



# **EC/CC200**

**Engel.opc**  
**Schnittstellenbeschreibung V 1.00**

**Originalbetriebsanleitung**

**Änderungsverzeichnis**

Versionsänd. von / auf	Datum	geänderte Seiten	Beschreibung	geändert von
V1.00	21.10.2010		neu erstellt	Sg

## Inhalt

<b>1</b>	<b>Einführung</b>	<b>7</b>
<b>2</b>	<b>Konfiguration</b>	<b>7</b>
	COM Interface „IKebaOpcConfig“	7
<b>3</b>	<b>Funktionalität via OPC Items</b>	<b>9</b>
	Der Browse-Baum des E-factory.Basecom	9
	„Funktionsaufrufe“ via OPC Items	11
	Funktionalität „Alarm“	11
	Alarmer lesen	12
	Alarmer quittieren	15
	Alarminformation abfragen	16
	Alarmdefinitionen abfragen	17
	Funktionalität „InfoLog“	18
	Einträge lesen	18
	Funktionalität „MultiLanguageSupport“	20
	Client-spezifische Sprache setzen	21
	MultiLanguageSupport synchronisieren	22
	Erzeugen von sprachabhängigen Alarm-Texten	23
	Erzeugen von sprachabhängigen Einheiten-Texten	25
	Erzeugen von sprachabhängigen Variablen, Lang- und Kurz-Texten	27
	Funktionalität „User“	29
	Bestätigung eines Login-Vorganges	29
	Benutzer hinzufügen	30
	Anmeldeversuch abbrechen	32
	Liste anstehender Anmeldeversuche	32
	Aktueller Status des Anmeldevorgangs	33
	Information über aktuelle Sitzung	33
	Information über bestehende Sitzungen	34
	Attribute eines Benutzers	35
	Benutzer-Konfiguration	36
	Registrierte Benutzernamen ermitteln	37
	Sitzungen mit Schreibzugriff ermitteln	38
	Remote/Lokal-Abfrage	38
	Passwort prüfen	39
	Benutzer anmelden	39
	Benutzer abmelden	40
	Benutzer abmelden und Profil ändern	41
	Passwort ändern	42
	Anwender ändern	43
	Benutzer löschen	44
	Sitzungen beenden	45
<b>4</b>	<b>Format der OpcSvc-INI-Datei</b>	<b>46</b>

**5    Keep Alive..... 47**

**6    Fehlercodes..... 48**





# 1 Einführung

Die Schnittstellenbeschreibung umfasst:

- die Beschreibung des in e-factory.Basecom zusätzlich verfügbaren COM Interfaces „IKebaOpcConfig“
- die Beschreibung der Struktur und der Funktionalität aller in e-factory.Basecom enthaltenen OPC Items

# 2 Konfiguration

## COM Interface „IKebaOpcConfig“

Das COM Konfigurationsinterface besteht aus den Methoden

```
HRESULT Login      ([in, string] LPCWSTR apUsername,  
                    [in, string] LPCWSTR apPassword);  
HRESULT Logout     ();  
HRESULT AddSystem   ([in, string] LPCWSTR apSynonym,  
                    [in, string] LPCWSTR apHostname);  
HRESULT RemoveSystem ([in, string] LPCWSTR apSynonym);
```

Siehe dazu die Interfacedatei KebaOPCAddOns.idl.

Mit den Methoden `AddSystem()` und `RemoveSystem()` können EC/CC200 Steuerungen zur Konfiguration eines OPC Clients hinzugefügt und wieder entfernt werden. Der Parameter `apHostname` identifiziert dabei die Netzwerkadresse der Steuerung entweder als IP-Adresse oder als Hostnamen.

### **Hinweis**

*Jene IP Adresse oder Hostname wird in der folgenden Beschreibung der einzelnen Funktionalitäten nur mehr als `<hostname>` bezeichnet.*

Alternativ dazu können auch Variablengruppen angegeben werden, indem nach dem Hostnamen (durch Beistriche getrennt) die Namen der gewünschten Variablengruppe (oder mehrerer Variablengruppen) angegeben wird.

### **Beispiel:**

```
AddSystem (L"localhost,PartDisk,Host");  
Rechnername:      localhost  
2 Variablengruppen: PartDisk, Host
```

**Aufruf von AddSystem():**

Beim Aufruf von `AddSystem()` wird nach erfolgreichem Verbindungsaufbau der Systemstatus der Steuerung überprüft. Sind zu diesem Zeitpunkt noch nicht alle Systemvariablen verfügbar, wartet das System eine definierte Zeit. Diese Zeit kann über `SystemStateWaitTimeout` (Section "General", Zeit in Millisekunden, Default-Wert: 0) eingestellt werden.

In dieser Zeit wird in definierten Intervallen die Verfügbarkeit der Systemvariablen geprüft. Die Anzahl der Intervalle kann über `SystemStateWaitPollInterval` (Section "General", Zeit in Millisekunden, Default-Wert: 1000) verändert werden.

Wenn nach Ablauf der definierten Zeit in `SystemStateWaitTimeout` die Steuerung keinen Status hat, in dem die Systemvariablen verfügbar sind, kehrt `AddSystem()` mit einer Fehlermeldung zurück.



### 3 Funktionalität via OPC Items

In diesem Kapitel werden sämtliche im unten gezeigten Browse-Baum angeführten Funktionalitäten mit den jeweiligen IN und OUT Parametern beschrieben.

#### Der Browse-Baum von e-factory.Basecom

```
+<hostname I> Browse-Baum der Steuerung <hostname I>
-<hostname II> Browse-Baum der Steuerung <hostname II>
  -Funcs
    -Alarm für Details siehe Kapitel "Alarm"
      + AckAlarm()
      + GetAlarmInfo()
      + GetAllDefinedAlarms()
      + GetDefinedAlarms()
      + ReadAlarms()
    -InfoLog für Details siehe Kapitel "InfoLog"
      + ReadEntries()
    -MultiLanguageSupport für Details siehe Kapitel "MultiLanguageSupport"
      + GenerateAlarmText()
      + GetAlarmDefinition()
      + GetUnitInfo()
      + GetPureSVText()
      + GetSVText()
      + InitAlarmDefinitionContainer()
      + InitUnitDefinitionContainer()
      + InitSVTextDefinitionContainer()
      + SetDefaultLanguageId()
      + SynchronizeMultiLanguageSupport()
    -User für Details siehe Kapitel "User"
      + AckLoginRequest()
      + AddUser()
      + CancelLogin()
      + GetLoginRequests()
      + GetLoginStatus()
      + GetSession()
      + GetSessions()
      + GetUserAttributes()
      + GetUserConfig()
      + GetUserNames()
      + GetWriteAccessSessions()
      + IsLocalStation()
      + IsValidPassword()
      + LogIn()
      + LogOut()
      + LogOutAndModify()
      + ModifyPassword()
      + ModifyUser()
      + RemoveUser()
      + TerminateUserSessions()
  -SVs
    +FU1 Browse-Baum der Systemvariablen der FU1 auf
    <hostname II>
    +FU2
```

**Hinweis**

*Jeder Funktionsaufruf beinhaltet IN- und OUT-Parameter, die beschrieben bzw. gelesen werden müssen.*

## „Funktionsaufrufe“ via OPC Items

Die allgemeine Verwendung von OPC-Items soll an folgendem Beispiel der Funktion `AckAlarm()` erklärt werden.

- **Eingabeparameter:**  
Das OPC Item `<hostname>.Funcs.Alarm.AckAlarm().In.Id.MainId` auf den gewünschten Wert setzen. Analog dazu `SubId1` bis `SubId4` beschreiben.
- **Funktionsaufruf:**  
Die Funktion `<hostname>.Funcs.Alarm.AckAlarm()` auslösen, indem sie mit einem beliebigen Wert beschrieben wird.
- **Ausgabeparameter:**  
Das Funktionsergebnis durch das Lesen des OPC Items `<hostname>.Funcs.Alarm.AckAlarm().Out.Error` abfragen.

Weitere Details: Siehe `OpcTestClient\Source\OpcTestClient.cpp`.  
(Teil des `OpcTestClient` Projektes.)

## Funktionalität „Alarm“

Alarmer können auf einer EC/CC200 Steuerung mit Hilfe der folgenden Funktionen gelesen und quittiert werden:

- `AckAlarm()`,
- `GetAllDefinedAlarms()`,
- `GetDefinedAlarms()`,
- `GetAlarmInfo()` und
- `ReadAlarms()`.

## Alarmer lesen

<code>&lt;hostname&gt;.Funcs.Alarm.ReadAlarms (↑Alarms[], ↑Error)</code>	
OUT:	<pre> Alarms[]   AckMode : INT   Class : INT   Logging : BOOL   MainId : INT   Name : STRING   Param1 : VARIANT   Param2 : VARIANT   State : INT   SubId1 : INT   SubId1Used : BOOL   SubId2 : INT   SubId2Used : BOOL   SubId3 : INT   SubId3Used : BOOL   SubId4 : INT   SubId4Used : BOOL   Text : STRING   TimeStamp : STRING (DATE) Error : INT </pre>

Mit dieser Funktion können die an einer EC/CC200 Steuerung eventuell anstehenden Alarmer gelesen werden. Die Funktion wird durch das Beschreiben des OPC Items `ReadAlarms()` aufgerufen.

Nach erfolgreichem Aufruf werden durch e-factory.Basecom zwei Ausgangsparameter beschrieben: `Alarms[]` und `Error.Alarms[]` ist ein Feld von Strukturen mit folgenden Einträgen:

- **AckMode** Die Quittierungsart des Alarms kann einen der folgenden Werte annehmen:  
55 ... USER,  
66 ... APPLICATION,  
77 ... USER AND APPLICATION.
- **Class** Die Alarmklasse ist ein Wert zwischen 1 und 10.
- **Logging** Bei `true` wurde der Alarm in ein Protokoll (InfoLog Datei) eingetragen.
- **MainId** Einer der fünf einen Alarm eindeutig identifizierenden Werte. Dieser Wert wird von der Steuerung vergeben. Er muss für die Funktion `AckAlarm()` verwendet werden.
- **Name** Der Name des Alarms.
- **Param1** Ein zusätzlicher, optionaler Parameter.
- **Param2** Ein zusätzlicher, optionaler Parameter.
- **(Param3 – Param8)** Zusätzliche, optionaler Parameter für Alarmklassen 7 - 10.
- **State** Der Alarmzustand kann einen d. folgenden Werte annehmen:  
1 ... DEACTIVATED,  
2 ... ACKED,  
3 ... ACTIVATED.

- **SubId1** Einer der fünf einen Alarm eindeutig identifizierenden Werte. Dieser Wert wird von der Steuerung vergeben. Er muss für die Funktion `AckAlarm()` verwendet werden.
- **SubId1Used** Dieser Wert wird von der Steuerung vergeben und gibt an, ob SubId1 für den Alarm benutzt wurde.
- **SubId2** Einer der fünf einen Alarm eindeutig identifizierenden Werte. Dieser Wert wird von der Steuerung vergeben. Er muss für die Funktion `AckAlarm()` verwendet werden.
- **SubId2Used** Dieser Wert wird von der Steuerung vergeben und gibt an, ob SubId2 für den Alarm benutzt wurde.
- **SubId3** Einer der fünf einen Alarm eindeutig identifizierenden Werte. Dieser Wert wird von der Steuerung vergeben. Er muss für die Funktion `AckAlarm()` verwendet werden.
- **SubId3Used** Dieser Wert wird von der Steuerung vergeben und gibt an, ob SubId3 für den Alarm benutzt wurde.
- **SubId4** Einer der fünf einen Alarm eindeutig identifizierenden Werte. Dieser Wert wird von der Steuerung vergeben. Er muss für die Funktion `AckAlarm()` verwendet werden.
- **SubId4Used** Dieser Wert wird von der Steuerung vergeben und gibt an, ob SubId4 für den Alarm benutzt wurde.
- **Text** Generierter Alarmtext in der gewünschten Sprache inklusive der verwendeten SubIds und Parameter. Auch Textreferenzen werden ersetzt. Die Sprache des Alarmtextes kann mit Hilfe der MultiLanguageSupport-Funktionalität `SetDefaultLanguageId()` Client-spezifisch geändert werden.  
 Wird keine Sprache gesetzt, so wird die Default-Sprache verwendet, die im OPCSvc.ini-File, Section `MultiLanguageSupport`, Key `DefaultLanguageId` spezifiziert ist. Diese Default-Sprache gilt für alle Clients. Kann der Eintrag im ini-File nicht gefunden werden, wird als `DefaultLanguageId` "DE" angenommen.  
 Sollen die Text-Dateien einer anderen Sprache (Properties-Files) für die Alarm-Text-Generierung verwendet werden, muß ein OPC-Client die MultiLanguageSupport-Funktionalität `SynchronizeMultiLanguageSupport()` aufrufen. Bei diesem Aufruf werden alle properties-files in den OPC-Server-Cache kopiert und dann beim nächsten Zugriff verwendet. Für Details siehe dazu die Beschreibung der Funktionalität `MultiLanguageSupport.SynchronizeMultiLanguageSupport()`.
- **TimeStamp** Der Zeitpunkt des Auftretens des Alarms. Wenn dieses OPC Item als Typ `VT_BSTR` (String) abgefragt wird, so wird das Datum und die Uhrzeit im IEC Format „DT#yyyy-mm-dd-hh:mm:ss.uuuuuu“ geliefert.

