to null.
Adding Value to Array	SRP_JsonX(Value, Hint)	Adds the given value to the end of the current array. If you omit the value, then you add null to the end of the array.
Starting Object /Array in Object	SRP_JsonX (MemberName, Opener)	Adds or replaces the member of the current object with an empty object or array. The opener must be "{" to start a new object or "[" to start a new array. The new object/array becomes the current element.
Starting Object /Array in Array	SRP_JsonX (Opener)	Adds an empty object or array to the end of the current array. The opener must be "{" to start a new object or "[" to start a new array. The new object/array becomes the current element.
Closing	SRP_JsonX(Closer)	Passing "}" or "]" closes the current object or array and makes it's parent the new current element.

The Value parameter will be smartly interpreted by the routine unless a Hint is specified. OI numbers are converted into json numbers. You can also pass json as a value, and this routine will parse the json and set it as the current element's value. Note. You can even pass partial json. For example, let's say every object has a "type" member, you can open a new object with the member in one call, leaving the object open for further calls to SRP_JsonX:

```
// we are currently in an array of employees, let's add a manager
// notice how we opened an object with a "type" member and left it open so we could add other members later
SRP_JsonX('{ "type": "manager" }')
    SRP_JsonX('firstname', 'Sarah')
    SRP_JsonX('lastname', 'Marshall')
    SRP_JsonX('age', 32)
SRP_JsonX('}')'
```

The Hint parameter can be used to force the value into a specific json type. Use "Bool" to force the value to be true or false. Use "Null" to ignore the given value and just use null as the value. Use "String" to force the value into a string, which is useful when you want an OI number to be surrounded by quotes in the json itself.

New in 2.2.4. The Hint supports the "Array" value. When you set the hint to "Array", SRP_JsonX assumes you are passing a delimited array as the value. It will parse the array into individual elements and produce a Json array as a value. You don't have to tell SRP_JsonX what delimiter your array is using. The "Array" hint tells it to use any OI delimiter it finds (that is, @FM, @VM, @TM, etc.). You can also combine the "Array" hint with any of the above hints as well, such as "String Array" or "Bool Array". This hint is meant as a quick and convenient way to convert 1-dimensional OI delimited arrays into Json arrays in a single call. It won't work for every scenario.

If you want to make simple changes to a document without affecting the state of that document, use [SRP_JsonX_Set](#) instead.

Examples

```
$insert SRPJSONX

SRP_JsonX_Begin('MyDocument', '{')
    SRP_JsonX('employees', '[')
        SRP_JsonX('{')
            SRP_JsonX('firstname', 'John')
            SRP_JsonX('lastname', 'Doe')
            SRP_JsonX('age', 21)
        SRP_JsonX('}')
        SRP_JsonX('{')
            SRP_JsonX('firstname', 'Anna')
            SRP_JsonX('lastname', 'Smith')
            SRP_JsonX('age', 32)
        SRP_JsonX('}')
        SRP_JsonX('{ "firstname": "Peter", "lastname": "Jones", "age": 43 }')
    SRP_JsonX(']')
    SRP_JsonX('count', 4)
    SRP_JsonX('active', 1, 'Bool')
    SRP_JsonX('alwaysnull')
    SRP_JsonX('alwaysstring', 4.321, 'String')
Json = SRP_JsonX_End('Pretty')
```