BECKHOFF New Automation Technology

Handbuch | DE

Benutzerverwaltung

TwinCAT 2 | System

19.07.2022 | Version: 1.0

Inhaltsverzeichnis

1 Vorwort

1.1 Hinweise zur Dokumentation

Diese Beschreibung wendet sich ausschließlich an ausgebildetes Fachpersonal der Steuerungs- und Automatisierungstechnik, das mit den geltenden nationalen Normen vertraut ist.
Zur Installation und Inbetriebnahme der Komponenten ist die Beachtung der Dokumentation und der nachfolgenden Hinweise und Erklärungen unbedingt notwendig.
Das Fachpersonal ist verpflichtet, für jede Installation und Inbetriebnahme die zu dem betreffenden Zeitpunkt veröffentliche Dokumentation zu verwenden.

Das Fachpersonal hat sicherzustellen, dass die Anwendung bzw. der Einsatz der beschriebenen Produkte alle Sicherheitsanforderungen, einschließlich sämtlicher anwendbaren Gesetze, Vorschriften, Bestimmungen und Normen erfüllt.

Disclaimer

Diese Dokumentation wurde sorgfältig erstellt. Die beschriebenen Produkte werden jedoch ständig weiter entwickelt.
Wir behalten uns das Recht vor, die Dokumentation jederzeit und ohne Ankündigung zu überarbeiten und zu ändern.
Aus den Angaben, Abbildungen und Beschreibungen in dieser Dokumentation können keine Ansprüche auf Änderung bereits gelieferter Produkte geltend gemacht werden.

Marken

Beckhoff®, TwinCAT®, TwinCAT/BSD®, TC/BSD®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC®, XTS® und XPlanar® sind eingetragene und lizenzierte Marken der Beckhoff Automation GmbH.
Die Verwendung anderer in dieser Dokumentation enthaltenen Marken oder Kennzeichen durch Dritte kann zu einer Verletzung von Rechten der Inhaber der entsprechenden Bezeichnungen führen.

Patente

Die EtherCAT-Technologie ist patentrechtlich geschützt, insbesondere durch folgende Anmeldungen und Patente:
EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702
mit den entsprechenden Anmeldungen und Eintragungen in verschiedenen anderen Ländern.

EtherCAT®

EtherCAT® ist eine eingetragene Marke und patentierte Technologie lizenziert durch die Beckhoff Automation GmbH, Deutschland

Copyright

1.2 Sicherheitshinweise

Sicherheitsbestimmungen

Beachten Sie die folgenden Sicherheitshinweise und Erklärungen!
Produktspezifische Sicherheitshinweise finden Sie auf den folgenden Seiten oder in den Bereichen Montage, Verdrahtung, Inbetriebnahme usw.

Haftungsausschluss

Die gesamten Komponenten werden je nach Anwendungsbestimmungen in bestimmten Hard- und Software-Konfigurationen ausgeliefert. Änderungen der Hard- oder Software-Konfiguration, die über die dokumentierten Möglichkeiten hinausgehen, sind unzulässig und bewirken den Haftungsausschluss der Beckhoff Automation GmbH & Co. KG.

Qualifikation des Personals

Diese Beschreibung wendet sich ausschließlich an ausgebildetes Fachpersonal der Steuerungs-, Automatisierungs- und Antriebstechnik, das mit den geltenden Normen vertraut ist.

Erklärung der Symbole

In der vorliegenden Dokumentation werden die folgenden Symbole mit einem nebenstehenden Sicherheitshinweis oder Hinweistext verwendet. Die Sicherheitshinweise sind aufmerksam zu lesen und unbedingt zu befolgen!

⚠ GEFAHR

Akute Verletzungsgefahr!

Wenn der Sicherheitshinweis neben diesem Symbol nicht beachtet wird, besteht unmittelbare Gefahr für Leben und Gesundheit von Personen!

⚠ WARNUNG

Verletzungsgefahr!

Wenn der Sicherheitshinweis neben diesem Symbol nicht beachtet wird, besteht Gefahr für Leben und Gesundheit von Personen!

⚠ VORSICHT

Schädigung von Personen!

Wenn der Sicherheitshinweis neben diesem Symbol nicht beachtet wird, können Personen geschädigt werden!

HINWEIS

Schädigung von Umwelt oder Geräten

Wenn der Hinweis neben diesem Symbol nicht beachtet wird, können Umwelt oder Geräte geschädigt werden.

Tipp oder Fingerzeig

Dieses Symbol kennzeichnet Informationen, die zum besseren Verständnis beitragen.