- Das OPC Item `Error` gibt über einen eventuell aufgetretenen Fehler Auskunft. Die folgenden Fehlercodes sind definiert:

Fehlercodes:	Beschreibung
-2	OUT_OF_MEMORY
-1	INTERNAL_Error
0	OK
1	INVALID_ID
2	INVALID_NAME

## Alarmer quittieren

<hostname>.Funcs.Alarm.AckAlarm (↓ID[], ↑Error)	
IN:	ID[] MainId : INT SubId1 : INT SubId1Used : BOOL SubId2 : INT SubId2Used : BOOL SubId3 : INT SubId3Used : BOOL SubId4 : INT SubId4Used : BOOL
OUT:	Error : INT

Mit dieser Funktion können an einer EC/CC200 Steuerung eventuell anstehende Alarmer – die zuvor mit der Funktion `ReadAlarms()` gelesen wurden – quittiert werden. Die Funktion wird durch das Beschreiben der OPC Items

- Id und
- `AckAlarm()`

aufgerufen. Nach erfolgreichem Aufruf wird durch e-factory.Basecom der Ausgangsparameter `Error` beschrieben. Der Eingangsparameter `ID` ist eine Struktur bestehend aus den Elementen `MainId`, `SubId1`, `SubId1Used`, `SubId2`, `SubId2Used`, `SubId3`, `SubId3Used`, `SubId4` und `SubId4Used`. Für eine Beschreibung dieser Elemente siehe unter „Alarmer lesen“. Ebenso für den Ausgangsparameter `Error`.

## Alarminformation abfragen

<hostname>.Funcs.Alarm.GetAlarmInfo (↑ALARM[], ↑Error, ↑PENDING)	
OUT:	ALARM[] AckMode : INT Class : INT Logging : BOOL MainId : INT Name : STRING Param1 : VARIANT Param2 : VARIANT State : INT SubId1 : INT SubId1Used : BOOL SubId2 : INT SubId2Used : BOOL SubId3 : INT SubId3Used : BOOL SubId4 : INT SubId4Used : BOOL Text : STRING TimeStamp : STRING (DATE) Error : INT PENDING : INT

Mit dieser Funktion kann die Anzahl der an einer EC/CC200 Steuerung eventuell anstehenden Alarme ermittelt werden. Die Funktion wird durch das Beschreiben des OPC Items `GetAlarmInfo()` aufgerufen.

Nach erfolgtem Aufruf werden durch e-factory.Basecom die Ausgangsparameter `Alarm`, `Pending` und `Error` beschrieben. Der Ausgangsparameter `Pending` gibt an, wie viele Alarme gerade an der EC/CC200 Steuerung anstehen. Besitzt dieser Parameter einen Wert größer gleich eins, so beschreibt der Ausgangsparameter `Alarm` den ersten anstehenden Alarm.

Auch hier gilt: Für die Alarmklassen 7 – 10 stehen weitere Parameter (`Param3` – `Param8`) zur Verfügung.

Für eine Beschreibung dieses Parameters und für den Parameter `Error` siehe unter „Alarme lesen“.



## Alarmdefinitionen abfragen

<hostname>.Funcs.Alarm.GetAllDefinedAlarms (↑Alarms, ↑Error)	
OUT:	Alarms : STRING (ARRAY) Error : INT

  

<hostname>.Funcs.Alarm.GetDefinedAlarms (↓FUNAME, ↑Alarms, ↑Error)	
IN:	FUNAME : STRING
OUT:	Alarms : STRING (ARRAY) Error : INT

Mit diesen Funktionen können alle definierten Alarme der Steuerung ausgelesen werden.

Bei der Funktionalität `Alarm.GetAllDefinedAlarms()` werden alle definierten Alarme aller Maschineneinheiten (FUs) von der Steuerung ausgelesen und deren Namen zurückgeliefert. Die Namen setzen sich dabei aus FU-Namen und Alarm-Namen zusammen. Beispiel: ["FU1.Alarm1" "FU1.Alarm2" "FU2.Alarm1"].

Bei der Funktionalität `Alarm.GetDefinedAlarms()` werden alle definierten Alarme der spezifizierten FU von der Steuerung ausgelesen und deren Namen zurückgeliefert. Beispiel: ["Alarm1" "Alarm2"].

Das OPC Item `Error` der Funktionalität `Alarm.GetAllDefinedAlarms()` gibt über einen eventuell aufgetretenen Fehler Auskunft. Die folgenden Fehlercodes sind definiert:

Fehlercodes:	Beschreibung
0	OK
≠0	Error (siehe WinAPI Funktion <code>GetLastError</code> und Kap. "Fehlercodes")

Das OPC Item `Error` der Funktionalität `Alarm.GetDefinedAlarms()` gibt über einen eventuell aufgetretenen Fehler Auskunft. Die folgenden Fehlercodes sind definiert:

Fehlercodes:	Beschreibung
0	OK
2	FU Name unbekannt
≠0	Error (siehe WinAPI Funktion <code>GetLastError</code> und Kap. "Fehlercodes")

## Funktionalität „InfoLog“

Das InfoLog-Protokoll einer EC/CC200 Steuerung kann mit Hilfe der folgenden Funktion gelesen werden:

- `ReadEntries()`.

### Einträge lesen

<code>&lt;hostname&gt;.Funcs.InfoLog.ReadEntries</code> <code>(↓Count, ↓Filter, ↓Index, ↑Entries[], ↑Error)</code>	
IN:	Count : INT Filter : INT Index : INT
OUT:	Entries[] Class : STRING CPU : INT Cycle : DECIMAL Index : INT TimeStamp : STRING Error : INT

Mit dieser Funktion werden alle noch nicht vom aktuellen OPC-Client gelesenen Einträge des Protokolls einer EC/CC200 Steuerung gelesen. Die Funktion wird durch das Beschreiben des OPC Items `ReadEntries()` aufgerufen.

Es werden immer maximal 100 Einträge gelesen. Beim ersten Aufruf des OPC-Clients werden die ersten 100 Infolog-Einträge gelesen, bei nächstem Aufruf die nächsten 100 Einträge, usw.

Wenn mehrere OPC-Clients mit einem OPC-Server verbunden sind, beeinflussen sich die OPC-Clients gegenseitig nicht, wenn sie `InfoLog.ReadEntries()` aufrufen. (Jeder Client bekommt grundsätzlich alle Infolog-Einträge, die von der Steuerung geliefert werden.)

Die Funktion wird durch das Beschreiben der OPC Items `Count`, `Filter` und `Index` aufgerufen. Der Parameter `Count` gibt die Anzahl der zu liefernden InfoLogs an, der Parameter `Filter` die Filter-Bits (Beschreibung siehe InfoLog-Klassen). Der Parameter `Index` gibt den Startindex an, ab dem Infolog ausgelesen werden soll (Start bei 0: alle verfügbaren Infologs).

Nach erfolgreichem Aufruf werden durch e-factory.Basecom die Ausgangsparameter `Entries[]` und `Error` beschrieben. `Entries[]` ist ein Feld von Strukturen der Form

- `Class` Die Klasse des Eintrags

### InfoLog-Klassen:

ILOG-Typ	Filter-Bit	Beschreibung
ILOG_SYSTEM_MESSAGE	1	Systemereignisse
ILOG_PCB_OPERATION	2	Printed Circuit Board
ILOG_Error_INFO	4	Error Info
ILOG_Error_WARNING	8	Fehler vom Typ Warning
ILOG_Error_Error	16	Fehler vom Typ Error
ILOG_Error_CRITICAL	32	Fehler vom Typ Critical
ILOG_Error_FATAL	64	Fehler vom Typ Fatal
ILOG_ALARM_CLASS1	128	Alarmeintrag Klasse 1
ILOG_ALARM_CLASS2	256	Alarmeintrag Klasse 2
ILOG_ALARM_CLASS3	512	Alarmeintrag Klasse 3
ILOG_ALARM_CLASS4	1024	Alarmeintrag Klasse 4
ILOG_ALARM_CLASS5	2048	Alarmeintrag Klasse 5
ILOG_ALARM_CLASS6	4096	Alarmeintrag Klasse 6
ILOG_ALARM_CLASS7	8192	Alarmeintrag Klasse 7
ILOG_HMI_OPERATION	16384	HMI Manipulation SV Änderungen
ILOG_FILE_OPERATION	32768	Dateioperationen
ILOG_SPS_OPERATION	65536	SPS
ILOG_ONLINE_OPERATION	131072	Online
ILOG_OPERATING_MODE_CHANGE	262144	Betriebszustandsänderungen
ILOG_USER_MANAGEMENT	524288	Benutzerverwaltung
ILOG_USER_MESSAGE	1048576	Benutzer-Ereignis
ILOG_USER_0	2097152	Benutzerdefinierter Eintrag
ILOG_USER_1	4194304	Benutzerdefinierter Eintrag
ILOG_PDP_MESSAGE	8388608	PDP-Eintrag
ILOG_UNKNOWN_TYPE	16777216	Unbekannte Klasse

- `Cycle` Der Produktionszyklus des Eintrags.
- `Index` Der von der Steuerung vergebene Index für diesen Eintrag.
- `TimeCode` ZeitCode der Aufrufe der oben angeführten Klassen.
- `TimeStamp` Der Zeitpunkt der Aufnahme dieses Eintrags in das Protokoll.  
Es wird das Datum und die Uhrzeit im IEC Format „DT#yyyy-mm-dd-hh:mm:ss.uuuuuu“ geliefert.

Das OPC Item `Error` gibt über einen eventuell aufgetretenen Fehler Auskunft. Die folgenden Fehlercodes sind definiert:

Fehlercodes:	Beschreibung
-2	OUT OF MEMORY
-1	INTERNAL Error
0	OK

## Funktionalität „MultiLanguageSupport“

Sprachabhängige Texte für Alarme und Systemvariablen (Lang-/Kurz-Text) werden grundsätzlich automatisch vom OPC-Server zur Verfügung gestellt. Um diese Funktionalität erfüllen zu können, werden die auf der Steuerung gespeicherten properties-files (eventuell gepackt in jar-files) verwendet.

### **Hinweis**

*In der Systemkonfiguration (system.cfg) muss der OPC-Server.classPath eingetragen werden, da ansonsten das Synchronisieren des MultiLanguageSupport fehlschlägt.*

*Der OPC-Server liest daraufhin den Catalog-Eintrag aus, lädt die JAR-Files von der Steuerung auf den lokalen Rechner und extrahiert die .properties Files.*

*Beispiel für den Konfigurationseintrag in der system.cfg:*

```
[OPCServer]
classPath = "/harddisk0/application/view/language.jar"
```

Die **Alarm-Texte** werden automatisch generiert: Siehe dazu die Beschreibungen zu den Funktionalitäten `Alarm.ReadAlarms()` und `Alarm.GetAlarmInfo()`. Die **SV-Kurz- und -Lang-Texte** sind automatisch für jede Variable als Properties verfügbar:

- KOPC\_PROP\_OPC\_TextLong
- KOPC\_PROP\_OPC\_TextShort

Die zur Generierung sprachabhängiger Texte benötigten properties-files werden nicht ständig direkt von der Steuerung gelesen (aus Performancegründen), sondern in einem Cache unterhalb des Installationsverzeichnis des OPC-Servers gehalten. Je SPS, die der OPC-Server visualisiert und je Steuerungs-Programm wird ein eigener Cache angelegt. Dazu ist es notwendig, daß der OPC-Server auf dieser Ebene uneingeschränkte Schreib- und Lese-Rechte besitzt.

Der Cache wird automatisch vom OPC-Server verwaltet und beim ersten Zugriff auf MultiLanguageSupport-Funktionalität gefüllt. Auch properties-files, die in jar-files gepackt sind, werden in diesem Zuge in den Cache extrahiert. Der Cache wird unabhängig von der aktuell eingestellten `LanguageId` gefüllt, das heißt, es werden *alle* properties-files des Steuerungsprogramms im Cache gespeichert. Ist der Cache des OPC-Servers für eine Steuerung einmal gefüllt, muß sich ein OPC-Client bei Bedarf um die Synchronisation der Daten kümmern. Die Synchronisation ist Client-übergreifend und nur notwendig, wenn sich properties-files des Steuerungs-Programms ändern. Es genügt daher, wenn *ein* Client diese Synchronisation durchführt. Andere OPC-Clients, die über den selben OPC-Server die betroffene Steuerung visualisieren, werden über die Synchronisation automatisch informiert; der OPC-Server bzw. der Client müssen dazu nicht erneut gestartet werden.

Für die Synchronisation steht folgende Funktion zur Verfügung:

- `SynchronizeMultiLanguageSupport()`

Um eine Client-spezifische Sprache für die Texte einzustellen, muß folgende Funktion verwendet werden:

- `SetDefaultLanguageId()`

Sollte es notwendig sein, sprachabhängigen Texten für Alarme und Lang-/Kurz-Texte manuell zu generieren, stehen folgenden Funktionen zur Verfügung:

- `GenerateAlarmText()`,
- `GetAlarmDefinition()`,
- `GetPureSVText()`,
- `GetSVText()`,
- `GetUnitInfo()`,
- `InitAlarmDefinitionContainer()`,
- `InitSVTextDefinitionContainer()`.
- `InitUnitDefinitionContainer()`,

Bevor die Funktionen `GenerateAlarmText()` und `GetAlarmDefinition()` aufgerufen werden können, muß die Initialisierungsfunktion `InitAlarmDefinitionContainer()` aufgerufen werden.

