SRP_Json Parse

Parses a standardized JSON string into a new JSON Entity.

Syntax

```
Result = SRP_Json(Handle, "Parse", JsonString, ReportErrors = 0)
```

Returns

Returns "" if successful or an error message if not.

Parameters

Parameter	Description
Handle *[OUT]*	Variable to receive new handle to a JSON Entity. **Required.**
JsonString	The JSON to parse. **Required.**
ReportErrors	0 to return only success/failure; 1 to return a list of parsing errors. *Optional.* **Added in SRP Utilities 2.0.5**

Remarks

The PARSE service parses a string containing standardized JSON and produces a single JSON Entity Handle. If the JSON cannot be parsed, then this service returns a generic error message indicating that the parsing failed. If it succeeds, then "" is returns. The new JSON Entity Handle is placed into the variable you pass into the Handle parameter. It is perfectly safe to set Handle to an unassigned variable, but you should check the return value to see if the parsing was successfully, like so:

```
Result = SRP_Json(ObjectHandle, "Parse", SampleJSON)
If Result NE "" then
   Msg(@Window, Result:@FM:@FM:@FM:"!")
end
```

The new JSON entity's type depends upon the nature of the JSON. Here is some sample code testing each entity's type after parsing some JSON:

```
If SRP_Json(ObjectHandle, "Parse", '{"id": "Open"}') EQ "" then
   Type = SRP_Json(ObjectHandle, "TYPE") ; // Type is "Object"
   SRP_Json(ObjectHandle, "RELEASE")
end

If SRP_Json(ObjectHandle, "Parse", '["1", "2", "3"]') EQ "" then
   Type = SRP_Json(ObjectHandle, "TYPE") ; // Type is "Array"
   SRP_Json(ObjectHandle, "RELEASE")
end

If SRP_Json(ObjectHandle, "Parse", '"Hello, World!"') EQ "" then
   Type = SRP_Json(ObjectHandle, "TYPE") ; // Type is "String"
   SRP_Json(ObjectHandle, "RELEASE")
end

If SRP_Json(ObjectHandle, "Parse", '1') EQ "" then
   Type = SRP_Json(ObjectHandle, "TYPE") ; // Type is "Number"
   SRP_Json(ObjectHandle, "RELEASE")
end

If SRP_Json(ObjectHandle, "Parse", 'false') EQ "" then
   Type = SRP_Json(ObjectHandle, "TYPE") ; // Type is "Boolean"
   SRP_Json(ObjectHandle, "RELEASE")
end
```

IMPORTANT: Any JSON entity created by this service must be deallocated from memory when no longer needed using the Release service. Forgetting to do this on occasion will not cause a fatal error since SRP Utilities will clean up all its memory when OpenInsight closes, but frequently failing to release entities causes memory to get used up, which can become a problem over long periods of time.

Reporting Errors

Normally, the parser is very forgiving. It will still parse even if there are minor errors. However, there might be times when you want to know all errors, even if the parser is able to succeed. In this case, set the ReportErrors parameter to 1 to enable this feature. When enabled, the service will return an @FM delimited list of parsing errors, or "" if there were no errors.

Example

```
// Test parsing an existing JSON string and then drilling down into specific values
SampleJSON  = '{"menu": {'
SampleJSON := '    "header": "SVG Viewer",'
SampleJSON := '    "items": ['
SampleJSON := '        {"id": "Open"},'
SampleJSON := '        {"id": "OpenNew", "label": "Open New"},'
SampleJSON := '        null,'
SampleJSON := '        {"id": "ZoomIn", "label": "Zoom In"},'
SampleJSON := '        {"id": "ZoomOut", "label": "Zoom Out"},'
SampleJSON := '        {"id": "OriginalView", "label": "Original View"},'
SampleJSON := '        null,'
SampleJSON := '        {"id": "Quality"},'
SampleJSON := '        {"id": "Pause"},'
SampleJSON := '        {"id": "Mute"},'
SampleJSON := '        null,'
SampleJSON := '        {"id": "Find", "label": "Find..."},'
SampleJSON := '        {"id": "FindAgain", "label": "Find Again"},'
SampleJSON := '        {"id": "Copy"},'
SampleJSON := '        {"id": "CopyAgain", "label": "Copy Again"},'
SampleJSON := '        {"id": "CopySVG", "label": "Copy SVG"},'
SampleJSON := '        {"id": "ViewSVG", "label": "View SVG"},'
SampleJSON := '        {"id": "ViewSource", "label": "View Source"},'
SampleJSON := '        {"id": "SaveAs", "label": "Save As"},'
SampleJSON := '        null,'
SampleJSON := '        {"id": "Help"},'
SampleJSON := '        {"id": "About", "label": "About Adobe CVG Viewer..."}'
SampleJSON := '    ],'
SampleJSON := '    "testbool": "true",'
SampleJSON := '    "testint": "1234567890",'
SampleJSON := '    "testreal": "1234567890.987654321"'
SampleJSON := '}}'

// Parse it
If SRP_Json(ObjectHandle, "Parse", SampleJSON) EQ "" then

   // Grab some values
   TestValue = SRP_Json(ObjectHandle, "GetValue", "menu.items[1].id")
   TestValue = SRP_Json(ObjectHandle, "GetValue", "menu.items[12].label")
   TestValue = SRP_Json(ObjectHandle, "GetValue", "menu.items[3]")
   TestValue = SRP_Json(ObjectHandle, "GetValue", "menu.testbool")
   TestValue = SRP_Json(ObjectHandle, "GetValue", "menu.testint")
   TestValue = SRP_Json(ObjectHandle, "GetValue", "menu.testreal")

   // Release the object because we're all done
   SRP_Json(ObjectHandle, "Release")

end
```

See Also

New, Release