



Documentazione libreria utente CndexLink Interfaccia al server Cndex

INTRODUZIONE	6
Il server Cndex.....	8
Descrizione generale delle funzioni	9
Tabelle delle compatibilità.....	11
Gestione del Server Cndex e connessione con il CNC.....	16
ConnectServer_C (funzione obsoleta)	17
ReleaseServer_C (funzione obsoleta)	18
OpenSession_C	19
CloseSession_C	21
Funzioni per il controllo della fase di bootstrap del CNC.....	21
BootPhaseEnquiry_C.....	23
BootReboot_C	24
BootShutDown_C	25
BootMode_C	26
BootEnterServer_C (solo per Serie 10)	27
BootLeaveServer_C (solo per Serie 10)	28
GetHWKey_C	29
Gestione variabili	30
ReadVarWord_C.....	31
ReadVarDouble_C	33
WriteVarWord_C	35
WriteVarWordBit_C.....	37
WriteVarDouble_C	38
ReadVarText_C	39
WriteVarText_C.....	41
Funzioni dedicate al PLC	42
ReadWarningMsg_C.....	43
GetPLVarWord_C (funzione obsoleta)	44
SetPLVarWord_C (funzione obsoleta).....	45
GetPLVarDouble_C (funzione obsoleta)	46
SetPLVarDouble_C (funzione obsoleta)	47
GetPLVarAscii_C (funzione obsoleta).....	48
SetPLVarAscii_C (funzione obsoleta)	49
GetPLStreamWord_C (funzione obsoleta)	50
SetPLStreamWord_C (funzione obsoleta)	52
GetPLStreamDouble_C (funzione obsoleta)	54
SetPLStreamDouble_C (funzione obsoleta).....	56
GetAxisTabRecord_C	58
SetAxisTabRecord_C.....	59
GetToolTabRecord_C	60
SetToolTabRecord_C	61
GetOffsetTabRecord_C	62
SetOffsetTabRecord_C.....	63
GetUserTabRecord_C.....	64
SetUserTabRecord_C	65
GetTabFieldDouble_C	66
SetTabFieldDouble_C.....	67
GetTabFieldShort_C.....	68
SetTabFieldShort_C	69



Documentazione libreria utente CndexLink Interfaccia al server Cndex

TabSearchDouble_C	70
TabSearchShort_C	72
SetTabSem_C	74
TabSemInfo_C	76
ResetSingleTableII_C	77
LockTableII_C	78
UnLockTableII_C	79
GetOriginTabRecordII_C	80
SetOriginTabRecordII_C	81
GetToolTabRecordII_C	82
SetToolTabRecordII_C	83
GetOffsetTabRecordII_C	84
SetOffsetTabRecordII_C	85
GetUserTabRecordII_C	86
SetUserTabRecordII_C	87
GetMagazineTabRecordII_C	88
SetMagazineTabRecordII_C	89
GetPocketTabRecordII_C	90
SetPocketTabRecordII_C	91
ToolSearch_C	92
ToolSearchLife_C	93
ToolSearchDir_C	94
ToolSearchLifeDir_C	96
ToolSameFamily_C	98
ToolLoad_C	99
ToolUnload_C	100
PocketSearch_C	102
PocketClosest_C	103
PocketNearest_C	105
ToolLife_C	106
LoadTableFromFile_C	107
SaveTables_C	109
SaveSingleTable_C	110
RestoreSingleTable_C	111
SaveBackupMemory_C	112
RestoreBackupMemory_C	113
Funzioni dedicate al Processo	115
Cycle_C	116
SyncroCycle_C	117
Reset_C	118
Hold_C	119
SetFeedManOver_C	120
SetFeedRateOver_C	121
SetFeedRapidOver_C	122
SetSpeedRateOver_C	123
SetManMovDirection_C	124
GetVarJOG_C	125
SetVarJOG_C	126
SetVarUAS_C	127
GetVarRCM_C	128
SetVarRCM_C	129
LoadPTech_C	130
GetPTechSizes_C	132



Documentazione libreria utente CndexLink Interfaccia al server Cndex

SetMdiString_C.....	133
SetProcessMode_C	134
SelectProcess_C	135
GetSelectedProcess_C.....	136
SelectProcAxis_C	137
SelectPartProgram_C	138
SelectPartProgramFromDrive_C.....	139
ManagePartProgram_C	140
GetActivePartProgram_C.....	142
GetActivePartProgramFullPath_C	143
GetPartProgramLines _C	144
GetAxOriginNum_C.....	145
GetAxesPosition_C.....	146
GetNcInfo1_C	148
GetNcInfo2_C	149
GetMarkerInfo_C.....	150
GetToolNames_C.....	151
GetProcessStatus_C.....	152
GetBlkNum_C.....	153
GetVarE_C (funzione obsoleta).....	154
SetVarE_C (funzione obsoleta)	155
GetVarSN_C (funzione obsoleta).....	156
SetVarSN_C (funzione obsoleta)	157
GetVarSC_C (funzione obsoleta).....	158
SetVarSC_C (funzione obsoleta)	159
ReadErrMsg_C.....	160
ReadPartProgramMsg_C	161
ReadCurrentErrorMsg_C.....	162
ReadCurrentEmergMsg_C	163
ReadCurrentAnomalyMsg_C	164
GetGCode_C	165
GetMCode_C.....	166
SkipPProgBlock_C.....	167
Ese_C.....	168
EseEx_C.....	169
AxesRef_C	171
SetProcVarWord_C	172
GetProcVarWord_C.....	174
Funzioni per la "Modalità passante"	176
DncInit_C.....	177
DncData_C.....	178
DncEof_C.....	179
DncStop_C.....	180
Funzioni generiche.....	181
GetAxesInfo3_C	182
GetCodeNumber_C.....	183
GetOptions_C.....	184
GetDateTime_C.....	185
SetDateTime_C	187
GRead_C	189
GWrite_C	191
SetIpAddress_C.....	193
GetSerialNumber_C	195



