



## CtiOs Object

---

- [CtiOs Object, on page 1](#)
- [Methods, on page 1](#)

## CtiOs Object

All of the interface objects in the CTI OS Client Interface Library support some common features, such as the IsValid and GetValue methods. This chapter describes these common features.

The CCtiOsObject class is the common base class for the objects in the CTI OS client interface library. You implement it as follows:

- In C++: All interface objects (CAgent, CCall, CCtiOsSession, CSkillGroup) derive from the CtiOs object. Thus, all the interface methods described in this chapter are directly available in the C++ objects.
- In COM (VB and C++): The COM objects for Agent, Call, Session, and SkillGroup publish a subset of these methods (as appropriate for the language), and the underlying implementation of the objects uses the C++ CCtiOsObject class to provide these features.
- In Java: All CTI OS interface objects (Agent, Call, Session, and SkillGroup) derive from the CtiOs object. Thus, all the interface methods described in this chapter are directly available in the Java objects.
- In .NET: All interface objects (Agent, Call, Session, and SkillGroup) derive from the CtiOs object. Thus, all the interface methods described in this chapter are directly available on the .NET objects.

The CCtiOsObject provides basic services including:

- Dynamic management of the object properties
- Object lifetime control using a reference counting mechanism
- Run-time class information

## Methods

The following table lists the available CCtiOsObject class methods.

Table 1: CtiOsObject Class Methods

Method	Description
DumpProperties	Returns a string listing all of an object's properties' names and values.
GetAllProperties	Returns all of the object's properties as Args (name/value pairs).
GetElement	Returns the value of an element.
GetLastError	Returns the last error that occurred on the calling thread.
GetNumProperties	Returns the number of properties of an object.
GetPropertyName	Returns a property name in a string format.
GetPropertyType	Returns the data type of the specified property.
GetValue, GetValueInt, GetValueString, GetValueArray	Returns the value of a specified property.
IsValid	Checks to see if the property of an object is valid.

## DumpProperties

The DumpProperties method returns all the properties of the object. This method builds a string showing all the properties in the form “key1 = value1; key2 = value2;...”.

### Syntax

#### C++

```
string DumpProperties ()
```

#### COM

```
HRESULT DumpProperties (/*[out,retval]*/ BSTR* bstrValue)
```

#### VB

```
DumpProperties() As String
```

#### Java

```
String DumpProperties ()
```

#### .NET

```
System.String DumpProperties ()
```

### Parameters

bstrValue

The output parameter (return parameter in VB) containing a string listing the names and values of the object's properties.

**Return Value**

COM: Default HRESULT return value. For more information, see [CIL Coding Conventions](#)

All Others: The string listing the names of all the object's properties.

## GetAllProperties

The GetAllProperties method returns all the object's properties and their values. For the properties that are calls, agents, or skillgroups, their string UniqueObjectIDs are returned, not the objects themselves. To get the objects themselves use [GetObjectFromObjectID](#), explained in [Session Object](#)

**Syntax****C++**

```
bool GetAllProperties (Arguments** arguments)
```

**COM**

```
HRESULT GetAllProperties (/*[out]*/ IArguments** arguments, /*[out,retval]*/ VARIANT_BOOL*  
errorcode)
```

**VB**

```
GetAllProperties arguments As (CTIOSCLIENTLib.IArguments) As Bool
```

**Java, .NET**

```
Arguments GetAllProperties()
```

**Parameters**

C++, COM, VB: arguments

Output parameter in the form of an Arguments array that has all the property names and values of the object.  
errorcode

An output parameter (return parameter in VB) that contains a boolean indicating success or lack thereof.

**Return Value**

C++ , VB: True upon success and false upon failure.

COM: Always returns S\_OK. Use the errorcode parameter to determine success or failure of the method call.

.NET, Java: NULL if the value requested is not found or if there is an error. If the method succeeds, it returns a reference to an Arguments object containing all the properties of the object.

## GetElement

Given a property of type Arguments whose name is specified by the key parameter, the GetElement method returns the Arg at position element of that Arguments array.