1.3 Hinweise zur Informationssicherheit

Die Produkte der Beckhoff Automation GmbH & Co. KG (Beckhoff) sind, sofern sie online zu erreichen sind, mit Security-Funktionen ausgestattet, die den sicheren Betrieb von Anlagen, Systemen, Maschinen und Netzwerken unterstützen. Trotz der Security-Funktionen sind die Erstellung, Implementierung und ständige Aktualisierung eines ganzheitlichen Security-Konzepts für den Betrieb notwendig, um die jeweilige Anlage, das System, die Maschine und die Netzwerke gegen Cyber-Bedrohungen zu schützen. Die von Beckhoff verkauften Produkte bilden dabei nur einen Teil des gesamtheitlichen Security-Konzepts. Der Kunde ist dafür verantwortlich, dass unbefugte Zugriffe durch Dritte auf seine Anlagen, Systeme, Maschinen und Netzwerke verhindert werden. Letztere sollten nur mit dem Unternehmensnetzwerk oder dem Internet verbunden werden, wenn entsprechende Schutzmaßnahmen eingerichtet wurden.

Zusätzlich sollten die Empfehlungen von Beckhoff zu entsprechenden Schutzmaßnahmen beachtet werden. Weiterführende Informationen über Informationssicherheit und Industrial Security finden Sie in unserem https://www.beckhoff.de/secguide.

Die Produkte und Lösungen von Beckhoff werden ständig weiterentwickelt. Dies betrifft auch die Security-Funktionen. Aufgrund der stetigen Weiterentwicklung empfiehlt Beckhoff ausdrücklich, die Produkte ständig auf dem aktuellen Stand zu halten und nach Bereitstellung von Updates diese auf die Produkte aufzuspielen. Die Verwendung veralteter oder nicht mehr unterstützter Produktversionen kann das Risiko von Cyber-Bedrohungen erhöhen.

Um stets über Hinweise zur Informationssicherheit zu Produkten von Beckhoff informiert zu sein, abonnieren Sie den RSS Feed unter https://www.beckhoff.de/secinfo.

2 Allgemein

TwinCAT ist ein allgemeines Automatisierungssystem bzw. ein Automatisierungsbaukasten, dessen Komponenten in unterschiedlichen Konfigurationen hinzugefügt oder weggelassen werden können. Da TwinCAT sehr modular aufgebaut ist und die Module unabhängig voneinander sind, ist die Benutzerverwaltung unabhängig von den anderen Modulen implementiert. Das Betriebssystem des PCs (Windows NT/2000, Windows CE) ist ebenfalls ein Teil des TwinCAT Automatisierungssystems und muss deshalb in die Benutzerverwaltung einbezogen werden.

Beim Einsatz von TwinCAT unter Windows NT*/2000 basiert die Benutzerverwaltung auf dem Sicherheitssystem von Windows NT, d.h. ein NT Administrator ist auch ein TwinCAT Administrator und Benutzer einer NT-Domäne sind auch in TwinCAT bekannt. So kann (wenn nötig) das gesamte System mit Windows NT Werkzeugen administriert werden. Das Sicherheitssystem von Windows NT kann Zugriffsrechte für jedes Objekt des Betriebssystems (Partition, Verzeichnis, Datei, Netzwerk, ...) entziehen oder vergeben. Der Zugriff auf das NT-Sicherheitssystem ist leider undurchsichtig, da sich das API auf mehrere Teile des Betriebssystems verteilt. So finden einige Funktionen in der "Lan Manager" API andere wiederum in der eigentlichen Security API. Die TwinCAT Benutzerverwaltung fasst die notwendigen Funktionen in einem COM-Objekt "TcUserManager [▶ 10]" zusammen. TcUserManager implementiert natürlich nicht alle Funktionen des NT-Sicherheitssystems, sondern nur die Funktionen, die an einer Maschinensteuerung brauchbar erscheinen. Sollten in Sonderfällen zusätzliche Windows NT Funktionalitäten notwendig sein, so kann auch direkt auf die entsprechenden Windows NT APIs zugegriffen werden, da Windows NT auf dieselbe Benutzerdatenbank zurückgreift wie die TwinCAT Benutzerverwaltung auch.

Einsatz

Idealerweise startet die Maschinensteuerung, ohne dass sich ein Benutzer einloggen muss. In der Regel wird ein Standard-Benutzer automatisch eingeloggt und die Bedienoberfläche durch TwinCAT gestartet. Der Standard-Benutzer sollte nur eingeschränkte Benutzerrechte (einfacher Windows NT Benutzer) haben, da er sonst Veränderungen am System (z.B. Installieren von Treibern) durchführen kann, die zu Beeinträchtigungen der Maschinenfunktion führen können. Falls im Produktionsbetrieb der Maschine die Ausübung von privilegierten Funktionen (Löschen von Dateien, Ändern von Produktionsabläufen, Ändern des SPS-Programms) notwendig ist, so muss ein komplettes Ausloggen und Wiedereinloggen in Windows NT natürlich vermieden werden. Stattdessen kann der Bedienoberflächenprozess (eigentlich ist es nur einer der Threads) den Status eines höher privilegierten Benutzers annehmen (ImpersonateUser [▶ 11]), oder es kann ein zusätzlicher Prozess wie z.B. der TwinCAT System Manager mit dem Benutzerkonto eines Administrators von der Bedienoberfläche mit CreateProcessAsUser [▶ 13] gestartet werden. Jeder Benutzer in TwinCAT kann zu mehreren Benutzergruppen gehören, um spezielle Funktionen der Bedienoberfläche zu schützen, kann der Zugriff mit UserIsMemberOf [▶ 17] oder mit UserIsAdmin [▶ 17] überprüft werden. Die Methode UserIsMemberOf überprüft die Benutzergruppe zurzeit anhand des Namens, wobei Probleme mit der Landessprache auftreten können. Um diesen Problemen aus dem Wege zu gehen und außerdem einheitliche Benutzergruppen im TwinCAT System zu haben, sind Standardbenutzergruppen für TwinCAT festgelegt:

- **TcUsers**
- **TcPowerUsers**
- **TcAdministrators**

Diese Benutzergruppen werden als globale Windows NT Benutzergruppen durch die TwinCAT Installation zur Windows NT Datenbasis hinzugefügt (ist noch zu implementieren).

*Nur TwinCAT 2.9 läuft unter Windows NT.

3 Lokale Sicherheitslinie

Die Lokale Sicherheitslinie enhält sicherheitsrelevante Einstellungen des Betriebssystems.

Wenn man einen Prozess aus einer Anwendung [▶ 13] der TwinCAT Benutzerschnittstelle aufrufen möchte, dann benötigen diese Benutzer einige spezielle Rechte auf dem Zielsystem.

Mit dem Befehl: **secpol.msc /s** kann man den "Local Securty Settings" Dialog aufrufen, um die Benutzerprivilegien im Einzelnen einstellen zu können.

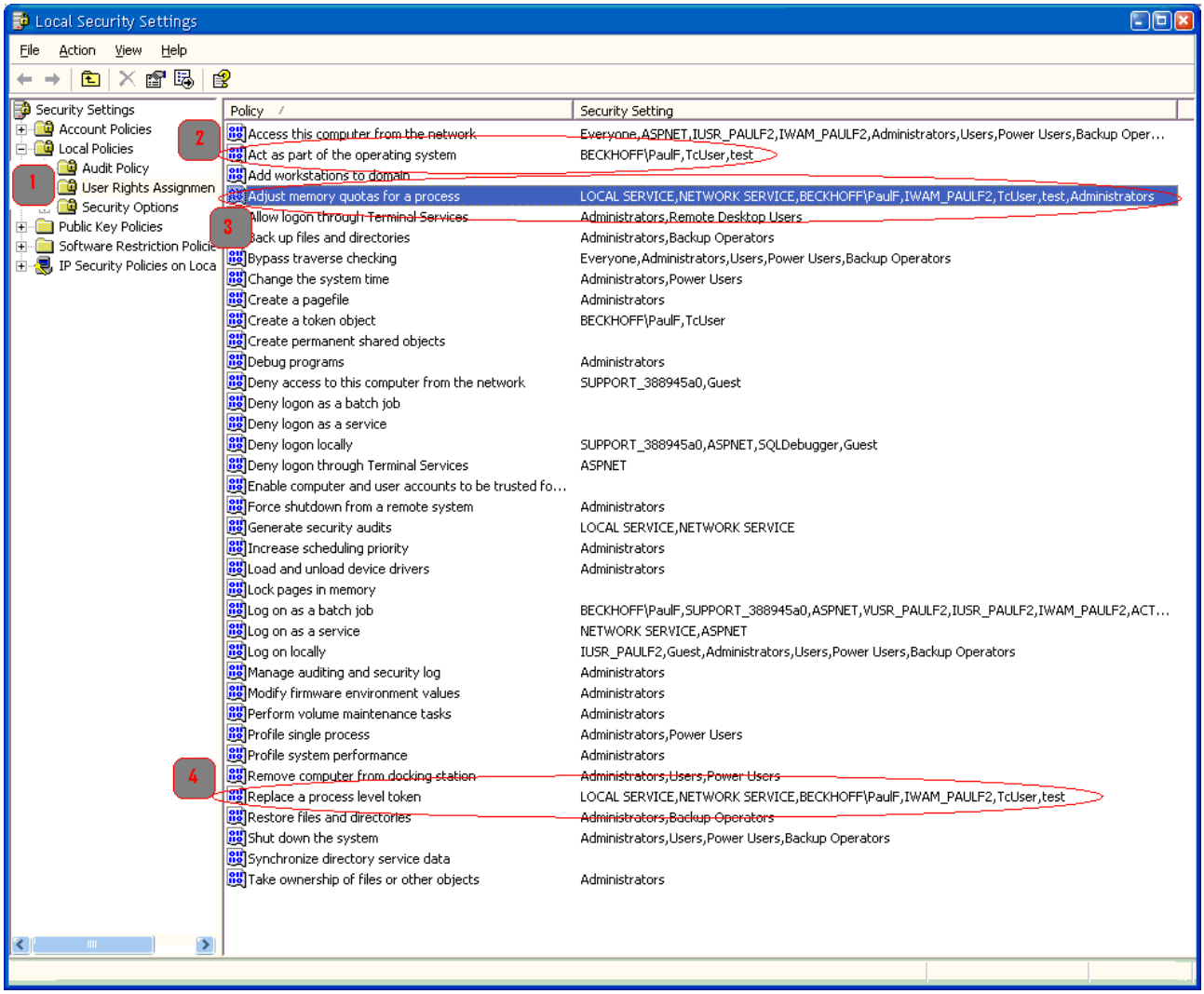

Abb. 1: "Local Security Settings" eines Windows XP Professional Systems

Links im Dialog sind die Sicherheitslinien, sogenannte "Policies", nach ihren Typen gruppiert. In dem Eintrag "User Rights Assignment" (Zuweisen von Benutzerrechten, Punkt 1) sind die notwendigen Privilegien enthalten, die man zum Aufrufen von Prozessen benötigt.

Mit einem Doppelklick auf die jeweiligen Privilegien kann man die Benutzer eintragen, die diese Rechte erhalten sollen.

Folgende Punkte müssen editiert werden:

- **2** (act as part of the operating system / Einsetzen als Teil des Betriebssystems)
- **3** (adjust memory quotas for a process / Anpassen von Speicherkontingenten für einen Prozess)
- **4** (Replace a process level token / Erstzen eines Token auf Prozessebene)

Dort sollten auf jeden Fall der Benutzer eingetragen werden, mit dem man den Prozess starten möchte und der Benutzer, mit dem man die Anwendung aufgerufen hat.

4 API

4.1 TcUserManager (CoClass)

In der Klasse TcUserManager ist die Benutzerverwaltung von TwinCAT implementiert. TcUserManager setzt direkt auf die Windows NT Benutzerverwaltung auf und kann deshalb mit allen Sicherheitseigenschaften von Windows NT aufwarten. Eine wesentliche Eigenschaft des Windows NT Sicherheitssystems ist die Möglichkeit, Rechte für jedes Betriebssystemobjekt (z. B. Festplatte, Datei, Netzwerk,…) zu vergeben bzw. zu entziehen. Weiterhin können die Benutzer, Benutzergruppen und deren Rechte im Netzwerk propagiert werden.