Documentazione libreria utente CndexLink Interfaccia al server Cndex

ReadRemapDefinitions_C	196
WriteRemapDefinitions_C	197
GetServoPar_C	198
SetServoPar_C	199
GetCNCRegKey_C	200
Funzioni per la gestione del file system	201
LogFSTransferFile_C	202
LogFSTransferFileW_C	204
LogFSSetSecurityLevel_C	205
LogFSGetSecurityLevel_C	206
LogFSLongFileNames_C	207
LogFSGetNumDrive_C	208
LogFSGetDriveList_C	209
LogFSGetHiddenDriveList_C	210
LogFSGetDrivePath_C	211
LogFSAddDrive_C	212
LogFSRemoveDrive_C	214
LogFSReloadDriveList_C	215
LogFSCreateDir_C	216
LogFSCreateFile_C	217
LogFSGetFileSize_C	218
LogFSGetFileAttrib_C	219
LogFSSetFileAttrib_C	222
LogFSChangeFileAttrib_C	223
LogFSFindFirst_C	224
LogFSFindNext_C	226
LogFSFindClose_C	227
LogFSRemoveFile_C	228
LogFSRemoveDir_C	229
LogFSRename_C	230
LogFSCopyFile_C	231
LogFSGetInfo_C	232
Funzioni per la gestione dei nomi file lunghi	234
PPInsertName_C	235
PPDeleteName_C	236
PPGetPLogicalName_C	237
PPGetPhysicalName_C	238
PPGetLogicalDir_C	239
PPUpdate_C	240
Funzioni per la gestione CANOPEN HILSCHER	241
CANInit_C	242
CANBoard_C	244
CANSync_C	246
CANNMT_C	248
CANReadSDO_C	250
CANWriteSDO_C	252
CANGetEmergency_C	255
CANConsoleCfg_C	258
Strumenti utilizzabili da riga di comando	260
Introduzione	260
Sintassi	260
Comandi	261
Opzioni	262



Documentazione libreria utente CndexLink Interfaccia al server Cndex

Esempi pratici	262
Descrizione delle strutture e definizioni	264
Gestione degli errori	273
Classe degli errori	274
Codici di errore COM (classe 1)	275
Codici di errore del server Cndex (classe 2)	276
Codici di errore del protocollo di rete NETBIOS (classe 3)	277
Codici di errore Serie 10 (classe 4)	278
Codici di errore FILESYS_CLASS (classe 5)	279
Codici di errore FILESYS_ERRNUM_CLASS (classe 6)	280
Codici di errore CNC_BOOT_CLASS (classe 7)	281
Codici di errore SERVER_EXCEPTION_CLASS (classe 8)	282
Codici di errore CNDEXLINK_DLL_ERR_CLASS (classe 9)	283
Codici di errore DLL_INTERFACE_ERR_CLASS (classe 10)	284
Codici di errore SOAP_INTERFACE_ERR_CLASS (classe 11)	285
Codici di errore OPENcontrol (classi da 17 a 62)	287

Ultima modifica: 20 gennaio 2016
Versione WinNBI: 4.3.4



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

INTRODUZIONE

Tutti i sistemi OSAI (CNC, GMC, PC, Pannelli operatore, Terminali portatili etc.), connessi in rete, possono comunicare, tra loro e con applicativi utente, attraverso un server OSAI chiamato Cndex e sviluppato con tecnologia DCOM (Microsoft®).

In generale ogni sistema OSAI è identificato da una istanza del server Cndex.

Il server Cndex è facilmente utilizzabile tramite una DLL OSAI chiamata CndexLink. CndexLink svolge internamente tutte le funzioni di gestione del server DCOM ed implementa, in modo trasparente per l'utilizzatore, il protocollo di comunicazione.

Questo documento spiega l'utilizzo della DLL CndexLink utilizzata in modalità utente (CndexLinkUser).

Tutti i componenti necessari per colloquiare con i sistemi OSAI sono presenti sul PC dopo l'installazione del prodotto WinNBI (interfaccia grafica in ambiente Windows per sistemi OSAI - codice C07), a partire dalla versione 2.4 e sui vari sistemi dopo aver installato il software specifico della macchina.

A partire dalla versione 3.1.1 della WinNBI è anche disponibile la DLL di interfaccia da utilizzare in ambiente di sviluppo Microsoft® .NET.

In particolare, per il funzionamento del meccanismo di comunicazione, devono essere presenti sul sistema i seguenti file:

Cndex.exe	Server Cndex (può anche risiedere su un altro sistema e non essere presente sul sistema che ospita l'applicazione).
Cndexps.dll	Libreria Proxy per l'aggancio del server DCOM.
CndexLink.dll	Libreria per l'utilizzo del server Cndex.

I seguenti file sono invece a disposizione per la generazione (compilazione e link) delle applicazioni che utilizzano il server Cndex:

CndexLink.lib	Libreria per il link statico della libreria utente CndexLink
CndexLinkUser.h	File di include per linguaggio C/C++ che contiene i prototipi delle funzioni della libreria e la definizione delle costanti, dati e strutture necessarie a CndexLink
CndexLinkUser.bas	Modulo per Visual Basic che contiene le dichiarazioni delle funzioni e la definizione delle costanti dati e strutture necessarie a CndexLink
CndexLinkDotNet.dll	Libreria per l'utilizzo di CndexLink in applicazioni basate su Microsoft® .NET.

La DLL CndexLink e' stata certificata per applicazioni sviluppate in Microsoft® C++ e Microsoft® Visual Basic 6.0.

La DLL CndexLinkDotNet e' stata certificata per applicazioni che usano Microsoft® .NET Framwork e CompactFramwork 2.0.



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

CndexLink e' stata sviluppata in modo da risultare molto simile, come interfaccia, alla DLL OSAI S10WLK32.DLL contenuta nell'opzione obsoleta E66 "Ethernet Mini-Dnc Communication", precedentemente usata per la comunicazione con i CNC Serie 10.

ATTENZIONE :

La DLL permette di collegarsi solamente ai CNC dotati dell'opzione A06 "CndexLink communication". Nel caso in cui l'opzione non sia presente sul CNC la funzione OpenSession_C ritornera' l'errore classe DLL_INTERFACE (valore 10) codice ERR_OPTION_NOT_ENABLED (valore 5)