**Syntax****C++**

```
Arg& GetElement (string& key, int element)
Arg& GetElement (int key, int element)
Arg& GetElement (char* key, int element)
```

**COM**

```
HRESULT GetElement /*[in]*/ VARIANT* key, /*[in]*/ int element, /*[out,retval]*/ IArg**
pIArg)
```

**VB**

```
GetElement (key As VARIANT) As CTIOSCLIENTLib.IArg
```

**Java**

```
Arg GetElement(String key, int element)
Arg GetElement(int key, int element)
```

**.NET**

```
System.Boolean GetElement(System.String key, int element, out arg rArg)
```

**Parameters**

key

A key designating the name of the Arguments property whose element you want.

element

The integer index of the element to retrieve from the property key.

COM, VB:pIArg

An output parameter (return parameter in VB) containing an IArg with the value of the desired element.

.NET: rArg

An output parameter containing the value of the specified element. This parameter is null if the element is not found.

**Return Value**

An Arg reference containing the value of the desired element.

The C++ and Java versions of this method return NULL if an error occurs, such as the key or element is not found. The .NET version of this method returns true upon success and false upon error.

## GetLastError (Java and .NET Only)

The GetLastError method returns the last error that occurred on the calling thread.

**Syntax****Java**

```
Integer GetLastError()
```

**.NET**

```
System.Boolean GetLastError(out System.Int32 nLastError)
```

### Parameters

#### Java

None.

#### .NET

nLastError

Output parameter that is a 32-bit signed integer that contains the returned value of the last error.

### Returns

Java: An Integer object containing the error, or null if the object is not found or if there is an error.

.NET: The Boolean value true if the value is successfully set; otherwise false.

### Remarks

The following example code gets the last error on the current thread and logs the error message. If GetLastError fails, the code writes a warning message to the log file:

```
// First get the last error System.Int32 myLastError;
bool success = GetLastError(out myLastError);
if (!success)
{
    // log a message indicating that GetLastError failed
}
else
{
    //log a message that indicates what the last error was
LOGBYID(Cisco.CtiOs.Cil.TraceLevel.WARN, "GetLastError returned
    last error" = + myLastError);
}
```

## GetNumProperties

The GetNumProperties method returns the number of properties in the current object.

### Syntax

#### C++

```
int GetNumProperties ()
```

#### COM

```
HRESULT GetNumProperties (/*[out,retval]*/ int * num
```

#### VB

```
GetNumProperties () As Long
```

#### Java, .NET

```
int GetNumProperties()
```

**Parameters****num**

In the COM version, an output parameter (return value in VB, C++, Java, and .NET) that contains an integer that is the number of properties in the object.

**Return Value**

COM: Default CTI OS return values. See [CIL Coding Conventions](#)

All Others: An integer that is the number of properties currently a part of the object.

## GetPropertyName

The GetPropertyName method returns the name of a property in a string format.

**Syntax****C++**

```
string GetPropertyName (int index)
```

**COM**

```
HRESULT GetPropertyName (/* [in] index, /*[out,retval]*/ name)
```

**VB**

```
GetPropertyName (index As Integer) As String
```

**Java**

```
string GetPropertyName (int iIndex)
```

**.NET**

```
virtual System.Boolean GetPropertyName(int iIndex, out System.String name)
```

**Parameters****index**

An integer parameter specifying the index number of the requested property.

**name**

A string output parameter (return value in C++, VB, and Java) containing the property's name.

**Return Value**

COM: Default CTI OS return values. For more information, see [CIL Coding Conventions](#)

.NET: Boolean value set to true if the method call succeeds, otherwise false.

All Others: A string that contains the property's name.

## GetPropertyType

The GetPropertyType method returns the data type of the specified property.

**Syntax****C++**

```
int GetPropertyType (string& key)
int GetPropertyType (int key)
int GetPropertyType (char* key)
```

**COM**

```
HRESULT GetPropertyType (/*[in]*/ VARIANT* key, /*[out,retval]*/ int* value)
```

**VB**

```
GetPropertyType (key As VARIANT) As Int
```

**Java**

```
int GetPropertyType (string sPropName)
int GetPropertyType (int key)
```

**.NET**

```
virtual ArgDataType GetPropertyType (Enum_CtiOs eKeyId)
virtual ArgDataType GetPropertyType (System.String sPropName)
```

**Parameters**

key

Keyword ID name of the property whose type you want. In .NET, eKeyId is the Enum\_CtiOs Keyword ID of the property.