Bevor die Funktionen `GetSVText()` und `GetPureSVText()` aufgerufen werden können, muß die Initialisierungsfunktion `InitSVTextDefinitionContainer()` aufgerufen werden.

Bevor die Funktion `GetUnitInfo()` aufgerufen werden kann, muß die Initialisierungsfunktion `InitUnitDefinitionContainer()` aufgerufen werden.

Diese Funktionen beeinflussen die oben erwähnten Standard-Funktionen für die automatische Alarm-Text-Generierung bzw. die SV-Text-Properties für Lang-/Kurz-Text nicht. Es können in diesem Fall ganz andere properties-files und auch eine andere Sprache definiert werden.

## Client-spezifische Sprache setzen

<code>&lt;hostname&gt;.Funcs.MultiLanguageSupport.SetDefaultLanguageId</code> (↓LanguageID)	
IN:	LanguageID : STRING

Mit dieser Funktion kann die Sprachkennung für den `MultiLanguageSupport` bei SV-Properties (Lang-/Kurz-Text) und bei den Alarm-Texten in der Alarm-Funktionalität neu gesetzt werden.

Der Initialzustand der Sprachkennung wird aus dem `OPCSvc.ini`-File gelesen:

Section `MultiLanguageSupport`, Key `DefaultLanguageId`

Kann der Eintrag nicht gefunden werden, wird als `DefaultLanguageId` "DE" angenommen.

Nach dem Einstellen der `DefaultLanguageId` ist es nicht notwendig den Cache-Datenbestand zu synchronisieren, da im Cache *alle* properties-files gehalten werden, unabhängig von der aktuell gesetzten Sprache.

## MultiLanguageSupport synchronisieren

```
<hostname>.Funcs.MultiLanguageSupport.SynchronizeMultiLanguageSupport  
(↑Error)  
OUT: Error : INT
```

Mit dieser Funktion kann der Cache-Datenbestand für das aktuelle SPS-System aktualisiert werden. Dabei werden die für den MultiLanguageSupport relevanten Daten (u.a. properties-files) von der Steuerung in den Cache transferiert. Der zum aktuellen SPS-System gehörige, nun veraltete Cache-Datenbestand wird vollständig gelöscht.

### **Hinweis**

*Der Datentransfer kann je nach SPS-und Netzwerk-Performance relativ lange dauern (im Minutenbereich)!  
Ein OPC-Client soll daher nicht standardmäßig diese Funktion beim Verbindungsaufbau zum OPC-Server aufrufen, sondern nur dann, wenn sich das Steuerungsprogramm geändert hat. Sollten mehrere OPC-Clients mit diesem OPC-Server arbeiten, genügt es, wenn **ein** Client die Synchronisation übernimmt. Die anderen Clients werden automatisch informiert. Ein Neustart des OPC-Servers oder der OPX-Clients ist nicht notwendig.*

Es werden bei der Synchronisation unabhängig von deren Spracherkennung alle properties-files transferiert.

## Erzeugen von sprachabhängigen Alarm-Texten

<hostname>.Funcs.MultiLanguageSupport.InitAlarmDefinitionContainer (↓LanguageID, ↓ProgramRoot, ↑Error)	
IN:	LanguageID : STRING ProgramRoot : STRING FirmwareErrorTextFile : STRING
OUT:	Error : INT

<hostname>.Funcs.MultiLanguageSupport.GenerateAlarmText (↓AlarmName, ↓Param1, ↓Param2, ↓SubId1, ↓SubId1Used, ↓SubId2, ↓SubId2Used, ↓SubId3, ↓SubId3Used, ↓SubId4, ↓SubId4Used, ↑AlarmText, ↑Error,)	
IN:	AlarmName : STRING Param1 : VARIANT Param2 : VARIANT SubId1 : INT SubId1Used : BOOL SubId2 : INT SubId2Used : BOOL SubId3 : INT SubId3Used : BOOL SubId4 : INT SubId4Used : BOOL
OUT:	AlarmText : STRING Error : INT

<hostname>.Funcs.MultiLanguageSupport.GetAlarmDefinition (↓AlarmName, ↑AlarmDefinition, ↑Error,)	
IN:	AlarmName : STRING
OUT:	AlarmDefinition : STRING Error : INT

Mit diesen Funktionen können sprachabhängige Alarm-Texte manuell generiert werden. Die sprachabhängigen Teile der Alarme sind in properties-files auf der Steuerung gespeichert.

Die Funktionen werden durch das Beschreiben der OPC Items

InitAlarmDefinitionContainer(), GenerateAlarmText() oder  
GetAlarmDefinition() aufgerufen.

Mit der Funktion InitAlarmDefinitionContainer() wird der OPC-Server-interne Alarm-Definition-Container initialisiert. Dabei wird die Sprache festgesetzt und auch das Root-Verzeichnis der zur Steuerung gehörigen properties-files spezifiziert, weiters wird noch die Datei, welche die Fehlermeldungen enthält, benötigt. Die Funktion kann mehrfach aufgerufen werden. Dies ist notwendig, wenn man die Alarm-Texte in einer anderen Sprache bekommen will oder sich die properties-files auf der Steuerung geändert haben.

### **Hinweis**

*Die Funktion kann **keine** jar-Files verarbeiten!*

Wird als `LanguageId` ein Leerstring übergeben, werden alle properties files (\*.properties) gescannt und in den Container aufgenommen. Dabei kann es zu unerwünschten Nebeneffekten kommen, wenn verschiedensprachige properties files gefunden werden. (z.B. werden auch \*\_DE.properties-files oder \*\_EN.properties-files gescannt). Wird als `LanguageId` der String "DE" übergeben, werden nur alle properties files gescannt die der Wildcard "\*\_DE.properties" entsprechen.

Nach Aufruf der beiden anderen Funktionen werden durch e-factory.Basecom die Ausgangsparameter `AlarmText` bzw. `AlarmDefinition` und `Error` beschrieben.

`AlarmDefinition` ist der in properties-files definierte Alarm-Text ohne Substituierungen (%i, %n, etc. werden nicht ersetzt), aber in der gewählten Sprache.

`AlarmText` ist der in properties-files definierte Alarm-Text mit Substituierungen und aufgelösten Textreferenzen in der gewählten Sprache.

Das OPC Item `Error` gibt für `InitAlarmDefinitionContainer()` über einen eventuell aufgetretenen Fehler Auskunft. Die folgenden Fehlercodes sind definiert:

Fehlercodes:	Beschreibung
0	OK
1	Error: DRIVE NOT FOUND WARNING: PATH CONTAINS NO FUS
2	Error: PATH NOT FOUND WARNING: NO PROPERTY FILES FOUND
3	Error: PATH CONTAINS NO DIRS WARNING: TEXT REFERENCE FILE NOT FOUND

Mit der Funktion `GenerateAlarmText()` kann ein dynamischer Alarmtext generiert werden. Für die Alarmklassen 7 – 10 (siehe `ReadAlarms()`) können noch noch zusätzliche Parameter (`Param3` – `Param8`) angegeben werden. Mit der Funktion `GetAlarmDefinition()` wird die Alarmdefinition ermittelt.

Das OPC Item `Error` gibt in beiden Fällen über einen eventuell aufgetretenen Fehler Auskunft. Die folgenden Fehlercodes sind definiert:

Fehlercodes:	Beschreibung
0	OK
1	Alarmname nicht gefunden
2	Alarmdefinition leer



## Erzeugen von sprachabhängigen Einheiten-Texten

<hostname>.Funcs.MultiLanguageSupport.InitUnitDefinitionContainer (↓LanguageID, ↓ProgramRoot, ↑Error)	
IN:	LanguageID : STRING ProgramRoot : STRING
OUT:	Error : INT

  

<hostname>.Funcs.MultiLanguageSupport.GetUnitInfo (↓FuName, ↓UnitName, ↑UnitInfo, ↑Error,)	
IN:	FuName : STRING UnitName : STRING
OUT:	UnitInfo: STRING Error : INT

Mit der Funktion `InitUnitDefinitionContainer()` wird der OPC-Server-interne Einheiten-Definition-Container initialisiert. Dabei wird die Sprache festgesetzt und auch das Root-Verzeichnis der zur Steuerung gehörigen properties-files spezifiziert. Die Funktion kann mehrfach aufgerufen werden. Dies ist notwendig, wenn man die Alarm-Texte in einer anderen Sprache bekommen will oder sich die properties-files auf der Steuerung geändert haben.

### **Hinweis**

*Die Funktion kann **keine** jar-Files verarbeiten!*

Wird als `LanguageId` ein Leerstring übergeben, werden alle properties files (\*.properties) gescannt und in den Container aufgenommen. Dabei kann es zu unerwünschten Nebeneffekten kommen, wenn verschiedensprachige properties files gefunden werden. (z.B. werden auch \*\_DE.properties-files oder \*\_EN.properties-files gescannt). Wird als `LanguageId` der String "DE" übergeben, werden nur alle properties files gescannt die der Wildcard "\*\_DE.properties" entsprechen.

Nach Aufruf der Funktion `GetUnitInfo()` werden durch e-factory.Basecom die Ausgangsparameter `UnitInfo` und `Error` beschrieben.

`UnitInfo` ist der in properties-files definierte Einheiten-Text ("`<iso>|<imperial>|<text>`"); z.B.: "`mm|in`") in der gewählten Sprache.

Das OPC Item `Error` gibt für `InitUnitDefinitionContainer()` über einen eventuell aufgetretenen Fehler Auskunft. Die folgenden Fehlercodes sind definiert:

Fehlercodes:	Beschreibung
0	OK
1	Error: DRIVE NOT FOUND WARNING: PATH CONTAINS NO FUS
2	Error: PATH NOT FOUND WARNING: NO PROPERTY FILES FOUND
3	Error: PATH CONTAINS NO DIRS

Das OPC Item `Error` gibt für `GetUnitInfo()` über einen eventuell aufgetretenen Fehler Auskunft. Die folgenden Fehlercodes sind definiert:

<b>Fehlercodes:</b>	<b>Beschreibung</b>
0	OK
1	SVName nicht gefunden
-1	Formatierungsfehler bei SVText-Definition
2	SVText-Definition leer

## Erzeugen von sprachabhängigen Variablen, Lang- und Kurz-Texten

<hostname>.Funcs.MultiLanguageSupport.InitSVTextDefinitionContainer (↓LanguageID, ↓ProgramRoot, ↑Error)	
IN:	LanguageID : STRING ProgramRoot : STRING
OUT:	Error : INT

  

<hostname>.Funcs.MultiLanguageSupport.GetSVText (↓SVName, ↑Error, ↑SVTextLong, ↑SVTextShort)	
IN:	SVName : STRING
OUT:	Error : INT SVTextLong : STRING SVTextShort : STRING

  

<hostname>.Funcs.MultiLanguageSupport.GetPureSVText (↓SVName, ↑Error, ↑PureSVText)	
IN:	SVName : STRING
OUT:	Error : INT PureSVText : STRING

Mit diesen Funktionen können sprachabhängige Lang- und Kurz-Texte zu SV-Variablen generiert werden. Die sprachabhängigen Lang- und Kurz-Texte sind in properties-files auf der Steuerung gespeichert.

Die Funktionen werden durch das Beschreiben der OPC Items  
InitSVTextDefinitionContainer(), GetSVText() oder  
GetPureSVText() aufgerufen.

Mit der Funktion InitSVTextDefinitionContainer() wird der OPC-Server-interne SVText-Definition-Container initialisiert. Dabei wird die Sprache festgesetzt und auch das Root-Verzeichnis zur Steuerung gehörigen properties-files spezifiziert. Die Funktion kann mehrfach aufgerufen werden. Dies ist notwendig, wenn man die SV-Lang- und -Kurz-Texte in einer anderen Sprache bekommen will oder sich die properties-files auf der Steuerung geändert haben.

### **Hinweis**

*Die Funktion kann **keine** jar-Files verarbeiten!*

Wird als LanguageId ein Leerstring übergeben werden alle properties files (\*.properties) gescannt und in den Container aufgenommen. Dabei kann es zu unerwünschten Nebeneffekten kommen, wenn verschiedensprachige properties files gefunden werden. (Z.B. werden auch \*\_DE.properties-files oder \*\_EN.properties-files gescannt.) Wird als LanguageId der String "DE" übergeben, werden nur alle properties files gescannt die der Wildcard "\*\_DE.properties" entsprechen.

Nach Aufruf der beiden anderen Funktionen werden durch e-factory.Basecom die Ausgangsparameter SVTextLong, SVTextShort bzw. PureSVText und Error beschrieben.

PureSVText ist der in properties-files definierte Lang- und Kurz-Text inkl.

Trennzeichen zwischen Lang- und Kurz-Text in der gewählten Sprache.

SVTextLong ist der Lang-Text zur spezifizierten SV-Variable in der gewählten Sprache, SVTextShort der Kurz-Text.