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

Il server Cndex

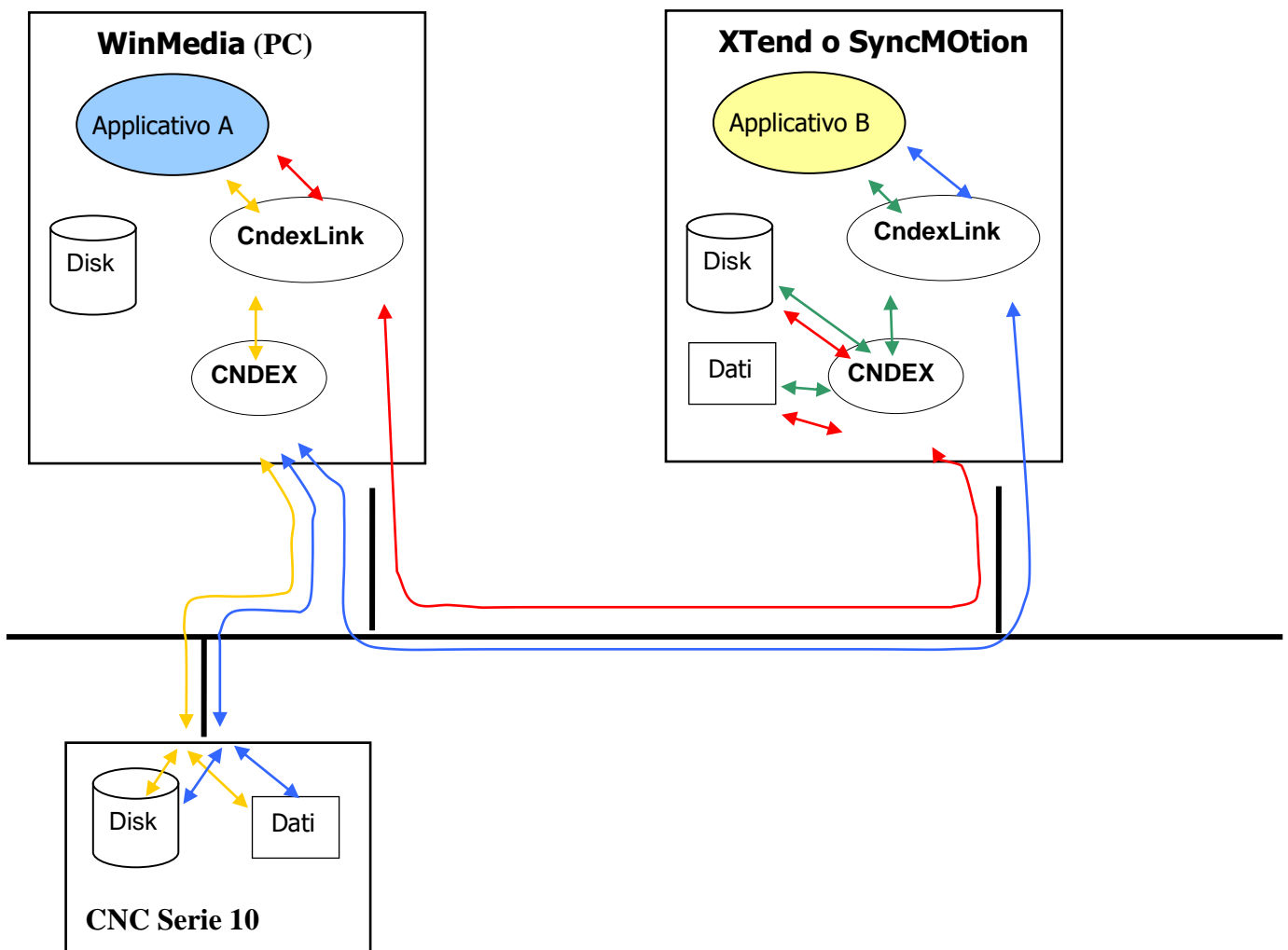
Come accennato nell'introduzione, tutti i sistemi OSAI possono comunicare tra loro tramite un server chiamato Cndex.

Collegando diversi sistemi in una rete locale, ciascun sistema sarà in grado di comunicare con gli altri. Per fare questo un applicativo dovrà istanziare un oggetto remoto (Cndex) per ogni sistema al quale si vuole collegare.

Con eccezione per la Serie 10, il server deve essere istanziato sul sistema con il quale si vuole comunicare, questa operazione è svolta in modo automatico dalla funzione `OpenSession_C`.

Nella figura seguente sono indicati due applicativi generici (A e B) ed i loro collegamenti per ottenere dati da diversi tipi di macchine. L'applicativo A è collegato al server locale per comunicare con il CNC Serie10 (freccie arancioni). Per i sistemi OPENControl e SyncMotion il server è invece a bordo del CNC (collegamento evidenziato dalla frecce rosse).

L'applicativo B è collegato al server locale per raccogliere dati del CNC locale (freccie verdi) ed al server sul WinMedia per colloquiare con il CNC Serie10 (freccie blu).





Documentazione CndexLinkUser DLL Interfaccia al server Cndex

Descrizione generale delle funzioni

Le funzioni di scambio dati con il CNC utilizzano il server OSAI Cndex descritto nei paragrafi precedenti.

Per le applicazioni .NET è necessario inserire nel progetto un "reference" alla libreria di interfaccia CndexLinkDotNet (click con tasto destro del mouse sul nome progetto nel Solution Explorer di VS2005, comando "add reference", tab "Browse" e selezione del file CndexLinkDotNet.Dll fornito assieme all'installazione della WinNBI).

Prima di poter chiamare le funzioni di scambio dati con il CNC è necessario collegarsi al server ed aprire una sessione di comunicazione con il CNC.

Per collegarsi al server e gestire le sessioni di comunicazione sono disponibili le funzioni OpenSession_C e CloseSession_C (si veda anche il capitolo "Funzioni per il controllo della fase di bootstrap del CNC").

La sequenza corretta per l'attivazione delle funzioni di scambio dati, assumendo che il CNC sia già acceso in modalità RUN, è la seguente:

Applicazioni C++ e VB6

```
OpenSession_C(...)           // Apre una sessione di comunicazione con un CNC
BootPhaseEnquiry_C(...)      // Per verificare che il CNC sia nella modalità RUN
.
.
ReadVarWord_C(...)           // Esempio di funzioni di scambio dati
WriteVarDouble_C(...)
.
.
CloseSession_C(...)          // Chiude la sessione di comunicazione
```

Applicazioni .NET (C#)

```
using CndexLinkDotNet;       // Permette l'uso di CndexLinkDotNet.
                              // La libreria deve essere aggiunta nei reference.

CndexLinkDotNet.Cndex Server; // Riferimento al Server Cndex.
Server = new CndexLinkDotNet.Cndex(); // Crea un'istanza del server.

Server.OpenSession_C(...)     // Apre una sessione di comunicazione con un CNC.

Server.BootPhaseEnquiry_C(...) // per verificare che il CNC sia nella modalità RUN.
.
.
Server.ReadVarWord_C(...)     // Esempio di funzioni di scambio dati.
Server.WriteVarDouble_C(...)
.
.
Server.CloseSession_C(...)    // Chiude la sessione di comunicazione con il CNC.
```



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

ConnectServer_C / ReleaseServer_C.

Con le versioni precedenti di questa libreria era indispensabile utilizzare le funzioni ConnectServer_C e ReleaseServer_C che servivano per creare e rilasciare il server.