COM:value

An integer pointer to the value of the type.

**Return Value**

COM: Default HRESULT return value. For more information, see [CIL Coding Conventions](#)

Others: An integer indicating the property's type with the following possible values:

Argument Type	Description
ARG_NOTSET	Argument type not determined
ARG_INT	Signed integer
ARG_SHORT	2 bytes signed integer
ARG_BOOL	1 byte integer
ARG_STRING	C++, COM: STL character string VB String object
ARG_ARGARRAY	Variable length array of Arg
ARG_UINT	32 bit unsigned int
ARG_USHORT	16 bit unsigned short int
ARG_ARGUMENT	Arguments array

ARG_REFERENCE	Contains a reference to an object of a CtiOsObject derived class
---------------	--

## GetValue

The GetValue method returns the value of the specified property. Use this method if you do not know the type of the property. Otherwise, use the more specific GetValue methods discussed later in this chapter. When using the COM CIL, do not use this method for properties of type IDispatch\*; instead, use GetCurrentCall, GetCurrentAgent, GetAllCalls, GetAllAgents, and GetAllSkillGroups as explained in [Session Object](#)

### Syntax

#### C++

```
Arg& GetValue (string& key)
Arg& GetValue (int key)
Arg& GetValue (char* key)
```

#### COM

```
HRESULT GetValue (/*[in]*/ VARIANT* key, /*[out,retval]*/ IArg** value)
```

#### VB

```
GetValue (key As VARIANT) As CTIOSCLIENTLib.IArg
```

#### Java

```
Arg GetValue( String key )
Arg& GetValue (int key)
```

#### .NET

```
virtual System.Boolean GetValue(Enum_CtiOs eKeyID, out Arg obArg)
virtual System.Boolean GetValue(System.String sKey, out Arg obArg)
```

### Parameters

key

The name of the property whose value you want.

COM: value

An output value of type Arg\*\* containing the property with the designated name. To get the value of the property, call GetType() on the Arg and then call the specific GetValue method, based on the type.

.NET: obArg

Output parameter (return value in C++, VB, and Java) containing the specified property, as described in the explanation of the value parameter.

### Return Value

COM: Default HRESULT return value. For more information, see [CIL Coding Conventions](#)

.NET: Returns true if the value is retrieved, and false if the value is not found.

Others: An Arg containing the specified property. To get the value of the property, call GetType() on the Arg and then call the specific GetValue method, based on the type.



## GetValueArray

The GetValueArray method returns the Arguments array value of the specified property. Use this method when you know that the property is of Arguments array type, such as ECC call variables.

### Syntax

#### C++

```
Arg& GetValueArray (string& key)
Arg& GetValueArray (enum_Keywords key)
Arg& GetValue (char * key)
```

#### COM

```
HRESULT GetValueArray (/*[in]*/ VARIANT * key, /*[out,retval]*/ IArguments ** value)
```

#### VB

```
GetValueArray (key As VARIANT) As CTIOSCLIENTLib.IArguments
```

#### Java

```
Arguments GetValueArray( String key )
GetValueArray (int key)
```

#### .NET

```
System.Boolean GetValueArray(Enum_CtiOs eKeyID, out Arguments arArguments)
```

### Parameters

key

The name of the property whose value you want.

value

COM: An output parameter (return value in VB, C++, and Java) containing an arArguments\*\* to the returned value of the property.

.NET: An output parameter containing the Arguments array value upon success. Upon failure, this parameter is set to null.

### Return Value

COM: Default HRESULT return value. For more information, see [CIL Coding Conventions](#).

.NET: Returns true if the value is retrieved. Returns false if the value is not found.

Others: A reference to an Arguments array containing the value of the specified property.

## GetValueBoolObj (Java and .NET Only)

The GetValueBool method retrieves the Boolean value of the specified property.

### Syntax

```
Boolean GetValueBoolObj(int iKey)
Boolean GetValueBoolObj(String sKey)
```

**Parameters**

Key

Key ID for the object to be retrieved.

**Returns**

A Boolean object representation of the contained value or null if error.

## GetValueInt

The GetValueInt method returns the integer value of the specified property. Use this method when you know that the property has an integer type.