Das OPC Item `Error` gibt über einen eventuell aufgetretenen Fehler Auskunft. Die folgenden Fehlercodes sind definiert:

Fehlercodes:	Beschreibung
0	OK
1	Error: DRIVE NOT FOUND
	WARNING: PATH CONTAINS NO FUS
2	Error: PATH NOT FOUND
	WARNING: NO PROPERTY FILES FOUND
3	Error: PATH CONTAINS NO DIRS
	WARNING: TEXT REFERENCE FILE NOT FOUND

## Funktionalität „User“

Das Manipulieren von Anwendern und Anwenderrechten einer EC/CC200 Steuerung kann mit Hilfe der folgenden Funktion erfolgen:

- AckLoginRequest()
- AddUser()
- CancelLogin()
- GetLoginRequests()
- GetLoginStatus()
- GetSession()
- GetSessions()
- GetUserAttributes()
- GetUserConfig()
- GetUserNames()
- GetWriteAccessSessions()
- IsLocalStation()
- IsValidPassword()
- Login()
- Logout()
- LogoutAndModify()
- ModifyPassword()
- ModifyUser()
- RemoveUser() und
- TerminateUserSessions()

Die möglichen Ausprägungen des `LoginStatus` bei der Funktionalität "User" sind:

```

0 .. USER_LOGIN_GRANTED      .. Anmeldung erfolgreich abgeschlossen
1 .. USER_LOGIN_PENDING      .. Wartet auf Bestätigung
2 .. USER_LOGIN_DENIED       .. Anmeldung zurückgewiesen

```

## Bestätigung eines Login-Vorganges

<hostname>.Funcs.User.AckLoginRequest (↓Granted, ↓LoginId, ↓RequestId)	
IN:	Granted : BOOL LoginId : INT RequestId : INT
OUT:	Error : INT

Mit dieser Funktion wird ein anstehender Anmeldeversuch positiv oder negativ bestätigt.

## Benutzer hinzufügen

<code>&lt;hostname&gt;.Funcs.User.AddUser  (↓BeginDate, ↓Classes, ↓Deletable, ↓EndDate,  ↓FunctionMask, ↓Locale, ↓LocalLevel, ↓LoginId, ↓Name,  ↓Password, ↓RemoteLevel, ↓TextPattern, ↓Unit, ↑Error)</code>	
IN:	BeginDate : STRING Classes : BOOL Deletable : BOOL EndDate : STRING FunctionMask : INT Locale : STRING LocalLevel : INT LoginId : INT Name : STRING Password : STRING RemoteLevel : INT TextPattern : STRING Unit : INT
OUT:	Error : INT

Mit dieser Funktion kann ein Anwender zu einer EC/CC200 Steuerung hinzugefügt werden.

Für die OPC Items gilt:

- **BeginDate** Ab diesem Datum ist es dem Anwender möglich, sich bei einer EC/CC200 Steuerung anzumelden. Wenn dieses OPC Item als Typ VT\_BSTR abgefragt wird, so wird das Datum und die Uhrzeit im IEC Format „DT#yyyy-mm-dd-hh:mm:ss.uuuuuu“ geliefert.
- **Classes[]** Wird nicht verwendet.
- **Deletable** legt die Löschbarkeit fest
- **EndDate** Bis zu diesem Datum ist es dem Anwender möglich, sich bei einer EC/CC200 Steuerung anzumelden. Wenn dieses OPC Item als Typ VT\_BSTR abgefragt wird, so wird das Datum und die Uhrzeit im IEC Format „DT#yyyy-mm-dd-hh:mm:ss.uuuuuu“ geliefert.
- **FunctionMask** Wird nicht verwendet.
- **Locale** Die Länderbezeichnung.
- **LocalLevel** Die lokale Berechtigungsstufe (von 1 bis 16).
- **LoginId** Eindeutige Kennung des Benutzers (von Login)
- **Name** Der Anmeldename des Anwenders (login name); case-sensitive
- **Password** Das Passwort des Anwenders; case-sensitive
- **RemoteLevel** Die nicht-lokale Berechtigungsstufe (von 1 bis 16).

- **TextPatternFilter** für InfoLog. Eine Kombination aus den Werten

SYSTEM_MESSAGE	PCB_OPERATION, Error_INFO
Error_WARNING	Error_Error
Error_CRITICAL	Error_FATAL
ALARM_CLASS1	ALARM_CLASS2
ALARM_CLASS3	ALARM_CLASS4
ALARM_CLASS5	ALARM_CLASS6
ALARM_CLASS7	HMI_OPERATION
FILE_OPERATION	SPS_OPERATION
ONLINE_OPERATION	OPERATING_MODE_CHANGE
SW_CHANGE	USER_MESSAGE
USER_MANAGEMENT	UNKNOWN_TYPE

Dieser Filter kommt bei der Funktion `ReadEntries()` in der Funktionalität „InfoLog“ zum Einsatz. Siehe dazu „Einträge lesen“.

- **Unit** Die Einheit kann einen der folgenden Werte annehmen:

- 1 ... metrisch,
- 2 ... imperial.

Das OPC Item `Error` gibt über einen eventuell aufgetretenen Fehler Auskunft. Die folgenden Fehlercodes sind definiert:

<b>Fehlercodes:</b>	<b>Beschreibung</b>
-2	USER_OUT_OF_MEMORY kein Speicher für anzulegende Daten
-1	USER_INTERNAL_Error
0	USER_OK
1	USER_INVALID_NAME ungültiger Benutzername z.B. beim Anmelden
2	USER_INVALID_PASSWORD ungültiges Passwort beim Anmelden
3	USER_INVALID_LEVEL lokale < nichtlokale B. in addUser/modify
4	USER_MAX_LOGINS konfiguriertes Sitzungsmaximum erreicht
5	USER_MAX_WRITE_LOGINS konfiguriertes Max. mit Schreibzugriff erreicht
6	USER_NOT_LOGGED_IN z.B. logout(), modifyPwd() ohne Anmeldung
7	USER_USER_UNAUTHORIZED Benutzer nicht autorisiert
100	BeginDate- oder EndDate-Formatierung fehlerhaft

## Anmeldeversuch abbrechen

<hostname>.Funcs.User.CancelLogin (↓LoginId)	
IN:	LoginId : INT
OUT:	Error : INT

Mit dieser Funktion kann ein laufender Anmeldeversuch des Anwender bei einer EC/CC200 Steuerung abgebrochen werden.

## Liste anstehender Anmeldeversuche

<hostname>.Funcs.User.GetLoginRequests (↓LoginId, ↑Error, ↑LoginRequests[])	
IN:	LoginId : INT
OUT:	Error : INT LoginRequests[] Address : INT IsLocal : BOOL Level : INT Name : STRING Override : BOOL RequestId : INT WriteAccess : BOOL

Mit dieser Funktion kann eine Liste aller anstehenden Anmeldeversuche bei einer EC/CC200 Steuerung abgerufen werden.

Das OPC Item `Error` gibt über einen eventuell aufgetretenen Fehler Auskunft. Die folgenden Fehlercodes sind definiert:

Fehlercodes:	Beschreibung
-2	USER_OUT_OF_MEMORY kein Speicher für anzulegende Daten
-1	USER_INTERNAL_Error
0	USER_OK
1	USER_INVALID_NAME ungültiger Benutzername z.B. beim Anmelden
2	USER_INVALID_PASSWORD ungültiges Passwort beim Anmelden
3	USER_INVALID_LEVEL lokale < nichtlokale B. in addUser/modify
4	USER_MAX_LOGINS konfiguriertes Sitzungsmaximum erreicht
5	USER_MAX_WRITE_LOGINS konfiguriertes Max. mit Schreibzugriff erreicht
6	USER_NOT_LOGGED_IN z.B. logout(), modifyPwd() ohne Anmeldung
7	USER_USER_UNAUTHORIZED Benutzer nicht autorisiert



## Aktueller Status des Anmeldevorgangs

<hostname>.Funcs.User.GetLoginStatus (↓LoginId, ↑Status)	
IN:	LoginId : INT
OUT:	Status : INT
	Error : INT

Mit dieser Funktion kann der aktuelle LoginStatus einer EC/CC200 Steuerung ermittelt werden.

## Information über aktuelle Sitzung

<hostname>.Funcs.User.GetSession (↓LoginId, ↑Address, ↑Error, ↑IsLocal, ↑Level, ↑Name, ↑Overridden, ↑WriteAccess)	
IN:	LoginId : INT
OUT:	Address : INT
	Error : INT
	IsLocal : BOOL
	Level : INT
	Name : STRING
	Overridden : BOOL
	WriteAccess : BOOL

Mit dieser Funktion kann die Information der momentanen Sitzung einer EC/CC200 Steuerung abrufen werden.

Das OPC Item `Error` gibt über einen eventuell aufgetretenen Fehler Auskunft. Die folgenden Fehlercodes sind definiert:

Fehlercodes:	Beschreibung
-2	USER_OUT_OF_MEMORY kein Speicher für anzulegende Daten
-1	USER_INTERNAL_Error
0	USER_OK
1	USER_INVALID_NAME ungültiger Benutzername z.B. beim Anmelden
2	USER_INVALID_PASSWORD ungültiges Passwort beim Anmelden
3	USER_INVALID_LEVEL lokale < nichtlokale B. in addUser/modify
4	USER_MAX_LOGINS konfiguriertes Sitzungsmaximum erreicht
5	USER_MAX_WRITE_LOGINS konfiguriertes Max. mit Schreibzugriff erreicht
6	USER_NOT_LOGGED_IN z.B. logout(), modifyPwd() ohne Anmeldung
7	USER_USER_UNAUTHORIZED Benutzer nicht autorisiert

## Information über bestehende Sitzungen

<hostname>.Funcs.User.GetSessions (↑Error, ↑Sessions[])	
OUT:	Error : INT Sessions[] Address : INT IsLocal : BOOL Level : INT Name : STRING Overriden : BOOL WriteAccess : BOOL

Mit dieser Funktion können alle momentan laufenden Sitzungen einer EC/CC200 Steuerung abgerufen werden.

Für Sessions[] gilt:

Address ... IP-Adresse der Station  
 IsLocal ... Lokal (1) oder Remote (0)  
 Level ... Berechtigungsstufe  
 Name ... Benutzername  
 Overriden ... Entzug des Schreibrechts  
 WriteAccess ... Schreibzugriff

Das OPC Item `Error` gibt über einen eventuell aufgetretenen Fehler Auskunft. Die folgenden Fehlercodes sind definiert:

Fehlercodes:	Beschreibung
-2	USER_OUT_OF_MEMORY kein Speicher für anzulegende Daten
-1	USER_INTERNAL_Error
0	USER_OK
1	USER_INVALID_NAME ungültiger Benutzername z.B. beim Anmelden
2	USER_INVALID_PASSWORD ungültiges Passwort beim Anmelden
3	USER_INVALID_LEVEL lokale < nichtlokale B. in addUser/modify
4	USER_MAX_LOGINS konfiguriertes Sitzungsmaximum erreicht
5	USER_MAX_WRITE_LOGINS konfiguriertes Max. mit Schreibzugriff erreicht
6	USER_NOT_LOGGED_IN z.B. logout(), modifyPwd() ohne Anmeldung
7	USER_USER_UNAUTHORIZED Benutzer nicht autorisiert

## Attribute eines Benutzers

<hostname>.Funcs.User.GetUserAttributes (↓Name, ↑BeginDate, ↑Classes, ↑Deletable, ↑EndDate, ↑Error, ↑FunctionMask, ↑Locale, ↑LocalLevel, ↑Name, ↑Password, ↑RemoteLevel, ↑TextPattern, ↑Unit)	
IN:	Name : STRING
OUT:	BeginDate : STRING Classes : BOOL Deletable : BOOL EndDate : STRING Error : INT FunctionMask : INT Locale : STRING LocalLevel : INT Name : STRING Password : ARRAY RemoteLevel : INT TextPattern : STRING Unit : INT

Mit dieser Funktion können Attribute eines registrierten Benutzers einer EC/CC200 Steuerung abrufen werden.

Das OPC Item `Error` gibt über einen eventuell aufgetretenen Fehler Auskunft. Die folgenden Fehlercodes sind definiert:

Fehlercodes:	Beschreibung
-2	USER_OUT_OF_MEMORY kein Speicher für anzulegende Daten
-1	USER_INTERNAL_Error
0	USER_OK
1	USER_INVALID_NAME ungültiger Benutzername z.B. beim Anmelden
2	USER_INVALID_PASSWORD ungültiges Passwort beim Anmelden
3	USER_INVALID_LEVEL lokale < nichtlokale B. in addUser/modify
4	USER_MAX_LOGINS konfiguriertes Sitzungsmaximum erreicht
5	USER_MAX_WRITE_LOGINS konfiguriertes Max. mit Schreibzugriff erreicht
6	USER_NOT_LOGGED_IN z.B. logout(), modifyPwd() ohne Anmeldung
7	USER_USER_UNAUTHORIZED Benutzer nicht autorisiert

## Benutzer-Konfiguration

<pre>&lt;hostname&gt;.Funcs.User.GetUserConfig (↑Error, ↑MaxLocalSessions, ↑MaxLocalWriteSessions, ↑MaxRemoteSessions, ↑MaxRemoteWriteSessions, ↑MaxTotalSessions)</pre>	
OUT:	<pre>Error : INT MaxLocalSessions : INT MaxLocalWriteSessions : INT MaxRemoteSessions : INT MaxRemoteWriteSessions : INT MaxTotalSessions : INT</pre>

Mit dieser Funktion kann aus einer EC/CC200 Steuerung die aktuelle Benutzerkonfiguration abgerufen werden.