Con questa versione le due funzioni, anche se continuano ad essere disponibili per compatibilità con programmi precedenti, non eseguono alcuna operazione.

Codici di errore

Per la gestione errori di tutte le funzioni, la libreria CndexLink utilizza il codice di ritorno e due variabili (puntate da pErrClass e pErrNum nei prototipi delle funzioni).

Il codice di ritorno è sempre una "WORD". Un valore diverso da 0 significa che la funzione è stata eseguita senza errori, zero significa errore durante l'esecuzione.

Quando il codice di ritorno vale zero le variabili puntate da pErrClass e pErrNum riportano la Classe ed il Codice relativi all'errore che si è verificato.

Gli errori sono stati organizzati usando diverse classi per evitare la sovrapposizione dei codici di errore provenienti da livelli differenti e per poter individuare immediatamente la tipologia dell'errore.

Ogni classe identifica un livello nella gerarchia di moduli utilizzati per comunicare con il CNC.

Per una descrizione dettagliata delle Classi e dei Codici di errore si faccia riferimento al paragrafo "Gestione degli Errori"

Interpretazione dei campi

Nei paragrafi che seguono sono riportate le funzioni della DLL CndexLink. Per ogni funzione è specificato in che "direzione" sono passati i singoli parametri, vale a dire chi assegna il parametro.

I parametri possono essere:

- [in]** parametri compilati dall'utente e forniti in input alla funzione
- [out]** parametri in output dalla funzione (sono sempre puntatori ad un area di memoria utente). La memoria è compilata dalla funzione durante l'esecuzione.
- [in,out]** parametri sia in input che in output dalla funzione (sono sempre puntatori ad un area di memoria utente). La memoria e' compilata dall'utente prima di chiamare la funzione e dalla funzione durante l'esecuzione.

Le funzioni documentate in questo manuale si applicano ai sistemi Serie 10 MC e Power GP, OPENControl, SyncMotion, Top5 e PC.

La compatibilità della singola funzione con i vari sistemi è riportata nelle tabelle di compatibilità nel paragrafo successivo.

In questo documento la sintassi usata per la descrizione dei prototipi delle singole funzioni è quella del linguaggio C++.

La sintassi dei prototipi per VisualBasic 6.0 si trova nel modulo CndexLinkUser.bas (vedi introduzione).

La sintassi delle funzioni per .NET è disponibile come help contestuale scrivendo il codice dell'applicazione.



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

Tabelle delle compatibilità

Nelle seguenti tabelle sono riportate le compatibilità tra le funzioni ed i vari sistemi OSAI.
Le funzioni contrassegnate con • sono disponibili sul sistema riportato nell'intestazione della colonna.
Le righe in grigio indicano le funzioni obsolete (anche se ancora disponibili).

Gestione del Server Cndex e connessione con il CNC

	<i>S10 MC</i>	<i>S10 GP</i>	<i>OPENControl</i>	<i>SyncMotion</i>	<i>Top5</i>	<i>PC</i>
ConnectServer_C	•	•	•	•	•	•
ReleaseServer_C	•	•	•	•	•	•
OpenSession_C	•	•	•	•	•	•
CloseSession_C	•	•	•	•	•	•

Funzioni per il controllo della fase di bootstrap del CNC

	<i>S10 MC</i>	<i>S10 GP</i>	<i>OPENControl</i>	<i>SyncMotion</i>	<i>Top5</i>	<i>PC</i>
BootPhaseEnquiry_C	•	•	•	•	•	
BootReboot_C	•	•	•			
BootShutDown_C	•	•				
BootMode_C	•	•	•			
BootEnterServer_C	•	•				
BootLeaveServer_C	•	•				
GetHWKey_C	•	•	•	•		

Gestione variabili

	<i>S10 MC</i>	<i>S10 GP</i>	<i>OPENControl</i>	<i>SyncMotion</i>	<i>Top5</i>	<i>PC</i>
ReadVarWord_C	•	•	•	•	•	
ReadVarDouble_C	•	•	•	•	•	
WriteVarWord_C	•	•	•	•	•	
WriteVarWordBit_C	•	•	•	•	•	
WriteVarDouble_C	•	•	•	•	•	
ReadVarText_C	•	•	•	•	•	
WriteVarText_C	•	•	•	•	•	

Funzioni dedicate al PLC

	<i>S10 MC</i>	<i>S10 GP</i>	<i>OPENControl</i>	<i>SyncMotion</i>	<i>Top5</i>	<i>PC</i>
ReadWarningMsg_C	•	•	•			
GetPLVarWord_C	•	•				
SetPLVarWord_C	•	•				
GetPLVarDouble_C	•	•				
SetPLVarDouble_C	•	•				
GetPLVarAscii_C	•	•				
SetPLVarAscii_C	•	•				
GetPLStreamWord_C	•	•				
SetPLStreamWord_C	•	•				
GetPLStreamDouble_C	•	•				
SetPLStreamDouble_C	•	•				
GetAxisTabRecord_C	•	•				
SetAxisTabRecord_C	•	•				
GetToolTabRecord_C	•	•				
SetToolTabRecord_C	•	•				
GetOffsetTabRecord_C	•	•				



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

SetOffsetTabRecord_C	•	•	
GetUserTabRecord_C	•	•	
SetUserTabRecord_C	•	•	
GetTabFieldDouble_C	•	•	
SetTabFieldDouble_C	•	•	
GetTabFieldShort_C	•	•	
SetTabFieldShort_C	•	•	
TabSearchDouble_C	•	•	
TabSearchShort_C	•	•	
SetTabSem_C	•	•	
TabSemInfo_C	•	•	
ResetSingleTableII_C			•
LockTableII_C			•
UnLockTableII_C			•
GetOriginTabRecordII_C			•
SetOriginTabRecordII_C			•
GetToolTabRecordII_C			•
SetToolTabRecordII_C			•
GetOffsetTabRecordII_C			•
SetOffsetTabRecordII_C			•
GetUserTabRecordII_C			•
SetUserTabRecordII_C			•
GetMagazineTabRecordII_C			•
SetMagazineTabRecordII_C			•
GetPocketTabRecordII_C			•
SetPocketTabRecordII_C			•
ToolSearch_C			•
ToolSearchLife_C			•
ToolSearchDir_C			•
ToolSearchLifeDir_C			•
ToolSameFamily_C			•
ToolLoad_C			•
ToolUnload_C			•
PocketSearch_C			•
PocketClosest_C			•
PocketNearest_C			•
ToolLife_C			•
LoadTableFromFile_C			•
SaveTables_C			•
SaveSingleTable_C			•
RestoreSingleTable_C			•
SaveBackupMemory_C			•
RestoreBackupMemory_C			•