**Syntax****C++**

```
int GetValueInt (string& key)
int GetValueInt (int key)
int GetValueInt (char* key)
```

**COM**

```
HRESULT GetValueInt /*[in]*/ VARIANT* key, /*[out,retval]*/ int* value)
```

**VB**

```
GetValueInt (key As VARIANT) As Integer
```

**Java**

Not implemented, use GetValueIntObj

**.NET**

```
System.Boolean GetValueInt(Enum_CtiOs eKeyID, out System.Int32 nValue)
System.Boolean GetValueInt(System.String sPropname, out System.Int32 nValue)
```

**Parameters**

C++: key

Depending on the method used, either a string or int that contains the name or ID of the property whose value you want to retrieve.

COM, VB: key

VARIANT containing the ID or name of the property to retrieve.

COM: value

An output parameter (return parameter in VB) containing an integer pointer to the returned value of the property.

.NET: sPropName

The name of the property.

.NET: nValue

Upon success, this output parameter contains the value of the specified property. Upon failure, this parameter is set to null.

.NET: eKeyID

Keyword ID of the property.

### Return Value

COM: Default HRESULT return value. For more information, see [CIL Coding Conventions](#).

.NET: True if the method succeeds; false if the method fails.

## GetValueIntObj (Java Only)

Gets the contained value as an integer.

### Syntax

- `Integer GetValueIntObj( int iKey)`
- `Integer GetValueIntObj( String sKey)`

### Parameters

#### key

Key ID of the value to be retrieved.

### Returns

An Integer object containing a 32 bit signed integer value or null if error.

## GetValueShortObj (Java Only)

Retrieves a 16 bit short with the specified key from the array.

### Syntax

```
Short GetValueShortObj( int iKey)
Short GetValueShortObj(short sKey)
```

### Parameters

#### key

Key ID of the value to be retrieved.

### Return Value

A Short object containing a 16 bit short value or null if error.

## GetValueString

The `GetValueString` method returns the string value of the property with the specified name. Use this method when you know that the property is of string type. For all CILs and all types, a call to `GetValueString` for an argument that is of another type, for example `Int`, returns a string representation of the argument's `Int` value.

**Syntax****C++**

```
string GetValueString (string& key)
string GetValueString (int key)
string GetValueString (char* key)
```

**COM**

```
HRESULT GetValueString (/*[in]*/ VARIANT* key, /*[out,retval]*/ BSTR* value)
```

**VB**

```
GetValueString (key As VARIANT) As String
```

**Java**

```
String GetValueString( String key )
String GetValueString (int key)
```

**.NET**

```
System.Boolean GetValueString(Enum_CtiOs eKeyID, out System.String strValue)
System.Boolean GetValueString(System.String sPropName, out System.String strValue)
```

**Parameters**

C++, Java: key

Depending on the method used, either a string or int that contains the name or ID of the property whose value you want to retrieve.

COM, VB: key

VARIANT containing the ID or name of the property to retrieve.

.NET: sPropName

The name of the property values to retrieve.

.NET: strValue

Upon success, this output parameter contains the value of the specified property. Upon failure, this parameter is set to null.

.NET: eKeyID

Keyword ID of the property.

value

In COM, an output parameter (return parameter in VB) containing a BSTR pointer to the returned string value of the property.

**Return Value**

COM: Default HRESULT return value. For more information, see [CIL Coding Conventions](#).

.NET: Boolean value indicating the success or failure of the method call (true, if success; otherwise false).

Others: A string containing the value of the specified property.

## GetValueUIntObj (Java Only)

Retrieves a 32 bit unsigned integer with the specified key from the array.

### Syntax

```
Long GetValueUIntObj( int key)
Long GetValueUIntObj( String sKey)
```

### Parameters

#### key

Key ID of the value to be retrieved.

### Returns

A Long object containing the 32 bit unsigned integer value or null if error.

## GetValueUShortObj (Java Only)

Retrieves a 16 bit unsigned short with the specified key from the array.

### Syntax

```
• Integer GetValueUShortObj( String sKey)
```

### Parameters

#### key

Key ID of the value to be retrieved.

### Returns

An Integer object containing the 16 bit unsigned short value or null if error.

## IsValid

The IsValid method tests to see if the object includes the specified property.