Das default Interface von **TcUserManager** heißt ITcUserManager [▶ 10], implementiert ist **TcUserManager** als "In Process Server" in der DLL "TCATUserMan.dll". Ein Teil der Funktionalität ist aus Kompatibilitätsgründen auch durch exportierte Funktionen der DLL zu erreichen. Neue Applikationen sollten allerdings das COM Interface ITcUserManager [▶ 10] benutzen.

4.2 ITcUserManager

Das Interface ITcUserManager erlaubt den Zugriff auf die TwinCAT Benutzerverwaltung. Die TwinCAT Benutzerverwaltung basiert auf dem Windows NT Sicherheitssystem. Eine wesentliche Eigenschaft des Windows NT Sicherheitssystems ist die Möglichkeit Rechte für jedes Betriebssystemobjekt (z.B. Festplatte, Datei, Netzwerk,…) zu vergeben bzw. zu entziehen. Weiterhin können die Benutzer, Benutzergruppen und deren Rechte im Netzwerk propagiert werden.

Tab. 1: Methoden in Vtable Reihenfolge

IUnknown Methoden	Beschreibung
QueryInterface	Liefert einen Zeiger auf das angeforderte Interface zurück.
AddRef	Inkrementiert den Referenzzähler.
Release	Dekrementiert den Referenzzähler.
IDispatch Methoden	Beschreibung
GetTypeInfoCount	Liefert die Anzahl der "Type Information" Interfaces, die ein Objekt anbietet. (0 or 1).
GetTypeInfo	Holt die Typinformationen für ein Objekt.
GetIDsOfNames	Verknüpft Namen von Methoden mit optionalen Argumenten mit einem zugehörigem Satz von DISPIDs.
Invoke	Bietet den Zugriff auf Eigenschaften und Methoden eines Objekts.

ITcUserManager Methoden	Beschreibung
CreateUserAccount [▶ 14]	Legt ein neues Benutzerkonto an.
CreateUserAccountDlg [▶ 15]	Dialog zur Generierung eines neuen Benutzerkontos.
ImpersonateUser [▶ 11]	Der aufrufende Thread personifiziert den mit Namen und Passwort bezeichneten Benutzer.
ImpersonateUserDlg [▶ 12]	Dialog mit Passwortabfrage zur Personifizierung eines Benutzers.
RevertToSelf [▶ 12]	Eine vorhergehende Personifizierung wird zurückgenommen.
UserIsMemberOf [▶ 17]	Überprüft ob der aufrufende Thread Mitglied der gewünschten Benutzergruppe ist.
CreateProcessAsUser [▶ 13]	Startet einen Process mit dem gewünschten Benutzerkonto.