Funzioni dedicate al Processo

	<i>S10 MC</i>	<i>S10 GP</i>	<i>OPENControl</i>	<i>SyncMotion</i>	<i>Top5</i>	<i>PC</i>
Cycle_C	•		•			
SyncroCycle_C	•					
Reset_C	•		•			
Hold_C	•		•			
SetFeedManOver_C	•		•			
SetFeedRateOver_C	•		•			
SetFeedRapidOver_C	•		•			
SetSpeedRateOver_C	•		•			
SetManMovDirection_C	•		•			
GetVarJOG_C	•		•			
SetVarJOG_C	•		•			



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

SetVarUAS_C	•	•
GetVarRCM_C	•	•
SetVarRCM_C	•	•
LoadPTech_C		•
GetPTechSizes_C		•
SetMdiString_C	•	•
SetProcessMode_C	•	•
SelectProcess_C	•	•
GetSelectedProcess_C	•	•
SelectProcAxis_C	•	•
SelectPartProgram_C	•	•
SelectPartProgramFromDrive_C	•	•
ManagePartProgram_C		•
GetActivePartProgram_C	•	•
GetActivePartProgramFullPath_C		•
GetPartProgramLines_C	•	•
GetAxOriginNum_C	•	•
GetAxesPosition_C	•	•
GetNcInfo1_C	•	•
GetNcInfo2_C	•	•
GetMarkerInfo_C		•
GetToolNames_C	•	•
GetProcessStatus_C	•	•
GetBlkNum_C	•	•
GetVarE_C	•	•
SetVarE_C	•	•
GetVarSN_C	•	•
SetVarSN_C	•	•
GetVarSC_C	•	•
SetVarSC_C	•	•
ReadErrMsg_C	•	
ReadPartProgramMsg_C	•	•
ReadCurrentErrorMsg_C		•
ReadCurrentEmergMsg_C		•
ReadCurrentAnomalyMsg_C		•
GetGCode_C	•	•
GetMCode_C	•	•
SkipPProgBlock_C	•	•
Ese_C	•	•
EseEx_C		•
AxesRef_C	•	•

Funzioni per la "Modalita' passante"

	<i>S10 MC</i>	<i>S10 GP</i>	<i>OPENControl</i>	<i>SyncMotion</i>	<i>Top5</i>	<i>PC</i>
DncInit_C	•					
DncData_C	•					
DncEof_C	•					
DncStop_C	•					

Funzioni generiche

	<i>S10 MC</i>	<i>S10 GP</i>	<i>OPENControl</i>	<i>SyncMotion</i>	<i>Top5</i>	<i>PC</i>
GetAxesInfo3_C	•		•			
GetCodeNumber_C	•	•				
GetOptions_C	•	•	•	•		
GetDateTime_C	•	•	•			
SetDateTime_C	•	•	•			
GRead_C	•	•				



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

GWrite_C	•	•	
SetIpAddress_C			•
GetSerialNumber_C			•
ReadRemapDefinitions_C			•
WriteRemapDefinitions_C			•
GetServoPar_C			•
SetServoPar_C			•

Funzioni per la gestione del file system

	<i>S10 MC</i>	<i>S10 GP</i>	<i>OPENControl</i>	<i>SyncMotion</i>	<i>Top5</i>	<i>PC</i>
LogFSTransferFile_C	•	•	•	•	•	•
LogFSTransferFileW_C	•	•	•	•	•	•
LogFSSetSecurityLevel_C	•	•	•	•	•	•
LogFSGetSecurityLevel_C	•	•	•	•	•	•
LogFSLongFileNames_C	•	•	•	•	•	•
LogFSGetNumDrive_C	•	•	•	•	•	•
LogFSGetDriveList_C	•	•	•	•	•	•
LogFSGetHiddenDriveList_C	•	•	•	•	•	•
LogFSGetDrivePath_C	•	•	•	•	•	•
LogFSAddDrive_C	•	•	•	•	•	•
LogFSRemoveDrive_C	•	•	•	•	•	•
LogFSReloadDriveList_C	•	•	•	•	•	•
LogFSCreateDir_C	•	•	•	•	•	•
LogFSCreateFile_C	•	•	•	•	•	•
LogFSGetFileSize_C	•	•	•	•	•	•
LogFSGetFileAttrib_C	•	•	•	•	•	•
LogFSSetFileAttrib_C	•	•	•	•	•	•
LogFSChangeFileAttrib_C	•	•	•	•	•	•
LogFSFindFirst_C	•	•	•	•	•	•
LogFSFindNext_C	•	•	•	•	•	•
LogFSFindClose_C	•	•	•	•	•	•
LogFSRemoveFile_C	•	•	•	•	•	•
LogFSRemoveDir_C	•	•	•	•	•	•
LogFSRename_C	•	•	•	•	•	•
LogFSCopyFile_C	•	•	•	•	•	•
LogFSGetInfo_C	•	•	•	•	•	•

Funzioni per la gestione dei nomi file lunghi

	<i>S10 MC</i>	<i>S10 GP</i>	<i>OPENControl</i>	<i>SyncMotion</i>	<i>Top5</i>	<i>PC</i>
PPInsertName_C	•					
PPDeleteName_C	•					
PPGetPLogicalName_C	•					
PPGetPhysicalName_C	•					
PPGetLogicalDir_C	•					
PPUpdate_C	•					

Funzioni per la gestione CANOPEN HILSCHER

	<i>S10 MC</i>	<i>S10 GP</i>	<i>OPENControl</i>	<i>SyncMotion</i>	<i>Top5</i>	<i>PC</i>
CANInit_C	•					
CANBoard_C	•					
CANSync_C	•					
CANNMT_C	•					
CANReadSDO_C	•					
CANWriteSDO_C	•					
CANGetEmergency_C	•					
CANConsoleCfg_C	•					



Documentazione CndexLinkUser DLL Interfaccia al server Cndex



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

Gestione del Server Cndex e connessione con il CNC

Queste funzioni permettono di creare e rilasciare istanze del server Cndex e di aprire delle sessioni di comunicazione con il CNC.



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

ConnectServer_C (funzione obsoleta)

Funzione obsoleta. Non esegue nulla. E' disponibile in questa libreria per compatibilità con applicazioni precedenti. Si veda la funzione OpenSession_C.

Crea una istanza del server Cndex localmente, dove risiede l'applicazione, oppure su un PC remoto. Per utilizzare il server su un PC remoto e' necessario che sia presente il protocollo TCP/IP tra i protocolli di rete utilizzati da entrambi i PC (locale e remoto).

