

goRegistry

Inhaltsverzeichnis

- [1 eRegistryKeys](#)
- [2 eRegistryPermissions](#)
- [3 bCreateKey](#)
- [4 bDeleteKey](#)
- [5 bDeleteValue](#)
- [6 bKeyExists](#)
- [7 bKeyPermission](#)
- [8 bSetValue](#)
- [9 colGetSubKeys](#)
- [10 vntGetValue](#)
- [11 bSetMultiUserValue](#)
- [12 sGetMultiUserValue](#)

Das Aufgaben-Center Objekt goRegistry

Das Objekt **goRegistry** bietet Funktionen zum Zugriff auf die Windows Registry

1 eRegistryKeys

Die Enumeration **eRegistryKeys** definiert die Bereiche der Windows Registry.

clsStaticRegistry.eRegistryKeys Wert

HKEY_CLASSES_ROOT	&H80000000
HKEY_CURRENT_USER	&H80000001
HKEY_LOCAL_MACHINE	&H80000002
HKEY_USERS	&H80000003
HKEY_PERFORMANCE_DATA	&H80000004
HKEY_CURRENT_CONFIG	&H80000005
HKEY_DYN_DATA	&H80000006

2 eRegistryPermissions

Die Enumeration **eRegistryPermissions** definiert Berechtigungen für die Windows Registry.

clsStaticRegistry.eRegistryPermissions Wert

KEY_GET_VALUE	&H1
KEY_SET_VALUE	&H2
KEY_CREATE	&H4
KEY_DELETE	&H10000

3 bCreateKey

Die Funktion **bCreateKey** erstellt einen Schlüssel in der Windows Registry. Über das Argument *eRootKey* wird der Bereich in der Windows Registry bestimmt. Mit dem Argument *sKeyPath* wird die Bezeichnung des Schlüssels gesetzt. Wurde der Schlüssel erfolgreich erstellt liefert die Funktion *True* zurück.

bCreateKey(eRootKey As eRegistryKeys, sKeyPath As String, Optional bIgnoreError As Boolean = False)
As Boolean

4 bDeleteKey

Die Funktion **bDeleteKey** löscht einen Schlüssel aus der Windows Registry. Über das Argument *eRootKey* wird der Bereich in der Windows Registry bestimmt. Mit dem Argument *sKeyPath* wird der zu löschende Schlüssel angegeben. Wurde der Schlüssel erfolgreich entfernt liefert die Funktion *True* zurück.

bDeleteKey(eRootKey As eRegistryKeys, sKeyPath As String, Optional bIgnoreError As Boolean = False)
As Boolean

5 bDeleteValue

Die Funktion **bDeleteValue** löscht einen Wert aus einem Schlüssel der Windows Registry. Über das Argument *eRootKey* wird der Bereich in der Windows Registry bestimmt. Mit dem Argument *sKeyPath* wird der Schlüssel und mit *sValueName* der Name des zu löschenden Wertes angegeben. Wurde der Wert erfolgreich entfernt liefert die Funktion *True* zurück.

bDeleteValue(eRootKey As eRegistryKeys, sKeyPath As String, sValueName As String, Optional
bIgnoreError As Boolean = False) As Boolean

6 bKeyExists

Die Funktion **bKeyExists** prüft ob ein Schlüssel in der Windows Registry existiert. Über das Argument *eRootKey* wird der Bereich in der Windows Registry bestimmt. Mit dem Argument *sKeyPath* wird Name des

Schlüssels angegeben. Wurde der Schlüssel gefunden liefert die Funktion *True* zurück.

`bKeyExists(eRootKey As eRegistryKeys, sKeyPath As String, Optional bIgnoreError As Boolean = False) As Boolean`

7 bKeyPermission

Mit der Funktion **bKeyPermission** kann die Zugriffs-Berechtigung des aktuellen Benutzers auf einen Schlüssel der Windows Registry geprüft werden. Über das Argument *eRootKey* wird der Bereich in der Windows Registry bestimmt. Mit dem Argument *sKeyPath* wird Name des Schlüssels angegeben. Und über *eRegPermission* die zu prüfende Berechtigung gesetzt. Ist die Berechtigung vorhanden liefert die Funktion *True* zurück.