ITcUserManager Methoden	Beschreibung
CreateProcessAsUserDlg [▶ 14]	Dialog zum Start eines Prozesses mit angegebenen Benutzerkonto
UserIsAdmin [▶ 17]	Prüft ob der aufrufende Thread zu der Gruppe der Administratoren gehört
UserName [▶ 18]	Eigenschaft (Property) zur Anzeige des aktuellen Benutzernamens
UserEnumLocalGroups [▶ 16]	Liefert die Benutzergruppen eines Benutzers zurück.
RemoveUserAccount [▶ 12]	Entfernt ein Benutzerkonto
EnumLocalGroups [▶ 16]	Liefert die lokal definierten Benutzergruppen zurück.
EnumLocalUsers [▶ 15]	Liefert die lokal definierten Benutzer zurück.

4.2.1 ImpersonateUser

ITcUserManager [▶ 10]::ImpersonateUser

Der aufrufende Thread personifiziert den mit Namen und Passwort bezeichneten Benutzer.

```
HRESULT ImpersonateUser(
  BSTR bstrUserName,
  BSTR bstrPassword,
  VARIANT varDomain
);
```

Parameters

bstrUserName	[in] Name des Benutzers, der personifiziert werden soll.
bstrPassword	[in] Passwort des Benutzers.
varDomain	[in, optional] Falls der Benutzer zu einer Domäne gehört, kann hier optional der Name der betreffenden Domäne übergeben werden. Der Übergabeparameter muss vom Typ BSTR sein, um übernommen zu werden.

Return Values

HRESULT == S_OK	Benutzerkonto wurde erfolgreich erzeugt.
HRESULT != S_OK	Im Fehlerfall steht in HRESULT ein von Windows NT generierter Fehlercode. Um eine COM/OLE Fehlerbehandlung zu provozieren, wird aus dem Fehlercode ein HRESULT durch HRESULT_FROM_NT(nErrror) bzw. HRESULT_FROM_WIN32(nErrror) erzeugt. Zu beachten ist hierbei, dass der Windows NT Fehlercode in niederwertigen 16 Bit dargestellt wird.

Bemerkungen

Die Methode **ImpersonateUser** weist den "Impersonation" Token des gewünschten Benutzers dem aufrufenden Thread zu. Dazu wird der Benutzer mit Benutzername und Passwort interaktiv eingeloggt und der erzeugte Token zur Personifizierung benutzt. Nach erfolgreicher Ausführung von ImpersonateUser hat der Thread den Benutzer angenommen, dies kann durch Aufruf der Methode RevertToSelf [▶ 12] wieder rückgängig gemacht werden.

4.2.2 ImpersonateUserDlg

ITcUserManager [▶ 10]::ImpersonateUserDlg

Dialog zum Wechseln des Benutzers im laufenden Prozess.

```
HRESULT ImpersonateUserDlg()
```

Return Values

HRESULT == S_OK	Benutzerkonto wurde erfolgreich erzeugt.
HRESULT != S_OK	Im Fehlerfall steht in HRESULT ein von Windows NT generierter Fehlercode. Um eine COM/OLE Fehlerbehandlung zu provozieren, wird aus dem Fehlercode ein HRESULT durch HRESULT_FROM_NT(nErrror) erzeugt. Zu beachten ist hierbei, dass der Windows NT Fehlercode in niederwertigen 16 Bit dargestellt wird.

Bemerkungen

ImpersonateUserDlg ruft intern die Methode ImpersonateUser [▶ 11] auf.

4.2.3 RemoveUserAccount

ITcUserManager [▶ 10]::RemoveUserAccount

ITcUserManager [▶ 10]

Entfernt ein Benutzerkonto.

```
HRESULT RemoveUserAccount(
BSTR bstrUserName,
  VARIANT serverName
);
```

Parameters

bstrUserName	[in] Name des Benutzerskontos, das entfernt werden soll.
varComputerName	[in, optional] Optional kann der Name des Computers als BSTR übergeben werden. Wird kein Name übergeben, so wird das Benutzerkonto auf dem lokalem Computer entfernt.

Return Values

HRESULT == S_OK	Entfernen erfolgreich durchgeführt.
HRESULT != S_OK	Im Fehlerfall steht in HRESULT ein von Windows NT generierter Fehlercode. Um eine COM/OLE Fehlerbehandlung zu provozieren, wird aus dem Fehlercode ein HRESULT durch HRESULT_FROM_NT(nErrror) erzeugt. Zu beachten ist hierbei, dass der Windows NT Fehlercode in niederwertigen 16 Bit dargestellt wird.