Questa funzione deve essere chiamata una sola volta al momento dell'attivazione dell'applicazione client o all'attivazione di un thread.

```
WORD ConnectServer_C (  
    LPSTR          ServerAddress,  
    DWORD * pErrClass,  
    DWORD * pErrNum  
);
```

Parametri

ServerAddress

[in] Puntatore ad una stringa che contiene il nome di rete oppure l'indirizzo TCP/IP del PC remoto su cui deve essere istanziato il server Cndex. Specificare un puntatore ad una stringa vuota se l'istanza deve essere creata sul PC locale.

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Vedi anche

ReleaseServer_C, OpenSession_C, CloseSession_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

ReleaseServer_C (funzione obsoleta)

Funzione obsoleta. Non esegue nulla. E' disponibile in questa libreria per compatibilità con applicazioni precedenti. Si veda la funzione CloseSession_C.

Rilascia l'istanza del server Cndex creata in precedenza con la funzione ConnectServer_C.

Questa funzione deve essere chiamata una sola volta al momento della chiusura dell'applicazione client o al termine di un thread, dopo aver chiuso tutte le sessioni di comunicazione (vedi OpnSession_C e CloseSession_C).

```
WORD ReleaseServer_C (  
    DWORD * pErrClass,  
    DWORD * pErrNum  
);
```

Parametri

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Vedi anche

ConnectServer_C, OpenSession_C, CloseSession_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

OpenSession_C

Crea un'istanza del server Cndex ed apre una sessione di comunicazione con il controllo numerico. La funzione restituisce un identificativo di sessione (UserSession) da utilizzare in tutte le chiamate successive (vedi note per l'utilizzazione della sessione in ambiente multi-thread). Per accedere ai controlli numerici Serie 10 è necessario che sia configurato il protocollo di rete NETBEUI sul PC (locale o remoto) in cui è creata l'istanza del server Cndex.

```
WORD OpenSession_C (  
    LPSTR      RemoteName,  
    WORD       *pUserSession,  
    DWORD      *pErrClass,  
    DWORD      *pErrNum  
);
```

Parametri

RemoteName

[in] Stringa che specifica il nome del controllo numerico che si vuole connettere. Si possono utilizzare indifferentemente lettere maiuscole o minuscole.

Per indirizzare un CNC **Serie10** si deve specificare in RemoteName il nome di rete del CNC.

Per tutti gli altri sistemi OSAI è necessario anteporre il prefisso "**IP.**" al nome di sistema o indirizzo TCP/IP di rete del sistema da collegare.

P.e. per collegare un CNC OPENControl configurato con nome di sistema "OPENControlCNC" bisogna scrivere in RemoteName il testo "**IP.OPENControlCNC**".

Per collegare un CNC Serie10 utilizzando come server un PC in rete, si deve specificare il prefisso "IP." seguito dal nome macchina o indirizzo IP del PC server seguito dal carattere slash "\" seguito dal nome del CNC Serie10.

P.e. per collegare un CNC Serie10 con nome "**CNC_S10**" passando tramite il PC con indirizzo **192.168.1.36** si deve scrivere in RemoteName il testo "**IP. 192.168.1.36\CNC_S10**".

Il solo prefisso "**IP.**" indica su OPENControl e SyncMotion la macchina locale (applicazioni "Windows CE").

pUserSession

[in] Puntatore ad una variabile nella quale verrà restituito l'identificatore della sessione di comunicazione con il controllo numerico specificato. L'identificatore deve essere usato in input alle altre funzioni per fare riferimento al CNC connesso.

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione è stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

Note

Normalmente le sessioni di comunicazione aperte con questa funzione possono essere utilizzate solo dal thread che ha chiamato la funzione. In altre parole, l'identificatore della sessione aperta, per esempio, dal processo principale di una applicazione, può essere utilizzato in un qualsiasi punto di quel processo ma non può essere "passato" ad un thread.

Questo è dovuto alla modalità di inizializzazione del "DCOM threading model" ovvero del modello impostato per la gestione dei thread per i server DCOM. Questo modello si può impostare una sola volta nell'applicazione e non può essere cambiato.

Le applicazioni per Win32 (PC) normalmente inizializzano OLE e di conseguenza DCOM come single-thread.

P.e. le applicazioni C++ che usano MFC inizializzano DCOM tramite la funzione di inizializzazione di OLE (vedi chiamata a funzione "AfxOleInit" nella classe derivata da "CWinApp").

Le applicazioni C# inizializzano il DCOM threading model a singler-thread con l'istruzione [[STAThread](#)] nella classe "Program".

Le applicazioni C# sviluppate per Windows CE inizializzano invece il DCOM threading model a multi-thread e non è disponibile il modello single-thread.

Sostanzialmente la funzione `OpenSession_C` inizializza il DCOM threading model a single-thread solo nel caso il modello sia ancora da definire. Negli altri casi utilizza il modello impostato dal chiamante.

Se l'applicazione deve collegare lo stesso CNC in più thread si può quindi fare in due modi:

- 1) Se l'applicazione lo permette, inizializzare il "DCOM threading model" come "multi threaded" ed utilizzare la sessione di comunicazione aperta con la funzione `OpenSession_C` in tutti i thread.
- 2) Se l'applicazione deve inizializzare OLE oppure il "DCOM threading model" come "apartment (single) threaded", aprire diverse sessioni di comunicazione indipendenti chiamando la funzione `OpenSession_C` in ogni thread che deve connettersi al CNC.

Vedi anche

`CloseSession_C`



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

CloseSession_C

Chiude la sessione di comunicazione con un controllo numerico e rilascia l'istanza del server Cndex creata con la funzione OpenSession_C.