Das OPC Item `Error` gibt über einen eventuell aufgetretenen Fehler Auskunft. Die folgenden Fehlercodes sind definiert:

Fehlercodes:	Beschreibung
-2	USER_OUT_OF_MEMORY kein Speicher für anzulegende Daten
-1	USER_INTERNAL_Error
0	USER_OK
1	USER_INVALID_NAME ungültiger Benutzername z.B. beim Anmelden
2	USER_INVALID_PASSWORD ungültiges Passwort beim Anmelden
3	USER_INVALID_LEVEL lokale < nichtlokale B. in addUser/modify
4	USER_MAX_LOGINS konfiguriertes Sitzungsmaximum erreicht
5	USER_MAX_WRITE_LOGINS konfiguriertes Max. mit Schreibzugriff erreicht
6	USER_NOT_LOGGED_IN z.B. logout(), modifyPwd() ohne Anmeldung
7	USER_USER_UNAUTHORIZED Benutzer nicht autorisiert

## Registrierte Benutzernamen ermitteln

<hostname>.Funcs.User.GetUserNames (↑Error, ↑Users[])	
OUT:	Error : INT Users[] Name : STRING

Mit dieser Funktion können alle Namen der registrierten Benutzer einer EC/CC200 Steuerung abrufen werden.  
Users[] ist dabei ein Array mit Usernamen.

Das OPC Item `Error` gibt über einen eventuell aufgetretenen Fehler Auskunft.  
Die folgenden Fehlercodes sind definiert:

Fehlercodes:	Beschreibung
-2	USER_OUT_OF_MEMORY kein Speicher für anzulegende Daten
-1	USER_INTERNAL_Error
0	USER_OK
1	USER_INVALID_NAME ungültiger Benutzername z.B. beim Anmelden
2	USER_INVALID_PASSWORD ungültiges Passwort beim Anmelden
3	USER_INVALID_LEVEL lokale < nichtlokale B. in addUser/modify
4	USER_MAX_LOGINS konfiguriertes Sitzungsmaximum erreicht
5	USER_MAX_WRITE_LOGINS konfiguriertes Max. mit Schreibzugriff erreicht
6	USER_NOT_LOGGED_IN z.B. logout(), modifyPwd() ohne Anmeldung
7	USER_USER_UNAUTHORIZED Benutzer nicht autorisiert

## Sitzungen mit Schreibzugriff ermitteln

<hostname>.Funcs.User.GetWriteAccessSessions (↑Error, ↑LocalSessionCount, ↑RemoteSessionCount)	
OUT:	Error : INT
	LocalSessionCount : INT
	RemoteSessionCount : INT

Mit dieser Funktion können alle lokalen und nicht-lokalen Sitzungen mit Schreibzugriff einer EC/CC200 Steuerung ermittelt werden.

Das OPC Item `Error` gibt über einen eventuell aufgetretenen Fehler Auskunft. Die folgenden Fehlercodes sind definiert:

Fehlercodes:	Beschreibung
-2	USER_OUT_OF_MEMORY kein Speicher für anzulegende Daten
-1	USER_INTERNAL_Error
0	USER_OK
1	USER_INVALID_NAME ungültiger Benutzername z.B. beim Anmelden
2	USER_INVALID_PASSWORD ungültiges Passwort beim Anmelden
3	USER_INVALID_LEVEL lokale < nichtlokale B. in addUser/modify
4	USER_MAX_LOGINS konfiguriertes Sitzungsmaximum erreicht
5	USER_MAX_WRITE_LOGINS konfiguriertes Max. mit Schreibzugriff erreicht
6	USER_NOT_LOGGED_IN z.B. logout(), modifyPwd() ohne Anmeldung
7	USER_USER_UNAUTHORIZED Benutzer nicht autorisiert

## Remote/Lokal-Abfrage

<hostname>.Funcs.User.IsLocalStation (↑IsLocal)	
OUT:	IsLocal : BOOL Error : INT

Mit dieser Funktion kann bei einer EC/CC200 Steuerung abgesehen werden, ob die Station als "lokal" definiert ist.

## Passwort prüfen

<hostname>.Funcs.User.IsValidPassword (↓Name, ↓Password, ↑IsValid)	
IN:	Name : STRING Password : STRING
OUT:	IsValid : BOOL Error : INT

Mit dieser Funktion kann die Gültigkeit des Passwortes einer EC/CC200 Steuerung geprüft werden.

## Benutzer anmelden

<hostname>.Funcs.User.LogIn (↓LocalLevel, ↓Name, ↓Override, ↓Password, ↓WriteAccess, ↑Error, ↑LoginId, ↑Status)	
IN:	LocalLevel : INT Name : STRING Override : BOOL Password : STRING WriteAccess : BOOL
OUT:	Error : INT LoginId : INT Status : INT

Mit dieser Funktion kann ein Benutzer auf einer EC/CC200 Steuerung angemeldet werden.

Das OPC Item `Error` gibt über einen eventuell aufgetretenen Fehler Auskunft. Die folgenden Fehlercodes sind definiert:

Fehlercodes:	Beschreibung
-2	USER_OUT_OF_MEMORY kein Speicher für anzulegende Daten
-1	USER_INTERNAL_Error
0	USER_OK
1	USER_INVALID_NAME ungültiger Benutzername z.B. beim Anmelden
2	USER_INVALID_PASSWORD ungültiges Passwort beim Anmelden
3	USER_INVALID_LEVEL lokale < nichtlokale B. in addUser/modify
4	USER_MAX_LOGINS konfiguriertes Sitzungsmaximum erreicht
5	USER_MAX_WRITE_LOGINS konfiguriertes Max. mit Schreibzugriff erreicht
6	USER_NOT_LOGGED_IN z.B. logout(), modifyPwd() ohne Anmeldung
7	USER_USER_UNAUTHORIZED Benutzer nicht autorisiert

## Benutzer abmelden

<hostname>.Funcs.User.LogOut (↓LoginId, ↑Error)	
IN:	LoginId : INT
OUT:	Error : INT

Mit dieser Funktion kann ein Benutzer von einer EC/CC200 Steuerung abgemeldet werden.

Das OPC Item `Error` gibt über einen eventuell aufgetretenen Fehler Auskunft. Die folgenden Fehlercodes sind definiert:

Fehlercodes:	Beschreibung
-2	USER_OUT_OF_MEMORY kein Speicher für anzulegende Daten
-1	USER_INTERNAL_Error
0	USER_OK
1	USER_INVALID_NAME ungültiger Benutzername z.B. beim Anmelden
2	USER_INVALID_PASSWORD ungültiges Passwort beim Anmelden
3	USER_INVALID_LEVEL lokale < nichtlokale B. in addUser/modify
4	USER_MAX_LOGINS konfiguriertes Sitzungsmaximum erreicht
5	USER_MAX_WRITE_LOGINS konfiguriertes Max. mit Schreibzugriff erreicht
6	USER_NOT_LOGGED_IN z.B. logout(), modifyPwd() ohne Anmeldung
7	USER_USER_UNAUTHORIZED Benutzer nicht autorisiert



## Benutzer abmelden und Profil ändern

<hostname>.Funcs.User.LogOutAndModify (↓BeginDate, ↓Classes, ↓EndDate, ↓FunctionMask, ↓Locale, ↓LoginId, ↓TextPattern, ↓Unit, ↑Error)	
IN:	BeginDate : STRING Classes : BOOL EndDate : STRING FunctionMask : INT Locale : STRING LoginId : INT TextPattern : STRING Unit : INT
OUT:	Error : INT

Mit dieser Funktion kann das Profil eines Benutzers auf einer EC/CC200 geändert werden, gleichzeitig wird der Benutzer abgemeldet.

Das OPC Item `Error` gibt über einen eventuell aufgetretenen Fehler Auskunft. Die folgenden Fehlercodes sind definiert:

Fehlercodes:	Beschreibung
-2	USER_OUT_OF_MEMORY kein Speicher für anzulegende Daten
-1	USER_INTERNAL_Error
0	USER_OK
1	USER_INVALID_NAME ungültiger Benutzername z.B. beim Anmelden
2	USER_INVALID_PASSWORD ungültiges Passwort beim Anmelden
3	USER_INVALID_LEVEL lokale < nichtlokale B. in addUser/modify
4	USER_MAX_LOGINS konfiguriertes Sitzungsmaximum erreicht
5	USER_MAX_WRITE_LOGINS konfiguriertes Max. mit Schreibzugriff erreicht
6	USER_NOT_LOGGED_IN z.B. logout(), modifyPwd() ohne Anmeldung
7	USER_USER_UNAUTHORIZED Benutzer nicht autorisiert
100	BeginDate- oder EndDate-Formatierung fehlerhaft

## Passwort ändern

<hostname>.Funcs.User.ModifyPassword (↓LoginId, ↓ Password, ↑Error)	
IN:	LoginId : INT Password : STRING
OUT:	Error : INT

Mit dieser Funktion kann das Passwort einer EC/CC200 Steuerung geändert werden.

Das OPC Item `Error` gibt über einen eventuell aufgetretenen Fehler Auskunft. Die folgenden Fehlercodes sind definiert:

Fehlercodes:	Beschreibung
-2	USER_OUT_OF_MEMORY kein Speicher für anzulegende Daten
-1	USER_INTERNAL_Error
0	USER_OK
1	USER_INVALID_NAME ungültiger Benutzername z.B. beim Anmelden
2	USER_INVALID_PASSWORD ungültiges Passwort beim Anmelden
3	USER_INVALID_LEVEL lokale < nichtlokale B. in addUser/modify
4	USER_MAX_LOGINS konfiguriertes Sitzungsmaximum erreicht
5	USER_MAX_WRITE_LOGINS konfiguriertes Max. mit Schreibzugriff erreicht
6	USER_NOT_LOGGED_IN z.B. logout(), modifyPwd() ohne Anmeldung
7	USER_USER_UNAUTHORIZED Benutzer nicht autorisiert

## Anwender ändern

<code>&lt;hostname&gt;.Funcs.User.ModifyUser  (↓BeginDate, ↓Classes, ↓Deletable, ↓EndDate,  ↓FunctionMask, ↓Locale, ↓LocalLevel, ↓LoginId, ↓Name,  ↓Password, ↓RemoteLevel, ↓TextPattern, ↓Unit, ↑Error)</code>	
IN:	BeginDate : STRING Classes : BOOL Deletable : BOOL EndDate : STRING FunctionMask : INT Locale : STRING LocalLevel : INT LoginId : INT Name : STRING Password : STRING RemoteLevel : INT TextPattern : STRING Unit : INT
OUT:	Error : INT

Mit dieser Funktion können die Parameter eines Benutzers auf einer EC/CC200 Steuerung verändert werden.

Das OPC Item `Error` gibt über einen eventuell aufgetretenen Fehler Auskunft. Die folgenden Fehlercodes sind definiert:

Fehlercodes:	Beschreibung
-2	USER_OUT_OF_MEMORY kein Speicher für anzulegende Daten
-1	USER_INTERNAL_Error
0	USER_OK
1	USER_INVALID_NAME ungültiger Benutzername z.B. beim Anmelden
2	USER_INVALID_PASSWORD ungültiges Passwort beim Anmelden
3	USER_INVALID_LEVEL lokale < nichtlokale B. in addUser/modify
4	USER_MAX_LOGINS konfiguriertes Sitzungsmaximum erreicht
5	USER_MAX_WRITE_LOGINS konfiguriertes Max. mit Schreibzugriff erreicht
6	USER_NOT_LOGGED_IN z.B. logout(), modifyPwd() ohne Anmeldung
7	USER_USER_UNAUTHORIZED Benutzer nicht autorisiert
100	BeginDate- oder EndDate-Formatierung fehlerhaft

## Benutzer löschen

<hostname>.Funcs.User.RemoveUser (↓LoginId, ↓Name, ↑Error)	
IN:	LoginId : INT Name : STRING
OUT:	Error : INT

Mit dieser Funktion kann ein Benutzer von einer EC/CC200 Steuerung gelöscht werden.

Das OPC Item `Error` gibt über einen eventuell aufgetretenen Fehler Auskunft. Die folgenden Fehlercodes sind definiert:

Fehlercodes:	Beschreibung
-2	USER_OUT_OF_MEMORY kein Speicher für anzulegende Daten
-1	USER_INTERNAL_Error
0	USER_OK
1	USER_INVALID_NAME ungültiger Benutzername z.B. beim Anmelden
2	USER_INVALID_PASSWORD ungültiges Passwort beim Anmelden
3	USER_INVALID_LEVEL lokale < nichtlokale B. in addUser/modify
4	USER_MAX_LOGINS konfiguriertes Sitzungsmaximum erreicht
5	USER_MAX_WRITE_LOGINS konfiguriertes Max. mit Schreibzugriff erreicht
6	USER_NOT_LOGGED_IN z.B. logout(), modifyPwd() ohne Anmeldung
7	USER_USER_UNAUTHORIZED Benutzer nicht autorisiert

## Sitzungen beenden

<hostname>.Funcs.User.TerminateUserSessions (↓LoginId, ↓Name, ↑Error)	
IN:	LoginId : INT Name : STRING
OUT:	Error : INT

Mit dieser Funktion können alle Sitzungen des angegebenen Benutzers einer EC/CC200 beendet werden.