4.2.4 RevertToSelf

ITcUserManager [▶ 10]::RevertToSelf

Eine vorhergehende Personifizierung wird zurückgenommen.

```
HRESULT RevertToSelf();
```

Return Values

HRESULT == S_OK	Benutzerkonto wurde erfolgreich erzeugt.
HRESULT != S_OK	Im Fehlerfall steht in HRESULT ein von Windows NT generierter Fehlercode. Um eine COM/OLE Fehlerbehandlung zu provozieren, wird aus dem Fehlercode ein HRESULT durch HRESULT_FROM_NT(nErrror) bzw. HRESULT_FROM_WIN32(nErrror) erzeugt. Zu beachten ist hierbei, dass der Windows NT Fehlercode in niederwertigen 16 Bit dargestellt wird.

Bemerkungen

Die Methode RevertToSelf weist dem aufrufenden Thread den "Primary Token" des Prozesses zu. RevertToSelf nimmt also nicht nur die letzte Personifizierung zurück, sondern stellt auch nach aufeinanderfolgenden Personifizierungen den ersten Benutzer des Prozesses wieder her.

4.2.5 CreateProcessAsUser

ITcUserManager [▶ 10]::CreateProcessAsUser

ITcUserManager

Startet ein Prozess mit dem gewünschten Benutzerkonto.

```
HRESULT CreateProcessAsUser(
  BSTR bstrUserName,
  BSTR bstrPassword,
  BSTR bstrProcessPath,
  VARIANT varDomain
);
```

Parameters

bstrUserName	[in] Name des Benutzerkontos mit dem der Prozess gestartet werden soll.
bstrPassword	[in] Passwort für das Benutzerkonto.
varDomain	[in, optional] Falls der Benutzer zu einer Domäne gehört, kann hier optional der Name der betreffenden Domäne übergeben werden. Der Übergabeparameter muss vom Typ BSTR sein, um übernommen zu werden.

Return Values

HRESULT == S_OK	Prozess wurde erfolgreich gestartet.
HRESULT != S_OK	Im Fehlerfall steht in HRESULT ein von Windows NT generierter Fehlercode. Um eine COM/OLE Fehlerbehandlung zu provozieren, wird aus dem Fehlercode ein HRESULT durch HRESULT_FROM_NT(nErrror) bzw. HRESULT_FROM_WIN32(nErrror) erzeugt. Zu beachten ist hierbei, dass der Windows NT Fehlercode in niederwertigen 16 Bit dargestellt wird.

Bemerkungen

Die Methode **CreateProcessAsUser** startet ein Prozess mit den Rechten des angegebenen Benutzers. Der Prozess hat so auf alle Objekte (Partitionen, Verzeichnisse, Dateien, Netzwerk, etc.) Zugriff, die für den betreffenden Benutzer freigegeben sind. Eine Besonderheit tritt beim Windows Explorer ein. Standardmäßig gehören alle Explorerfenster zum selben Prozess, wird eine neue Instanz des Explorers mit **CreateProcessAsUser** gestartet, kommt es zum Konflikt, da der Explorerprozess schon aktiv ist und zu einem anderen Benutzerkonto gehört. Die Instanz beendet sich dann wieder ohne vom Benutzer bemerkt worden zu sein. Soll also eine Instanz des Explorers mit einem anderen Benutzerkonto gestartet werden, so muss der laufende Explorer-Prozess zuerst gestoppt werden.

4.2.6 CreateProcessAsUserDlg

ITcUserManager [▶ 10]::CreateProcessAsUserDlg

Dialog zum Start eines Prozesses mit dem angegebenen Benutzerkonto .

```
HRESULT CreateProcessAsUserDlg()
```

Return Values

HRESULT == S_OK	Der Prozess wurde erfolgreich gestartet.
HRESULT != S_OK	Im Fehlerfall steht in HRESULT ein von Windows NT generierter Fehlercode. Um eine COM/OLE Fehlerbehandlung zu provozieren, wird aus dem Fehlercode ein HRESULT durch HRESULT_FROM_NT(nErrror) erzeugt. Zu beachten ist hierbei, dass der Windows NT Fehlercode in niederwertigen 16 Bit dargestellt wird.

Bemerkungen

CreateProcessAsUserDlg ruft intern die Methode CreateProcessAsUser [▶ 13] auf.

4.2.7 CreateUserAccount

ITcUserManager [▶ 10]::CreateUserAccount

Legt ein neues lokales Benutzerkonto an.

```
HRESULT CreateUserAccount(
  BSTR bstrUserName,
  BSTR bstrPassword,
  VARIANT varUserGroup,
  VARIANT varDomain
);
```

Parameters

bstrUserName	[in] Name des neuen Benutzerkontos.
bstrPassword	[in] Zeichenkette für das gewünschte Passwort.
varUserGroup	[in, optional] Optional kann eine Benutzergruppe angegeben werden, zu der das neue Benutzerkonto gehören soll. Der Übergabeparameter muss vom Typ BSTR sein, um übernommen zu werden.
varDomain	[in, optional] Ebenfalls optional kann die Domain übergeben werden, in der das neue Benutzerkonto erzeugt werden soll. Der Übergabeparameter muss vom Typ BSTR sein, um übernommen zu werden. Diese Option wird zur Zeit noch nicht unterstützt.