All'uscita da questa funzione l'identificativo di sessione non e' piu' valido e non deve piu' essere utilizzato.

```
WORD CloseSession_C (  
    WORD        UserSession,  
    DWORD       *pErrClass,  
    DWORD       *pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Note

Se la sessione di comunicazione è utilizzata da più thread (vedi note della funzione OpenSession_C) assicurarsi che tutti i thread abbiano abbandonato (non utilizzino più) la sessione prima di chiuderla.

Vedi anche

OpenSession_C

Funzioni per il controllo della fase di bootstrap del CNC

Il CNC durante l'accensione passa attraverso diverse "fasi" che si possono controllare tramite la funzione **BootPhaseEnquiry_C**.

Le prime fasi (fase 1 e fase 2) sono eseguite da un ambiente di boot che attiva il Sistema Operativo OSAI multi-tasking e tutti i task che costituiscono il CNC. Quando il Sistema Operativo viene attivato, distrugge l'ambiente di boot che e' stato utilizzato per l'accensione.

L'ambiente di boot che esegue le fasi 1 e 2 predispone il CNC alla connessione in rete.

A causa di questo meccanismo di funzionamento, risulterà possibile aprire una sessione di comunicazione con il CNC quando e' in esecuzione l'ambiente di boot ma questa connessione cade quando viene attivato il Sistema Operativo.

Dopo l'avvio del S.O. sara' nuovamente possibile connettere il CNC in modo definitivo.



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

Il CNC, quando e' attivo l'ambiente di boot, può essere pilotato in modalità operative dedicate ad operazioni di configurazione e aggiornamento del software.

Una applicazione client deve prevedere questa situazione ed attendere la modalità RUN (fase 4) prima di eseguire le funzioni di richiesta e invio dati al CNC.

Le possibili fasi di accensione del CNC sono (il numero della fase corrisponde al numero restituito nel parametro pPhase della funzione **BootPhaseEnquiry**) :

- 1 = EMERG_SWITCH_PHASE
- 2 = HW_BOOT_PHASE
- 3 = SW_BOOT_PHASE
- 4 = SYSTEM_UP_PHASE
- 5 = SERVER_MODE_PHASE
- 6 = REMOTE_SETUP_PHASE
- 7 = SERVICE_MODE_PHASE
- 8 = AX_PARAM_VERIFY_PHASE
- 15 = PRIMARY_INIT_PHASE
- 16 = NOT_INIT_PHASE
- 17 = SHUTDOWN_PHASE
- 19 = ERROR_PHASE

- Fase 1** Il CNC e' in attesa del comando che lo piloti in uno degli stati successivi. Questa e' una fase di transizione. Se dopo un certo tempo nessun client si e' connesso, il CNC continua la normale procedura di BOOT per portarsi nello stato RUN.
- Fase 2** IL CNC sta eseguendo i programmi di diagnostica e configurazione dell'hardware. E' una fase di transizione.
- Fase 3** IL CNC sta eseguendo il caricamento del software del CNC per portarsi nel modo RUN (fase 4). E' una fase di transizione
- Fase 4** Il CNC e' nello stato RUN e puo' iniziare la lavorazione. Le applicazioni client possono chiamare le funzioni di scambio dati. Questa e' la fase stabile di lavorazione pezzo, per portarsi in un'altra modalità operativa il CNC deve rieseguire il bootstrap.
- Fase 5** Il CNC e' nello stato di EMERGENZA. E' una fase "stabile", per portarsi in un'altra modalità operativa (p.e. RUN) il CNC deve rieseguire il bootstrap.
- Fase 6** Il CNC e' nello stato di SETUP. E' una fase "stabile", per portarsi in un'altra modalità operativa (p.e. RUN) il CNC deve rieseguire il bootstrap.
- Fase 7** Il CNC e' nello stato di SERVICE. E' una fase "stabile", per portarsi in un'altra modalità operativa (p.e. RUN) il CNC deve rieseguire il bootstrap.
- Fase 8** Il CNC e' nello stato di VERIFICA PARAMETRI. Non e' una fase stabile. Su consenso dell'utente del CNC il CNC si porterà in fase 4.
- Fase 15** Il CNC è in corso di avvio ma non ha ancora raggiunto la fase 1.
- Fase 16** Il CNC non è inizializzato.
- Fase 17** Il CNC è in fase di reboot.
- Fase 19** L'avvio del CNC si è interrotto a causa di un errore non recuperabile.



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

BootPhaseEnquiry_C

Richiede la fase di accensione del controllo numerico.

```
WORD BootPhaseEnquiry_C (  
    WORD        UserSession,  
    WORD        *pPhase,  
    DWORD       *pErrClass,  
    DWORD       *pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

pPhase

[out] Puntatore ad una variabile dove verrà scritta la fase di boot del CNC (vedi la descrizione a inizio paragrafo).

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione è stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Vedi anche

BootReboot_C

Descrizione a inizio paragrafo.



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

BootReboot_C

Provoca il reset hardware del controllo numerico al quale segue il reboot. L'attivazione di questa funzione corrisponde all'operazione di spegnimento e riaccensione del CNC.

Questo comando e' disponibile in tutte le fasi di accensione.

Per i CNC OPENControl viene eseguito un reboot software della sola sezione CNC della macchina a meno che il CNC si trovi nella modalit  di boot SETUP. In questo caso anche OPENControl esegue un reboot hardware.

```
WORD BootReboot_C (  
    WORD        UserSession,  
    DWORD       *pErrClass,  
    DWORD       *pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

pErrClass

[out] Puntatore alla variabile dove verr  scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verr  scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Vedi anche

BootShutDown_C, BootPhaseEnquiry_C

Descrizione a inizio paragrafo.



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

BootShutDown_C

Richiede al CNC di eseguire la procedura di spegnimento controllato (chiusura dei file aperti e spegnimento).

```
WORD BootShutDown_C (  
    WORD        UserSession,  
    DWORD       *pErrClass,  
    DWORD       *pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione è stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Vedi anche

BootReboot_C

Descrizione a inizio paragrafo.



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

BootMode_C

Permette di scegliere la modalità di lavoro del CNC durante la fase di accensione del CNC stesso. Si possono richiedere le modalità di lavoro EMERGENZA, RUN (accensione normale), SETUP o SERVICE.

Per i CNC Serie10 questo comando è disponibile solamente quando la fase di accensione del CNC è EMERG_SWITCH_PHASE (vedi BootPhaseEnquiry_C). In tutte le altre fasi di accensione il comando non è valido e può provocare malfunzionamenti del CNC.