### Syntax

#### C++

```
bool IsValid (string& key)
bool IsValid (char* key)
bool IsValid (int key)
```

#### COM

```
HRESULT IsValid (/*[in]*/ VARIANT* key, /*[out,retval]*/ VARIANT_BOOL* value)
```

#### VB

```
IsValid (key As VARIANT) As Bool
```

**Java**

```
boolean IsValid( String key)
boolean IsValid (int key)
```

**.NET**

```
virtual bool IsValid(Enum_CtiOs eKeyID)
virtual bool IsValid(System.String sKey)
```

**Parameters**

Environment	Parameter	Description
C++, Java	key	A key containing the name or ID of the property whose validity you are testing.
COM, VB	key	VARIANT containing the name or ID of the property to retrieve.
.NET	eKeyID	The ID of the property whose validity you are testing.
.NET	sKey	The name of the property whose validity you are testing.
COM	value	An output parameter (return parameter in VB) containing a VARIANT_BOOL pointer indicating whether or not a property with the specified name exists for the object.

**Return Value**

COM: Default HRESULT return value. For more information, see [CIL Coding Conventions](#).

Others: A boolean indicating whether or not a property with the specified name exists for the object.

## ReportError (Java and .NET only)

The ReportError method sets the value of the LastError property to iErrCode and writes the error to the log as critical.

**Syntax**

```
int ReportError(int iError)
```

**Parameters**

Error

The error to report.

**Returns**

The same error code that was passed in through iErrCode.

## SetValue (Java and .NET)

The SetValue method adds a new property to the object's property list and gives it the provided value. If the property already exists, it replaces its value with the one provided.

### Syntax

```
boolean SetValue(int iKeyID, Arg rArg)
boolean SetValue(int iKeyID, Arguments rArgs)
boolean SetValue(int iKeyID, boolean bValue)
boolean SetValue(int iKeyID, CtiOsObject rObj)
boolean SetValue(int iKeyID, int iValue)
boolean SetValue(int iKeyID, short nValue)
boolean SetValue(int iKeyID, java.lang.String sValue)
boolean SetValue(java.lang.String sPropName, Arg rArg)
boolean SetValue(java.lang.String sPropName, Arguments rArgs)
boolean SetValue(java.lang.String sPropName, boolean bValue)
boolean SetValue(java.lang.String sPropName, CtiOsObject rObj)
boolean SetValue(java.lang.String sPropName, int iValue)
boolean SetValue(java.lang.String sPropName, short nValue)
boolean SetValue(java.lang.String sPropName, java.lang.String sValue)
boolean SetValueUInt (int key, long value)
boolean SetValueUInt (String key, long value)
boolean SetValueUShort (int key, int value)
boolean SetValueUShort (String key, int value)
```

### Parameters

key

The key whose value is to be set.

value

The value to use in setting the element with the designated key.

### Returns

True if successfully added, false if not.

## SetValue (C++ COM and VB)

The SetValue method sets the value of the specified Agent property.

### Syntax

C++

```
bool SetValue( string& key, string& value)
bool SetValue( string& keyValuePair)
bool SetValue( string& key, int value)
bool SetValue( const char * key, const char * value)
bool SetValue( const char * keyValuePair)
bool SetValue( const char * key, int value)
```

**COM**

```
HRESULT SetValue (/*[in]*/ VARIANT *key, /*[in]*/ VARIANT *value, /*[out,retval]*/  
VARIANT_BOOL * errorcode)
```

**VB**

```
SetValue (key As Variant, value As Variant) As Bool
```

**Parameters****key**

An input parameter that contains the name of the property whose value you want to set.

**value**

An input parameter containing the value to be used in setting the specified property.

**keyValuePair**

An input parameter containing a string in the format “key=value” where key is a property to set and value is the new value.

**errorcode**

An output parameter (return parameter in VB) that contains an error code from [Table 1](#).

**Return Values****COM**

Default CTI OS return values. For more information, see [CIL Coding Conventions](#).

**All Others**

A boolean indicating the success or failure of the method.

**Remarks**

You should only use this method when creating a new Agent in preparation for logging in. Therefore, use it to set the AgentID, AgentInstrument, AgentPassword, PeripheralID, and AutoLogin only.