

SafeArray UDF - Advanced COM SafeArray Structure Manipulation

Overview

The SafeArray UDF is a comprehensive, standalone library for creating, managing, and interacting with COM SafeArray and VARIANT structures in Autolt. This UDF provides native interface capabilities for working with COM data without external dependencies.

Key Features

- **Standalone** : No dependencies on AutoltObject.au3 or Variant.au3
- **Comprehensive** : 40+ functions covering all SafeArray aspects
- **Multi-dimensional** : Support for up to 5-dimensional arrays
- **Bi-directional conversion** : Autolt Array ↔ SafeArray seamless conversion
- **Type-safe** : Complete VARIANT type handling
- **Microsoft compliant** : Implementation based on official oleaut32.dll API

Core Functionality

Creation and Destruction

- `_SafeArrayCreate()` / `_SafeArrayCreateEx()` / `_SafeArrayCreateVector()`
- `_SafeArrayDestroy()` / `_SafeArrayDestroyData()`
- `_CreateSafeArrayBounds()` for dimension definition

Data Manipulation

- `_SafeArrayPutElement()` / `_SafeArrayGetElement()`
- `_SafeArrayAccessData()` / `_SafeArrayUnaccessData()`
- `_SafeArrayLock()` / `_SafeArrayUnlock()` for thread synchronization

Conversion Functions

- `_SafeArrayFromAutoItArray()` : Autolt Array → SafeArray
- `_AutoItArrayFromSafeArray()` : SafeArray → Autolt Array
- Preserves Autolt dimension ordering

Information and Metadata

- `_SafeArrayGetDim()` / `_SafeArrayGetLBound()` / `_SafeArrayGetUBound()`
- `_SafeArrayGetVartype()` / `_SafeArrayGetElementSize()`

VARIANT and BSTR Management

- `_VariantInit()` / `_VariantClear()` / `_VariantSet()` / `_VariantRead()`
- `_SysAllocString()` / `_SysFreeString()` / `_SysReadString()`

Use Cases

- **COM Interoperability** : Data exchange with complex COM objects
- **Office Applications** : Excel, Word array manipulation
- **System APIs** : Windows APIs returning SafeArray structures
- **Database Operations** : COM recordset processing
- **Component Development** : Creating COM-compatible DLLs/OCX



Usage Example

```
#include "SafeArray.au3"

; Create Autolt array
Local $aData[3][2] = [["A", 1], ["B", 2], ["C", 3]]

; Convert to SafeArray
Local $pSafeArray = _SafeArrayFromAutoltArray($aData)

; Use with COM object
$oExcel.Range("A1:B3").Value = $pSafeArray

; Convert back to Autolt
Local $aResult = _AutoltArrayFromSafeArray($pSafeArray)

; Cleanup
_SafeArrayDestroy($pSafeArray)
```



Technical Advantages

- **Performance** : Direct access to native Windows APIs
- **Memory Management** : Automatic and manual memory handling available

- **Robustness** : Complete error handling with HRESULT codes
- **Compatibility** : Full x86 and x64 support
- **Documentation** : Complete headers with usage examples

Supported Types

- All standard VARIANT types (VT_I4, VT_R8, VT_BSTR, etc.)
- Multi-dimensional arrays (VT_ARRAY | VT_VARIANT)
- COM Objects (VT_DISPATCH)
- Binary data (VT_UI1 arrays)
- Date/Time values (VT_DATE)

Documentation Standards

Each function follows Autolt documentation standards with:

- Complete syntax specification
- Parameter descriptions
- Return values and error codes
- Usage examples
- Microsoft documentation links

Target Audience

- Autolt developers working with COM automation
- Office application automation specialists
- Advanced system integration developers
- Enterprise solution architects

Why This UDF?

This UDF fills a critical gap in the Autolt ecosystem by providing complete, professional access to COM SafeArray structures, which are essential for:

- Advanced COM interoperability
- Enterprise-grade automation solutions
- Complex data exchange scenarios
- Professional Windows development

Professional Benefits

- **Reduces development time** for COM-intensive applications
- **Eliminates dependency issues** with self-contained implementation
- **Ensures compatibility** with Microsoft COM standards
- **Provides enterprise-ready** error handling and memory management

This comprehensive library empowers AutoIt developers to work seamlessly with COM technologies at a professional level, opening new possibilities for automation and system integration projects.