`bKeyPermission(eRootKey As eRegistryKeys, sKeyPath As String, eRegPermission As eRegistryPermissions, Optional bIgnoreError As Boolean = False) As Boolean`

8 bSetValue

Mit der Funktion **bSetValue** kann ein Wert in die Windows Registry geschrieben werden. Über das Argument *eRootKey* wird der Bereich in der Windows Registry bestimmt. Mit dem Argument *sKeyPath* wird Name des Schlüssels und mit *sValueName* der Name für den neuen Wert angegeben. Mit dem optionalen Argument *bDWORD* kann festgelegt werden, dass der Wert als Dword in der Registry erstellt werden soll, ist das Argument nicht gesetzt wird eine Zeichenfolge erstellt. Ist der Wert erfolgreich geschrieben worden, liefert die Funktion *True* zurück.

`bSetValue(eRootKey As eRegistryKeys, sKeyPath As String, sValueName As String, vntValue As Object, Optional bDWORD As Boolean = False, Optional bIgnoreError As Boolean = False) As Boolean`

9 colGetSubKeys

Die Funktion **colGetSubKeys** liefert eine *Collection* mit Strings von untergeordneten Registry Schlüsseln. Über das Argument *eRootKey* wird der Bereich in der Windows Registry bestimmt. Mit dem Argument *sKeyPath* wird Name des übergeordneten Schlüssels angegeben.

`colGetSubKeys(eRootKey As eRegistryKeys, sKeyPath As String, Optional bIgnoreError As Boolean = False) As Collection`

10 vntGetValue

Mit der Funktion **vntGetValue** kann ein Wert aus der Windows Registry ausgelesen werden. Über das Argument *eRootKey* wird der Bereich in der Windows Registry bestimmt. Mit dem Argument *sKeyPath* wird Name des Schlüssels und mit *sValueName* der Name des auszulesenden Eintrags angegeben. Mit dem optionalen Argument *bDWORD* kann festgelegt werden, ob der Wert als Dword in der Registry vorliegt, ist das Argument nicht gesetzt wird eine Zeichenfolge ausgelesen.

```
vntGetValue(eRootKey As eRegistryKeys, sKeyPath As String, ByVal sValueName As String, Optional  
bDWORD As Boolean = False, Optional blgnoreError As Boolean = False) As Object
```

11 bSetMultiUserValue

Über die Funktion **bSetMultiUserValue** kann ein Wert *sValue* in die Windows Registry des Servers mit dem Mehrbenutzerdienst geschrieben werden. Das Argument *sKey* gibt den Registry-Schlüssel und *sValueName* den Name des Eintrags an. Im Standard wird der Schlüssel unter `Software\Sage\Office Line\x.x\` angelegt, soll ein anderer Pfad verwendet werden kann dies mit dem Argument *sPath* bestimmt werden.

```
bSetMultiUserValue(sKey As String, sValueName As String, sValue As String, Optional sPath As String = "",  
Optional blgnoreError As Boolean = False) As Boolean
```

12 sGetMultiUserValue

Über die Funktion **sGetMultiUserValue** kann ein Wert aus der Windows Registry des Servers mit dem Mehrbenutzerdienst ausgelesen werden. Das Argument *sKey* gibt den Registry-Schlüssel und *sValueName* den Name des Eintrags an. Mit dem Argument *sDefault* kann ein Standardwert gesetzt werden, dieser wird als Rückgabewert herangezogen falls der zu lesende Eintrag nicht existiert. Im Standard wird der Schlüssel unter `Software\Sage\Office Line\x.x\` gelesen, soll ein anderer Pfad verwendet werden kann dies mit dem Argument *sPath* bestimmt werden.

```
sGetMultiUserValue(sKey As String, sValueName As String, Optional sDefault As String = "", Optional sPath  
As String = "", Optional blgnoreError As Boolean = False) As String
```