Return Values

HRESULT = S_OK	Benutzerkonto wurde erfolgreich erzeugt.
HRESULT <> S_OK	Im Fehlerfall steht in HRESULT ein von Windows NT generierter Fehlercode. Um eine COM/OLE Fehlerbehandlung zu provozieren, wird aus dem Fehlercode ein HRESULT durch HRESULT_FROM_NT(nErrror) erzeugt. Zu beachten ist hierbei, dass der Windows NT Fehlercode in niederwertigen 16 Bit dargestellt wird.

Bemerkungen

CreateUserAccount erzeugt neben dem neuen Benutzerkonto auch noch zusätzliche Benutzerrechte, die für weitere Funktionen der TwinCAT Benutzerverwaltung (z.B. CreateProcessAsUser) benötigt werden. Die zusätzlichen Benutzerrechte sind:

SE_TCB_NAME	"Als Teil des Betriebssystems handeln". Dieses Recht wird intern zum Einloggen eines Benutzers benötigt.
SE_ASSIGNPRIMARYTOKEN_NAME	"Ersetzen eines Tokens auf Prozessebene", wird für CreateProcessAsUser benötigt.
SE_INCREASE_QUOTA_NAME	"Anheben einer Quote", wird ebenfalls für CreateProcessAsUser benötigt.

4.2.8 CreateUserAccountDlg

ITcUserManager [▶ 10]::CreateUserAccountDlg

Dialog zur Generierung eines neuen Benutzerkontos.

```
HRESULT CreateUserAccountDlg()
```

Return Values

HRESULT == S_OK	Benutzerkonto wurde erfolgreich erzeugt.
HRESULT != S_OK	Im Fehlerfall steht in HRESULT ein von Windows NT generierter Fehlercode. Um eine COM/OLE Fehlerbehandlung zu provozieren, wird aus dem Fehlercode ein HRESULT durch HRESULT_FROM_NT(nErrror) erzeugt. Zu beachten ist hierbei, dass der Windows NT Fehlercode in niederwertigen 16 Bit dargestellt wird.

Bemerkungen

CreateUserAccountDlg ruft intern die Methode CreateUserAccount [▶ 14] auf.

4.2.9 EnumLocalUsers

ITcUserManager [▶ 10]::EnumLocalUsers

ITcUserManager [▶ 10]

Liefert die lokal definierten Benutzer zurück.

```
HRESULT EnumLocalUsers(
  VARIANT varComputerName,
  SAFEARRAY(BSTR)* ppUsers
);
```

Parameters

varComputerName	[in, optional] Optional kann der Name des Zielsystems als BSTR übergeben werden. Wird kein Name übergeben, so werden die Benutzer auf dem lokalen Computer aufgezählt.
ppUsers	[out, retval] Zeiger auf SAFEARRAY von BSTRs, in denen die Namen der Benutzergruppen hinterlegt sind.

Return Values

HRESULT == S_OK	Abfrage erfolgreich durchgeführt.

HRESULT != S_OK	Im Fehlerfall steht in HRESULT ein von Windows NT generierter Fehlercode. Um eine COM/OLE Fehlerbehandlung zu provozieren, wird aus dem Fehlercode ein HRESULT durch HRESULT_FROM_NT(nErrror) erzeugt. Zu beachten ist hierbei, dass der Windows NT Fehlercode in niederwertigen 16 Bit dargestellt wird.

4.2.10 EnumLocalGroups

ITcUserManager [▶ 10]::EnumLocalGroups

Liefert die lokal definierten Benutzergruppen zurück.

```
HRESULT EnumLocalGroups(
  VARIANT varComputerName,
  SAFEARRAY(BSTR)* ppGroups
);
```

Parameters

varComputerName	[in, optional] Optional kann der Name des Zielsystems als BSTR übergeben werden. Wird kein Name übergeben, so werden die Benutzer auf dem lokalen Computer aufgezählt.
ppGroups	[out, retval] Zeiger auf SAFEARRAY von BSTRs, in denen die Namen der Benutzergruppen hinterlegt sind.

Return Values

HRESULT == S_OK	Abfrage erfolgreich durchgeführt.
HRESULT != S_OK	Im Fehlerfall steht in HRESULT ein von Windows NT generierter Fehlercode. Um eine COM/OLE Fehlerbehandlung zu provozieren, wird aus dem Fehlercode ein HRESULT durch HRESULT_FROM_NT(nErrror) erzeugt. Zu beachten ist hierbei, dass der Windows NT Fehlercode in niederwertigen 16 Bit dargestellt wird.