Das OPC Item `Error` gibt über einen eventuell aufgetretenen Fehler Auskunft. Die folgenden Fehlercodes sind definiert:

Fehlercodes:	Beschreibung
-2	USER_OUT_OF_MEMORY kein Speicher für anzulegende Daten
-1	USER_INTERNAL_Error
0	USER_OK
1	USER_INVALID_NAME ungültiger Benutzername z.B. beim Anmelden
2	USER_INVALID_PASSWORD ungültiges Passwort beim Anmelden
3	USER_INVALID_LEVEL lokale < nichtlokale B. in addUser/modify
4	USER_MAX_LOGINS konfiguriertes Sitzungsmaximum erreicht
5	USER_MAX_WRITE_LOGINS konfiguriertes Max. mit Schreibzugriff erreicht
6	USER_NOT_LOGGED_IN z.B. logout(), modifyPwd() ohne Anmeldung
7	USER_USER_UNAUTHORIZED Benutzer nicht autorisiert

## 4 Format der OpcSvc-INI-Datei

Die Initialisierungsdatei des OPC-Servers (bei Standard-Konfiguration) befindet sich im gleichen Verzeichnis wie die Datei `OpcSvc.ini`. Sie hat folgenden Aufbau:

```
OpcSvc.ini      = Host [ MultiLanguage ].
Host           = "[DefaultHosts]" { HostEntry }.
HostEntry      = [ ";" ] Synonym "=" HostIdentifier.
Synonym        = AlphaNumString.
HostIdentifier = HostName | IPAddress.
HostName       = AlphaNumString.
IPAddress      = Byte "." Byte "." Byte "." Byte.
Byte           = Digit [Digit] [Digit].

[MultiLanguageSupport]
MultiLanguage  = "DefaultLanguageId" "=" LangId.
LangId        = AlphaNumString.

Digit          = "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" |
"9".
AlphaNumString = { "A" | "B" | .... "Z" | "a" | .. "z" | ... | Digit }.
```

`AlphaNumString` sind alle gültigen Zeichen ohne Whitespace, Umlaute, Sonderzeichen.

Das Semikolon ist ein Kommentar - die gesamte restliche Zeile wird überlesen. Für die Steuerung werden sowohl Hostnamen als auch IP-Adressen verarbeitet. Die `LangId` ist die Kennung der Sprache bei den Properties-Files.

**Beispiel für eine INI-Datei mit mehreren Einträgen:**

```
[DefaultHosts]
    Local      = localhost
;    Local0    = 127.0.0.1

[MultiLanguageSupport]
    DefaultLanguageId = DE
```

### **Hinweis**

*Die Einträge in der INI-Datei (mit Ausnahme des Hostnamens) sind "case-sensitive", d.h. es wird zwischen Groß- und Kleinschreibung unterschieden..*

## **5 Keep Alive**

Der OPC-Client muss zum Aufrechterhalten der Verbindung zum OPC-Server zyklisch Keep-Alive-Messages absetzen. Die Zeitüberwachung wird dabei von der EC/CC200 Steuerung übernommen. Durch das zyklische Aufrufen der Funktion `GetSystemStatus()` wird zum einen die Verbindung zur Steuerung gehalten und zum anderen für den OPC-Client notwendige Systeminformation der Steuerung abgerufen (globaler Änderungszähler, Statusinformationen, Benutzerrechte und Berechtigungsstufen, etc. ). Ohne diesem zyklischen Aufruf wird die Netzverbindung nach dem von der Steuerung festgelegten Zeitintervall getrennt.

## 6 Fehlercodes

- 0L      Error\_SUCCESS**  
*Message Text:*  
The operation completed successfully.
- 1L      Error\_INVALID\_FUNCTION**  
*Message Text:*  
Incorrect function.
- 2L      Error\_FILE\_NOT\_FOUND**  
*Message Text:*  
The system cannot find the file specified.
- 3L      Error\_PATH\_NOT\_FOUND**  
*Message Text:*  
The system cannot find the path specified.
- 4L      Error\_TOO\_MANY\_OPEN\_FILES**  
*Message Text:*  
The system cannot open the file.
- 5L      Error\_ACCESS\_DENIED**  
*Message Text:*  
Access is denied.
- 6L      Error\_INVALID\_HANDLE**  
*Message Text:*  
The handle is invalid.
- 7L      Error\_ARENA\_TRASHED**  
*Message Text:*  
The storage control blocks were destroyed.
- 8L      Error\_NOT\_ENOUGH\_MEMORY**  
*Message Text:*  
Not enough storage is available to process this command.
- 9L      Error\_INVALID\_BLOCK**  
*Message Text:*  
The storage control block address is invalid.



**10L      Error\_BAD\_ENVIRONMENT***Message Text:*

The environment is incorrect.

**11L      Error\_BAD\_FORMAT***Message Text:*

An attempt was made to load a program with an incorrect format.

**12L      Error\_INVALID\_ACCESS***Message Text:*

The access code is invalid.

**13L      Error\_INVALID\_DATA***Message Text:*

The data is invalid.

**14L      Error\_OUTOFMEMORY***Message Text:*

Not enough storage is available to complete this operation.

**15L      Error\_INVALID\_DRIVE***Message Text:*

The system cannot find the drive specified.

**16L      Error\_CURRENT\_DIRECTORY***Message Text:*

The directory cannot be removed.

**17L      Error\_NOT\_SAME\_DEVICE***Message Text:*

The system cannot move the file to a different disk drive.

**18L      Error\_NO\_MORE\_FILES***Message Text:*

There are no more files.

**19L      Error\_WRITE\_PROTECT***Message Text:*

The media is write protected.

**20L      Error\_BAD\_UNIT***Message Text:*

The system cannot find the device specified.

- 21L      Error\_NOT\_READY**  
*Message Text:*  
The device is not ready.
- 22L      Error\_BAD\_COMMAND**  
*Message Text:*  
The device does not recognize the command.
- 23L      Error\_CRC**  
*Message Text:*  
Data error (cyclic redundancy check).
- 24L      Error\_BAD\_LENGTH**  
*Message Text:*  
The program issued a command but the command length is incorrect.
- 25L      Error\_SEEK**  
*Message Text:*  
The drive cannot locate a specific area or track on the disk.
- 26L      Error\_NOT\_DOS\_DISK**  
*Message Text:*  
The specified disk or diskette cannot be accessed.
- 27L      Error\_SECTOR\_NOT\_FOUND**  
*Message Text:*  
The drive cannot find the sector requested.
- 28L      Error\_OUT\_OF\_PAPER**  
*Message Text:*  
The printer is out of paper.
- 29L      Error\_WRITE\_FAULT**  
*Message Text:*  
The system cannot write to the specified device.
- 30L      Error\_READ\_FAULT**  
*Message Text:*  
The system cannot read from the specified device.
- 31L      Error\_GEN\_FAILURE**  
*Message Text:*

A device attached to the system is not functioning.

**32L      Error\_SHARING\_VIOLATION**

*Message Text:*

The process cannot access the file because it is being used by another process.

**33L      Error\_LOCK\_VIOLATION**

*Message Text:*

The process cannot access the file because another process has locked a portion of the file.

**34L      Error\_WRONG\_DISK**

*Message Text:*

The wrong diskette is in the drive. Insert %2 (Volume Serial Number: %3) into drive %1.

**36L      Error\_SHARING\_BUFFER\_EXCEEDED**

*Message Text:*

Too many files opened for sharing.

**38L      Error\_HANDLE\_EOF**

*Message Text:*

Reached the end of the file.

**39L      Error\_HANDLE\_DISK\_FULL**

*Message Text:*

The disk is full.

**50L      Error\_NOT\_SUPPORTED**

*Message Text:*

The network request is not supported.

**51L      Error\_REM\_NOT\_LIST**

*Message Text:*

The remote computer is not available.

**52L      Error\_DUP\_NAME**

*Message Text:*

A duplicate name exists on the network.

**53L      Error\_BAD\_NETPATH**

*Message Text:*

The network path was not found.

- 54L      Error\_NETWORK\_BUSY**  
*Message Text:*  
The network is busy.
- 55L      Error\_DEV\_NOT\_EXIST**  
*Message Text:*  
The specified network resource or device is no longer available.
- 56L      Error\_TOO\_MANY\_CMDS**  
*Message Text:*  
The network BIOS command limit has been reached.
- 57L      Error\_ADAP\_HDW\_ERR**  
*Message Text:*  
A network adapter hardware error occurred.
- 58L      Error\_BAD\_NET\_RESP**  
*Message Text:*  
The specified server cannot perform the requested operation.
- 59L      Error\_UNEXP\_NET\_ERR**  
*Message Text:*  
An unexpected network error occurred.
- 60L      Error\_BAD\_REM\_ADAP**  
*Message Text:*  
The remote adapter is not compatible.
- 61L      Error\_PRINTQ\_FULL**  
*Message Text:*  
printer queue is full.
- 62L      Error\_NO\_SPOOL\_SPACE**  
*Message Text:*  
Space to store the file waiting to be printed is not available on the server.

**63L      Error\_PRINT\_CANCELLED***Message Text:*

Your file waiting to be printed was deleted.

**64L      Error\_NETNAME\_DELETED***Message Text:*

The specified network name is no longer available.

**65L      Error\_NETWORK\_ACCESS\_DENIED***Message Text:*

Network access is denied.

**66L      Error\_BAD\_DEV\_TYPE***Message Text:*

The network resource type is not correct.

**67L      Error\_BAD\_NET\_NAME***Message Text:*

The network name cannot be found.

**68L      Error\_TOO\_MANY\_NAMES***Message Text:*

The name limit for the local computer network adapter card was exceeded.

**69L      Error\_TOO\_MANY\_SESS***Message Text:*

The network BIOS session limit was exceeded.

**70L      Error\_SHARING\_PAUSED***Message Text:*

The remote server has been paused or is in the process of being started.

**71L      Error\_REQ\_NOT\_ACCEP***Message Text:*

No more connections can be made to this remote computer at this time because there are already as many connections as the computer can accept.

**72L      Error\_REDIR\_PAUSED***Message Text:*

The specified printer or disk device has been paused.

- 80L      Error\_FILE\_EXISTS**  
*Message Text:*  
The file exists.
- 82L      Error\_CANNOT\_MAKE**  
*Message Text:*  
The directory or file cannot be created.
- 83L      Error\_FAIL\_I24**  
*Message Text:*  
Fail on INT 24.
- 84L      Error\_OUT\_OF\_STRUCTURES**  
*Message Text:*  
Storage to process this request is not available.
- 85L      Error\_ALREADY\_ASSIGNED**  
*Message Text:*  
The local device name is already in use.
- 86L      Error\_INVALID\_PASSWORD**  
*Message Text:*  
The specified network password is not correct.
- 87L      Error\_INVALID\_PARAMETER**  
*Message Text:*  
The parameter is incorrect.
- 88L      Error\_NET\_WRITE\_FAULT**  
*Message Text:*  
fault occurred on the network.
- 89L      Error\_NO\_PROC\_SLOTS**  
*Message Text:*  
The system cannot start another process at this time.
- 100L     Error\_TOO\_MANY\_SEMAPHORES**  
*Message Text:*  
Cannot create another system semaphore.

- 101L      Error\_EXCL\_SEM\_ALREADY\_OWNED**  
*Message Text:*  
The exclusive semaphore is owned by another process.
- 102L      Error\_SEM\_IS\_SET**  
*Message Text:*  
The semaphore is set and cannot be closed.
- 103L      Error\_TOO\_MANY\_SEM\_REQUESTS**  
*Message Text:*  
The semaphore cannot be set again.
- 104L      Error\_INVALID\_AT\_INTERRUPT\_TIME**  
*Message Text:*  
Cannot request exclusive semaphores at interrupt time.
- 105L      Error\_SEM\_OWNER\_DIED**  
*Message Text:*  
The previous ownership of this semaphore has ended.
- 106L      Error\_SEM\_USER\_LIMIT**  
*Message Text:*  
Insert the diskette for drive %1.
- 107L      Error\_DISK\_CHANGE**  
*Message Text:*  
The program stopped because an alternate diskette was not inserted.
- 108L      Error\_DRIVE\_LOCKED**  
*Message Text:*  
The disk is in use or locked by another process.
- 109L      Error\_BROKEN\_PIPE**  
*Message Text:*  
The pipe has been ended.
- 110L      Error\_OPEN\_FAILED**  
*Message Text:*  
The system cannot open the device or file specified.

- 111L      Error\_BUFFER\_OVERFLOW**  
*Message Text:*  
The file name is too long.
- 112L      Error\_DISK\_FULL**  
*Message Text:*  
There is not enough space on the disk.
- 113L      Error\_NO\_MORE\_SEARCH\_HANDLES**  
*Message Text:*  
No more internal file identifiers available.
- 114L      Error\_INVALID\_TARGET\_HANDLE**  
*Message Text:*  
The target internal file identifier is incorrect.
- 117L      Error\_INVALID\_CATEGORY**  
*Message Text:*  
The IOCTL call made by the application program is not correct.
- 118L      Error\_INVALID\_VERIFY\_SWITCH**  
*Message Text:*  
The verify-on-write switch parameter value is not correct.
- 119L      Error\_BAD\_DRIVER\_LEVEL**  
*Message Text:*  
system does not support the command requested.
- 120L      Error\_CALL\_NOT\_IMPLEMENTED**  
*Message Text:*  
This function is not supported on this system.
- 121L      Error\_SEM\_TIMEOUT**  
*Message Text:*  
The semaphore timeout period has expired.
- 122L      Error\_INSUFFICIENT\_BUFFER**  
*Message Text:*  
The data area passed to a system call is too small.
- 123L      Error\_INVALID\_NAME**  
*Message Text:*  
The filename, directory name, or volume label syntax is incorrect.



**124L      Error\_INVALID\_LEVEL***Message Text:*

The system call level is not correct.

**125L      Error\_NO\_VOLUME\_LABEL***Message Text:*

The disk has no volume label.

**126L      Error\_MOD\_NOT\_FOUND***Message Text:*

The specified module could not be found.

**127L      Error\_PROC\_NOT\_FOUND***Message Text:*

The specified procedure could not be found.

**128L      Error\_WAIT\_NO\_CHILDREN***Message Text:*

There are no child processes to wait for.

**129L      Error\_CHILD\_NOT\_COMPLETE***Message Text:*

The %1 application cannot be run in Win32 mode.

**130L      Error\_DIRECT\_ACCESS\_HANDLE***Message Text:*

Attempt to use a file handle to an open disk partition for an operation other than raw disk I/O.

**131L      Error\_NEGATIVE\_SEEK***Message Text:*

An attempt was made to move the file pointer before the beginning of the file.

**132L      Error\_SEEK\_ON\_DEVICE***Message Text:*

The file pointer cannot be set on the specified device or file.

**133L      Error\_IS\_JOIN\_TARGET***Message Text:*

A JOIN or SUBST command cannot be used for a drive that contains previously joined drives.

**134L      Error\_IS\_JOINED***Message Text:*

An attempt was made to use a JOIN or SUBST command on a drive that has already been joined.

**135L      Error\_IS\_SUBSTED***Message Text:*

An attempt was made to use a JOIN or SUBST command on a drive that has already been substituted.

**136L      Error\_NOT\_JOINED***Message Text:*

The system tried to delete the JOIN of a drive that is not joined.

**137L      Error\_NOT\_SUBSTED***Message Text:*

The system tried to delete the substitution of a drive that is not substituted.

**138L      Error\_JOIN\_TO\_JOIN***Message Text:*

system tried to join a drive to a directory on a joined drive.

**139L      Error\_SUBST\_TO\_SUBST***Message Text:*

The system tried to substitute a drive to a directory on a substituted drive.

**140L      Error\_JOIN\_TO\_SUBST***Message Text:*

system tried to join a drive to a directory on a substituted drive.

**141L      Error\_SUBST\_TO\_JOIN***Message Text:*

The system tried to SUBST a drive to a directory on a joined drive.

**142L      Error\_BUSY\_DRIVE***Message Text:*

The system cannot perform a JOIN or SUBST at this time.

**143L      Error\_SAME\_DRIVE***Message Text:*

The system cannot join or substitute a drive to or for a directory on the same drive.

**144L      Error\_DIR\_NOT\_ROOT***Message Text:*

The directory is not a subdirectory of the root directory.

**145L      Error\_DIR\_NOT\_EMPTY***Message Text:*

The directory is not empty.

**146L      Error\_IS\_SUBST\_PATH***Message Text:*

The path specified is being used in a substitute.

**147L      Error\_IS\_JOIN\_PATH***Message Text:*

Not enough resources are available to process this command.

**148L      Error\_PATH\_BUSY***Message Text:*

The path specified cannot be used at this time.

**149L      Error\_IS\_SUBST\_TARGET***Message Text:*

An attempt was made to join or substitute a drive for which a directory on the drive is the target of a previous substitute.

**150L      Error\_SYSTEM\_TRACE***Message Text:*

System trace information was not specified in your CONFIG.SYS file, or tracing is disallowed.

**151L      Error\_INVALID\_EVENT\_COUNT***Message Text:*

The number of specified semaphore events for DosMuxSemWait is not correct.

**152L      Error\_TOO\_MANY\_MUXWAITERS***Message Text:*

DosMuxSemWait did not execute; too many semaphores are already set.

**153L      Error\_INVALID\_LIST\_FORMAT***Message Text:*

The DosMuxSemWait list is not correct.

**154L      Error\_LABEL\_TOO\_LONG***Message Text:*

The volume label you entered exceeds the label character limit of the target file system.

**155L      Error\_TOO\_MANY\_TCBS***Message Text:*

Cannot create another thread.

**156L      Error\_SIGNAL\_REFUSED***Message Text:*

The recipient process has refused the signal.

**157L      Error\_DISCARDED***Message Text:*

The segment is already discarded and cannot be locked.

**158L      Error\_NOT\_LOCKED***Message Text:*

segment is already unlocked.

**159L      Error\_BAD\_THREADID\_ADDR***Message Text:*

The address for the thread ID is not correct.

**160L      Error\_BAD\_ARGUMENTS***Message Text:*

The argument string passed to DosExecPgm is not correct.

**161L      Error\_BAD\_PATHNAME***Message Text:*

The specified path is invalid.

**162L      Error\_SIGNAL\_PENDING***Message Text:*

A signal is already pending.

**164L      Error\_MAX\_THRDS\_REACHED***Message Text:*

No more threads can be created in the system.

**167L      Error\_LOCK\_FAILED***Message Text:*

Unable to lock a region of a file.

**170L      Error\_BUSY***Message Text:*

The requested resource is in use.

**173L      Error\_CANCEL\_VIOLATION***Message Text:*

A lock request was not outstanding for the supplied cancel region.

**174L      Error\_ATOMIC\_LOCKS\_NOT\_SUPPORTED***Message Text:*

The file system does not support atomic changes to the lock type.

**180L      Error\_INVALID\_SEGMENT\_NUMBER***Message Text:*

The system detected a segment number that was not correct.

**182L      Error\_INVALID\_ORDINAL***Message Text:*

The operating system cannot run %1.

**183L      Error\_ALREADY\_EXISTS***Message Text:*

Cannot create a file when that file already exists.

**186L      Error\_INVALID\_FLAG\_NUMBER***Message Text:*

The flag passed is not correct.

**187L      Error\_SEM\_NOT\_FOUND***Message Text:*

The specified system semaphore name was not found.

**188L      Error\_INVALID\_STARTING\_CODESEG***Message Text:*

The operating system cannot run %1.

- 189L      Error\_INVALID\_STACKSEG**  
*Message Text:*  
The operating system cannot run %1.
- 190L      Error\_INVALID\_MODULETYPE**  
*Message Text:*  
The operating system cannot run %1.
- 191L      Error\_INVALID\_EXE\_SIGNATURE**  
*Message Text:*  
Cannot run %1 in Win32 mode.
- 192L      Error\_EXE\_MARKED\_INVALID**  
*Message Text:*  
operating system cannot run %1.
- 193L      Error\_BAD\_EXE\_FORMAT**  
*Message Text:*  
%1 is not a valid Win32 application.
- 194L      Error\_ITERATED\_DATA\_EXCEEDS\_64k**  
*Message Text:*  
The operating system cannot run %1.
- 195L      Error\_INVALID\_MINALLOCSIZE**  
*Message Text:*  
The operating system cannot run %1.
- 196L      Error\_DYNLINK\_FROM\_INVALID\_RING**  
*Message Text:*  
The operating system cannot run this application program.
- 197L      Error\_IOPL\_NOT\_ENABLED**  
*Message Text:*  
The operating system is not presently configured to run this application.
- 198L      Error\_INVALID\_SEGDPL**  
*Message Text:*  
The operating system cannot run %1.
- 199L      Error\_AUTODATASEG\_EXCEEDS\_64k**  
*Message Text:*  
The operating system cannot run this application program.

**200L      Error\_RING2SEG\_MUST\_BE\_MOVABLE***Message Text:*

The code segment cannot be greater than or equal to 64K.

**201L      Error\_RELOC\_CHAIN\_XEEDS\_SEGLIM***Message Text:*

The operating system cannot run %1.

**202L      Error\_INFLOOP\_IN\_RELOC\_CHAIN***Message Text:*

The operating system cannot run %1.

**203L      Error\_ENVVAR\_NOT\_FOUND***Message Text:*

The system could not find the environment option that was entered.

**205L      Error\_NO\_SIGNAL\_SENT***Message Text:*

No process in the command subtree has a signal handler.

**206L      Error\_Filename\_EXCED\_RANGE***Message Text:*

The filename or extension is too long.

**207L      Error\_RING2\_STACK\_IN\_USE***Message Text:*

The ring 2 stack is in use.

**208L      Error\_META\_EXPANSION\_TOO\_LONG***Message Text:*

The global filename characters, \* or ?, are entered incorrectly or too many global filename characters are specified.

**209L      Error\_INVALID\_SIGNAL\_NUMBER***Message Text:*

The signal being posted is not correct.

**210L      Error\_THREAD\_1\_INACTIVE***Message Text:*

The signal handler cannot be set.

**212L      Error\_LOCKED***Message Text:*

The segment is locked and cannot be reallocated.

**214L      Error\_TOO\_MANY\_MODULES***Message Text:*

Too many dynamic-link modules are attached to this program or dynamic-link module.

**215L      Error\_NESTING\_NOT\_ALLOWED***Message Text:*

Can't nest calls to LoadModule.

**216L      Error\_EXE\_MACHINE\_TYPE\_MISMATCH***Message Text:*

The image file %1 is valid, but is for a machine type other than the current machine.

**230L      Error\_BAD\_PIPE***Message Text:*

The pipe state is invalid.

**231L      Error\_PIPE\_BUSY***Message Text:*

All pipe instances are busy.

**232L      Error\_NO\_DATA***Message Text:*

The pipe is being closed.

**233L      Error\_PIPE\_NOT\_CONNECTED***Message Text:*

No process is on the other end of the pipe.

**234L      Error\_MORE\_DATA***Message Text:*

More data is available.



**240L      Error\_VC\_DISCONNECTED***Message Text:*

The session was canceled.

**254L      Error\_INVALID\_EA\_NAME***Message Text:*

The specified extended attribute name was invalid.

**255L      Error\_EA\_LIST\_INCONSISTENT***Message Text:*

The extended attributes are inconsistent.

**259L      Error\_NO\_MORE\_ITEMS***Message Text:*

No more data is available.

**266L      Error\_CANNOT\_COPY***Message Text:*

The copy functions cannot be used.

**267L      Error\_DIRECTORY***Message Text:*

The directory name is invalid.

**275L      Error\_EAS\_DIDNT\_FIT***Message Text:*

The extended attributes did not fit in the buffer.

**276L      Error\_EA\_FILE\_CORRUPT***Message Text:*

The extended attribute file on the mounted file system is corrupt.

**277L      Error\_EA\_TABLE\_FULL***Message Text:*

The extended attribute table file is full.

**278L      Error\_INVALID\_EA\_HANDLE***Message Text:*

The specified extended attribute handle is invalid.

**282L      Error\_EAS\_NOT\_SUPPORTED***Message Text:*

The mounted file system does not support extended attributes.

**288L      Error\_NOT\_OWNER***Message Text:*

Attempt to release mutex not owned by caller.

**298L      Error\_TOO\_MANY\_POSTS***Message Text:*

Too many posts were made to a semaphore.

**299L      Error\_PARTIAL\_COPY***Message Text:*

Only part of a ReadProcessMemory or WriteProcessMemory request was completed.

**300L      Error\_OPLOCK\_NOT\_GRANTED***Message Text:*

The oplock request is denied.

**301L      Error\_INVALID\_OPLOCK\_PROTOCOL***Message Text:*

An invalid oplock acknowledgment was received by the system.

**317L      Error\_MR\_MID\_NOT\_FOUND***Message Text:*

The system cannot find message text for message number 0x%1 in the message file for %2.

**487L      Error\_INVALID\_ADDRESS***Message Text:*

Attempt to access invalid address.

**534L      Error\_ARITHMETIC\_OVERFLOW***Message Text:*

Arithmetic result exceeded 32 bits.

**535L      Error\_PIPE\_CONNECTED***Message Text:*

There is a process on other end of the pipe.

**536L      Error\_PIPE\_LISTENING***Message Text:*

Waiting for a process to open the other end of the pipe.

**994L      Error\_EA\_ACCESS\_DENIED***Message Text:*

Access to the extended attribute was denied.

**995L      Error\_OPERATION\_ABORTED***Message Text:*

The I/O operation has been aborted because of either a thread exit or an application request.

**996L      Error\_IO\_INCOMPLETE***Message Text:*

Overlapped I/O event is not in a signaled state.

**997L      Error\_IO\_PENDING***Message Text:*

Overlapped I/O operation is in progress.

**998L      Error\_NOACCESS***Message Text:*

Invalid access to memory location.

**999L      Error\_SWAPError***Message Text:*

Error performing inpage operation.

**1001L      Error\_STACK\_OVERFLOW***Message Text:*

Recursion too deep; the stack overflowed.

**1002L      Error\_INVALID\_MESSAGE***Message Text:*

The window cannot act on the sent message.

**1003L      Error\_CAN\_NOT\_COMPLETE***Message Text:*

Cannot complete this function.

**1004L      Error\_INVALID\_FLAGS***Message Text:*

Invalid flags.

**1005L      Error\_UNRECOGNIZED\_VOLUME***Message Text:*

The volume does not contain a recognized file system. Please make sure that all required file system drivers are loaded and that the volume is not corrupted.

**1006L      Error\_FILE\_INVALID***Message Text:*

The volume for a file has been externally altered so that the opened file is no longer valid.

**1007L      Error\_FULLSCREEN\_MODE***Message Text:*

The requested operation cannot be performed in full-screen mode.

**1008L      Error\_NO\_TOKEN***Message Text:*

An attempt was made to reference a token that does not exist.

**1009L      Error\_BADDB***Message Text:*

The configuration registry database is corrupt.

**1010L      Error\_BADKEY***Message Text:*

The configuration registry key is invalid.

**1011L      Error\_CANTOPEN***Message Text:*

The configuration registry key could not be opened.

**1012L      Error\_CANTREAD***Message Text:*

The configuration registry key could not be read.