4.2.11 UserEnumLocalGroups

ITcUserManager [▶ 10]::UserEnumLocalGroups

Liefert die Benutzergruppen zu denen ein Benutzer gehört zurück.

```
HRESULT UserEnumLocalGroups(
  VARIANT varUserName,
  SAFEARRAY(BSTR)* ppGroups
);
```

Parameters

varUserName	[in, optional] Optional kann der Name eines Benutzers als BSTR übergeben werden. Wird kein Name übergeben, so wird der Benutzer des aufrufenden Threads abgefragt.
ppGroups	[out, retval] Zeiger auf SAFEARRAY von BSTRs, in denen die Namen der Benutzergruppen hinterlegt sind.

Return Values

HRESULT == S_OK	Abfrage erfolgreich durchgeführt.

HRESULT != S_OK	Im Fehlerfall steht in HRESULT ein von Windows NT generierter Fehlercode. Um eine COM/OLE Fehlerbehandlung zu provozieren, wird aus dem Fehlercode ein HRESULT durch HRESULT_FROM_NT(nErrror) erzeugt. Zu beachten ist hierbei, dass der Windows NT Fehlercode in niederwertigen 16 Bit dargestellt wird.

4.2.12 UserIsAdmin

ITcUserManager [▶ 10]::UserIsAdmin

Prüft, ob der aufrufende Thread zu der Gruppe der Administratoren gehört

```
HRESULT UserIsAdmin(
VARIANT varUserName,
  VARIANT_BOOL* pbResult
);
```

Parameters

varUserName	[in] Optional: Name eines Benutzers, übergeben als BSTR. Zurzeit wird diese Option nicht unterstützt. Der Benutzerstatus des aufrufenden Threads wird überprüft.
pbResult	[out, retval] Zeiger auf ein VARIANT_BOOL, in dem das Abfrageergebnis hinterlegt werden soll.

Return Values

HRESULT == S_OK	Abfrage erfolgreich durchgeführt.
HRESULT != S_OK	Im Fehlerfall steht in HRESULT ein von Windows NT generierter Fehlercode. Um eine COM/OLE Fehlerbehandlung zu provozieren, wird aus dem Fehlercode ein HRESULT durch HRESULT_FROM_NT(nErrror) erzeugt. Zu beachten ist hierbei, dass der Windows NT Fehlercode in niederwertigen 16 Bit dargestellt wird.

4.2.13 UserIsMemberOf

ITcUserManager [▶ 10]::UserIsMemberOf

Überprüft, ob der aufrufende Thread Mitglied der gewünschten Benutzergruppe ist.

```
HRESULT UserIsMemberOf(
  BSTR bstrUserGroup,
  VARIANT varUserName,
  VARIANT_BOOL* pbResult
);
```

Parameters

bstrUserGroup	[in] Name der Benutzergruppe.
varUserName	[in] Optionaler Name eines Benutzers, der als BSTR übergeben wird.
pbResult	[out, retval] Zeiger auf ein VARIANT_BOOL, in dem das Abfrageergebnis hinterlegt wird.

Return Values

HRESULT == S_OK	Abfrage erfolgreich durchgeführt.

HRESULT != S_OK	Im Fehlerfall steht in HRESULT ein von Windows NT generierter Fehlercode. Um eine COM/OLE Fehlerbehandlung zu provozieren, wird aus dem Fehlercode ein HRESULT durch HRESULT_FROM_NT(nErrror) erzeugt. Zu beachten ist hierbei, dass der Windows NT Fehlercode in niederwertigen 16 Bit dargestellt wird.

4.2.14 UserName

ITcUserManager [▶ 10]::UserName

Eigenschaft (Property) zur Anzeige des aktuellen Benutzernamens.

```
HRESULT UserName(
BSTR* bstrUserName,
);
```

Parameters

bstrUserName	[out, retval] Name des Benutzerskontos, zu dem der aufrufende Thread gehört, übergeben als Zeiger auf ein BSTR Objekt.

Return Values

HRESULT == S_OK	Abfrage erfolgreich durchgeführt.
HRESULT != S_OK	Im Fehlerfall steht in HRESULT ein von Windows NT generierter Fehlercode. Um eine COM/OLE Fehlerbehandlung zu provozieren, wird aus dem Fehlercode ein HRESULT durch HRESULT_FROM_NT(nErrror) erzeugt. Zu beachten ist hierbei, dass der Windows NT Fehlercode in niederwertigen 16 Bit dargestellt wird.

Bemerkungen

UserName zeigt den Namen des Benutzerkontos an, zu dem der aufrufende Thread gehört. Dieser Name kann sich vom Namen des eingeloggten Benutzers unterscheiden.

Mehr Informationen:
www.beckhoff.com/automation

Beckhoff Automation GmbH & Co. KG
Hülshorstweg 20
33415 Verl
Deutschland
Telefon: +49 5246 9630
info@beckhoff.de
www.beckhoff.de