**1013L      Error\_CANTWRITE***Message Text:*

The configuration registry key could not be written.

**1014L      Error\_REGISTRY\_RECOVERED***Message Text:*

One of the files in the registry database had to be recovered by use of a log or alternate copy. The recovery was successful.

**1015L      Error\_REGISTRY\_CORRUPT***Message Text:*

The registry is corrupted. The structure of one of the files that contains registry data is corrupted, or the system's image of the file in memory is corrupted, or the file could not be recovered because the alternate copy or log was absent or corrupted.

**1016L      Error\_REGISTRY\_IO\_FAILED***Message Text:*

An I/O operation initiated by the registry failed unrecoverably. The registry could not read in, or write out, or flush, one of the files that contain the system's image of the registry.

**1017L      Error\_NOT\_REGISTRY\_FILE***Message Text:*

The system has attempted to load or restore a file into the registry, but the specified file is not in a registry file format.

**1018L      Error\_KEY\_DELETED***Message Text:*

Illegal operation attempted on a registry key that has been marked for deletion.

**1019L      Error\_NO\_LOG\_SPACE***Message Text:*

System could not allocate the required space in a registry log.

**1020L      Error\_KEY\_HAS\_CHILDREN***Message Text:*

Cannot create a symbolic link in a registry key that already has subkeys or values.

**1021L      Error\_CHILD\_MUST\_BE\_VOLATILE***Message Text:*

Cannot create a stable subkey under a volatile parent key.

**1022L      Error\_NOTIFY\_ENUM\_DIR***Message Text:*

A notify change request is being completed and the information is not being returned in the caller's buffer. The caller now needs to enumerate the files to find the changes.

**1051L      Error\_DEPENDENT\_SERVICES\_RUNNING***Message Text:*

A stop control has been sent to a service that other running services are dependent on.

**1052L      Error\_INVALID\_SERVICE\_CONTROL***Message Text:*

The requested control is not valid for this service.

**1053L      Error\_SERVICE\_REQUEST\_TIMEOUT***Message Text:*

The service did not respond to the start or control request in a timely fashion.

**1054L      Error\_SERVICE\_NO\_THREAD***Message Text:*

A thread could not be created for the service.

**1055L      Error\_SERVICE\_DATABASE\_LOCKED***Message Text:*

The service database is locked.

**1056L      Error\_SERVICE\_ALREADY\_RUNNING***Message Text:*

An instance of the service is already running.

**1057L      Error\_INVALID\_SERVICE\_ACCOUNT***Message Text:*

The account name is invalid or does not exist.

**1058L      Error\_SERVICE\_DISABLED***Message Text:*

The service cannot be started, either because it is disabled or because it has no enabled devices associated with it.

**1059L      Error\_CIRCULAR\_DEPENDENCY***Message Text:*

Circular service dependency was specified.

**1060L      Error\_SERVICE\_DOES\_NOT\_EXIST***Message Text:*

The specified service does not exist as an installed service.

**1061L      Error\_SERVICE\_CANNOT\_ACCEPT\_CTRL***Message Text:*

The service cannot accept control messages at this time.

**1062L      Error\_SERVICE\_NOT\_ACTIVE***Message Text:*

The service has not been started.

**1063L      Error\_FAILED\_SERVICE\_CONTROLLER\_CONNECT***Message Text:*

The service process could not connect to the service controller.

**1064L      Error\_EXCEPTION\_IN\_SERVICE***Message Text:*

An exception occurred in the service when handling the control request.

**1065L      Error\_DATABASE\_DOES\_NOT\_EXIST***Message Text:*

The database specified does not exist.

**1066L      Error\_SERVICE\_SPECIFIC\_Error***Message Text:*

The service has returned a service-specific error code.

**1067L      Error\_PROCESS\_ABORTED***Message Text:*

The process terminated unexpectedly.

**1068L      Error\_SERVICE\_DEPENDENCY\_FAIL***Message Text:*

The dependency service or group failed to start.

**1069L      Error\_SERVICE\_LOGON\_FAILED***Message Text:*

The service did not start due to a logon failure.

**1070L      Error\_SERVICE\_START\_HANG***Message Text:*

After starting, the service hung in a start-pending state.

**1071L      Error\_INVALID\_SERVICE\_LOCK***Message Text:*

The specified service database lock is invalid.

**1072L      Error\_SERVICE\_MARKED\_FOR\_DELETE***Message Text:*

The specified service has been marked for deletion.

**1073L      Error\_SERVICE\_EXISTS***Message Text:*

The specified service already exists.

**1074L      Error\_ALREADY\_RUNNING\_LKG***Message Text:*

The system is currently running with the last-known-good configuration.

**1075L      Error\_SERVICE\_DEPENDENCY\_DELETED***Message Text:*

The dependency service does not exist or has been marked for deletion.

**1076L      Error\_BOOT\_ALREADY\_ACCEPTED***Message Text:*

The current boot has already been accepted for use as the last-known-good control set.

**1077L      Error\_SERVICE\_NEVER\_STARTED***Message Text:*

No attempts to start the service have been made since the last boot.

**1078L      Error\_DUPLICATE\_SERVICE\_NAME***Message Text:*

The name is already in use as either a service name or a service display name.

**1079L      Error\_DIFFERENT\_SERVICE\_ACCOUNT***Message Text:*

The account specified for this service is different from the account specified for other services running in the same process.

**1080L      Error\_CANNOT\_DETECT\_DRIVER\_FAILURE***Message Text:*

Failure actions can only be set for Win32 services, not for drivers.

**1081L      Error\_CANNOT\_DETECT\_PROCESS\_ABORT***Message Text:*

This service runs in the same process as the service control manager. Therefore, the service control manager cannot take action if this service's process terminates unexpectedly.



**1082L      Error\_NO\_RECOVERY\_PROGRAM***Message Text:*

No recovery program has been configured for this service.

**1100L      Error\_END\_OF\_MEDIA***Message Text:*

The physical end of the tape has been reached.

**1101L      Error\_FILEMARK\_DETECTED***Message Text:*

A tape access reached a filemark.

**1102L      Error\_BEGINNING\_OF\_MEDIA***Message Text:*

The beginning of the tape or a partition was encountered.

**1103L      Error\_SETMARK\_DETECTED***Message Text:*

A tape access reached the end of a set of files.

**1104L      Error\_NO\_DATA\_DETECTED***Message Text:*

No more data is on the tape.

**1105L      Error\_PARTITION\_FAILURE***Message Text:*

Tape could not be partitioned.

**1106L      Error\_INVALID\_BLOCK\_LENGTH***Message Text:*

When accessing a new tape of a multivolume partition, the current blocksize is incorrect.

**1107L      Error\_DEVICE\_NOT\_PARTITIONED***Message Text:*

Tape partition information could not be found when loading a tape.

**1108L      Error\_UNABLE\_TO\_LOCK\_MEDIA***Message Text:*

Unable to lock the media eject mechanism.

**1109L      Error\_UNABLE\_TO\_UNLOAD\_MEDIA***Message Text:*

Unable to unload the media.

**1110L      Error\_MEDIA\_CHANGED***Message Text:*

The media in the drive may have changed.

**1111L      Error\_BUS\_RESET***Message Text:*

The I/O bus was reset.

**1112L      Error\_NO\_MEDIA\_IN\_DRIVE***Message Text:*

No media in drive.

**1113L      Error\_NO\_UNICODE\_TRANSLATION***Message Text:*

No mapping for the Unicode character exists in the target multi-byte code page.

**1114L      Error\_DLL\_INIT\_FAILED***Message Text:*

A dynamic link library (DLL) initialization routine failed.

**1115L      Error\_SHUTDOWN\_IN\_Progress***Message Text:*

A system shutdown is in progress.

**1116L      Error\_NO\_SHUTDOWN\_IN\_Progress***Message Text:*

Unable to abort the system shutdown because no shutdown was in progress.

**1117L      Error\_IO\_DEVICE***Message Text:*

The request could not be performed because of an I/O device error.

**1118L      Error\_SERIAL\_NO\_DEVICE***Message Text:*

No serial device was successfully initialized. The serial driver will unload.

**1119L      Error\_IRQ\_BUSY***Message Text:*

Unable to open a device that was sharing an interrupt request (IRQ) with other devices. At least one other device that uses that IRQ was already opened.

**1120L      Error\_MORE\_WRITES***Message Text:*

A serial I/O operation was completed by another write to the serial port. (The IOCTL\_SERIAL\_XOFF\_COUNTER reached zero.)

**1121L      Error\_COUNTER\_TIMEOUT***Message Text:*

A serial I/O operation completed because the timeout period expired. (The IOCTL\_SERIAL\_XOFF\_COUNTER did not reach zero.)

**1122L      Error\_FLOPPY\_ID\_MARK\_NOT\_FOUND***Message Text:*

No ID address mark was found on the floppy disk.

**1123L      Error\_FLOPPY\_WRONG\_CYLINDER***Message Text:*

Mismatch between the floppy disk sector ID field and the floppy disk controller track address.

**1124L      Error\_FLOPPY\_UNKNOWN\_Error***Message Text:*

The floppy disk controller reported an error that is not recognized by the floppy disk driver.

**1125L      Error\_FLOPPY\_BAD\_REGISTERS***Message Text:*

The floppy disk controller returned inconsistent results in its registers.

**1126L      Error\_DISK\_RECALIBRATE\_FAILED***Message Text:*

While accessing the hard disk, a recalibrate operation failed, even after retries.

**1127L      Error\_DISK\_OPERATION\_FAILED***Message Text:*

While accessing the hard disk, a disk operation failed even after retries.

**1128L      Error\_DISK\_RESET\_FAILED***Message Text:*

While accessing the hard disk, a disk controller reset was needed, but

**1129L      Error\_EOM\_OVERFLOW**

*Message Text:*

Physical end of tape encountered.

**1130L      Error\_NOT\_ENOUGH\_SERVER\_MEMORY**

*Message Text:*

Not enough server storage is available to process this command.

**1131L      Error\_POSSIBLE\_DEADLOCK**

*Message Text:*

A potential deadlock condition has been detected.

**1132L      Error\_MAPPED\_ALIGNMENT**

*Message Text:*

The base address or the file offset specified does not have the proper alignment.

**1140L      Error\_SET\_POWER\_STATE\_VETOED**

*Message Text:*

An attempt to change the system power state was vetoed by another application or driver.

**1141L      Error\_SET\_POWER\_STATE\_FAILED**

*Message Text:*

The system BIOS failed an attempt to change the system power state.

**1142L      Error\_TOO\_MANY\_LINKS**

*Message Text:*

An attempt was made to create more links on a file than the file system supports.

**1150L      Error\_OLD\_WIN\_VERSION**

*Message Text:*

The specified program requires a newer version of Windows.

**1151L      Error\_APP\_WRONG\_OS**

*Message Text:*

The specified program is not a Windows or MS-DOS program.

**1152L      Error\_SINGLE\_INSTANCE\_APP**

*Message Text:*

Cannot start more than one instance of the specified program.

**1153L      Error\_RMODE\_APP***Message Text:*

The specified program was written for an earlier version of Windows.

**1154L      Error\_INVALID\_DLL***Message Text:*

One of the library files needed to run this application is damaged.

**1155L      Error\_NO\_ASSOCIATION***Message Text:*

No application is associated with the specified file for this operation.

**1156L      Error\_DDE\_FAIL***Message Text:*

An error occurred in sending the command to the application.

**1157L      Error\_DLL\_NOT\_FOUND***Message Text:*

One of the library files needed to run this application cannot be found.

**1158L      Error\_NO\_MORE\_USER\_HANDLES***Message Text:*

The current process has used all of its system allowance of handles for Window Manager objects.

**1159L      Error\_MESSAGE\_SYNC\_ONLY***Message Text:*

The message can be used only with synchronous operations.

**1160L      Error\_SOURCE\_ELEMENT\_EMPTY***Message Text:*

The indicated source element has no media.

**1161L      Error\_Destination\_ELEMENT\_FULL***Message Text:*

The indicated destination element already contains media.

**1162L      Error\_ILLEGAL\_ELEMENT\_ADDRESS***Message Text:*

The indicated element does not exist.

**1163L      Error\_MAGAZINE\_NOT\_PRESENT***Message Text:*

The indicated element is part of a magazine that is not present.

**1164L      Error\_DEVICE\_REINITIALIZATION\_NEEDED***Message Text:*

The indicated device requires reinitialization due to hardware errors.

**1165L      Error\_DEVICE\_REQUIRES\_CLEANING***Message Text:*

The device has indicated that cleaning is required before further operations are attempted.

**1166L      Error\_DEVICE\_DOOR\_OPEN***Message Text:*

The device has indicated that its door is open.

**1167L      Error\_DEVICE\_NOT\_CONNECTED***Message Text:*

The device is not connected.

**1168L      Error\_NOT\_FOUND***Message Text:*

Element not found.

**1169L      Error\_NO\_MATCH***Message Text:*

There was no match for the specified key in the index.

**1170L      Error\_SET\_NOT\_FOUND***Message Text:*

The property set specified does not exist on the object.

**1171L      Error\_POINT\_NOT\_FOUND***Message Text:*

The point passed to GetMouseMovePoints is not in the buffer.

**1172L      Error\_NO\_TRACKING\_SERVICE***Message Text:*

The tracking (workstation) service is not running.

**1173L      Error\_NO\_VOLUME\_ID***Message Text:*

The Volume ID could not be found.