Per i CNC OPENControl il comando è invece sempre disponibile e deve essere inviato prima di richiedere la riaccensione della macchina tramite la funzione BootReboot_C.

```
WORD BootMode_C (  
    WORD      UserSession,  
    BootMode  Mode,  
    DWORD    *pErrClass,  
    DWORD    *pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

Mode

[in] Modalità di accensione del CNC richiesta. I possibili valori sono:

BOOT_EMERGENCY_MODE: Richiede al CNC di continuare la fase di accensione e portarsi nella modalità EMERGENZA.

BOOT_RUN_MODE: Richiede al CNC di continuare la fase di accensione e portarsi nella modalità RUN (modalità di lavoro normale).

BOOT_SETUP_MODE: Richiede al CNC di continuare la fase di accensione e portarsi nella modalità SETUP (modalità di installazione del software).

BOOT_SERVICE_MODE: Richiede al CNC di continuare la fase di accensione e portarsi nella modalità SERVICE (Modalità di taratura degli azionamenti OS-Wire).

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione è stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Vedi anche

BootReboot_C, BootPhaseEnquiry_C

Descrizione a inizio paragrafo.



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

BootEnterServer_C (solo per Serie 10)

Attenzione: l'utilizzo di questa funzione è riservato agli utenti esperti e può avvenire solamente dietro esplicita autorizzazione da parte di Prima Electro. L'uso improprio può causare la perdita di dati e provocare danni al file system del CNC.

Questa funzione può essere utilizzata solamente con sistemi Serie 10 in modalità SETUP. Un sistema Serie 10 avviato in modalità SETUP carica automaticamente il server delle chiamate a procedura remota alla fine del bootstrap. Questo server gestisce, tra le altre cose, le chiamate a funzione di Cndex.

Dopo una chiamata a BootEnterServer_C il CNC chiude il server delle chiamate a procedura remota e lancia il file server, permettendo ad un sistema remoto di accedere alle directory condivise PRJ, UPR e UPP. La directory condivisa SYS non è disponibile in modalità SETUP. La maggior parte delle chiamate a funzione di Cndex non è disponibile quando è attivo il file server.

È possibile riattivare il server delle chiamate a procedura remota in qualsiasi momento chiamando BootLeaveServer_C. Quest'operazione causa la chiusura asincrona del file server e può causare la perdita di dati se eseguita mentre è in corso un accesso remoto ad una directory condivisa del CNC.

```
WORD BootEnterServer_C (  
    WORD        UserSession,  
    DWORD       *pErrClass,  
    DWORD       *pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione è stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Vedi anche

BootLeaveServer_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

BootLeaveServer_C (solo per Serie 10)

Attenzione: l'utilizzo di questa funzione è riservato agli utenti esperti e può avvenire solamente dietro esplicita autorizzazione da parte di Prima Electro. L'uso improprio può causare la perdita di dati e provocare danni al file system del CNC.

Questa funzione può essere utilizzata solamente con sistemi Serie 10 in modalità SETUP. La chiamata a funzione richiede al CNC di chiudere il file server e attivare il server delle chiamate a procedura remota, permettendo l'esecuzione delle chiamate a funzione di Cndex.

```
WORD BootLeaveServer_C (  
    WORD    UserSession,  
    DWORD   *pErrClass,  
    DWORD   *pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione è stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Vedi anche

BootEnterServer_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

GetHWKey_C

Legge la Product Authorization Key (PAK), la maschera dei flag di security e la release software dal CNC.

```
WORD GetHWKey_C (  
    WORD        UserSession,  
    BYTE        *pHwKey,  
    BYTE        *pSecurLevel,  
    BYTE        *pRelease,  
    DWORD       *pErrClass,  
    DWORD       *pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

pHwKey

[out] Puntatore ad un buffer di 32 byte dove verrà scritta la PAK.

pSecurLevel

[out] Puntatore ad un buffer di 12 byte dove verrà scritta la maschera dei flag di security.

pRelease

[out] Puntatore ad un buffer di 11 byte dove verrà scritto il numero di release software del CNC in formato stringa.

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione è stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

Gestione variabili

Le funzioni di questo gruppo permettono di leggere e scrivere molte delle variabili del CNC. Queste funzioni possono essere chiamate solo dopo aver creato il server Cndex (vedi `ConnectServer_C`), aperto una sessione di comunicazione (vedi `OpenSession_C`) ed aver verificato che il CNC si trovi nella modalità RUN (vedi `BootPhaseEnquiry_C`).



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

ReadVarWord_C

Legge le variabili del sistema che hanno formato WORD (16 bit).

```
WORD ReadVarWord_C (  
    WORD    UserSession,  
    WORD    Code,  
    WORD    Process,  
    WORD    Index,  
    WORD    NumVar,  
    WORD    *pValue  
    DWORD   *pErrClass,  
    DWORD   *pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

Code

[in] Codice identificativo della variabile. I possibili valori sono riportati nella prima (define) e seconda (valore) colonna della tabella che segue. Le altre colonne della tabella riportano, per ogni tipo di variabile, il numero massimo di elementi disponibili per ogni sistema OSAI. Il simbolo (R) indica le variabili ritentive.

<i>define</i>	<i>valore</i>	<i>Serie 10 MC</i>	<i>OPENControl</i>	<i>PrimaLogic</i>	<i>SyncMotion</i>
I_CODE	0	512	512 (fino a V3.3.1) 4096 (da V3.3.2)	512	0
O_CODE	1	512	512 (fino a V3.3.1) 4096 (da V3.3.2)	512	0
MW_CODE	20	10000	20000	20000	0
GW_CODE (R)	21	256	2000	2000	0
SW_CODE	22	20 + 20 * numero processi	20 + 20 * numero processi	20	0
PW_CODE	62	0	6000 (fino a V3.3.1) 20000 (da V3.3.2)	0	0
UW_CODE	63	0	0	0	0
SYMO_W_CODE	100	0	20000	0	8192
SYMORET_W_CODE (R)	103	0	2000	0	8192

Process

[in] Identificativo (numero da 1 a 20 per Serie MC o numero da 1 a 24 per OPENControl) del processo al quale appartiene la variabile.

E' preso in considerazione solo se il tipo di variabile indicata con il parametro Code appartiene ad un processo. Per gli altri tipi di variabile può valere zero.

Index

[in] indice della prima variabile da leggere

NumVar



Documentazione CndexLinkUser DLL

Interfaccia al server Cndex

[in] Numero di variabili da leggere. Questo numero sommato all'indice Index non deve superare il massimo numero di variabili ammesso per il tipo (Code) selezionato.

pValue

[out] Puntatore ad un array di variabili di formato WORD (16 bit) che conterranno i valori delle variabili lette. L'array deve comprendere come minimo un numero di elementi pari a NumVar.

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

ReadVarDouble_C

Legge le variabili del sistema che hanno formato double (floating point 64 bit).

```
WORD ReadVarDouble_C (  
    WORD    UserSession,  
    WORD    Code,  
    WORD    Process,  
    WORD    Index,  
    WORD    NumVar,  
    Double  *pValue  
    DWORD  *pErrClass,  
    DWORD  *pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

Code

[in] Codice identificativo della variabile. I possibili valori sono riportati nella prima (define) e seconda (valore) colonna della tabella che segue. Le altre colonne della tabella riportano, per ogni tipo di variabile, il numero massimo di elementi disponibili per ogni sistema OSAI. Il simbolo (R) indica le variabili ritentive.

<i>define</i>	<i>valore</i>	<i>Serie 10 MC</i>	<i>OPENControl</i>	<i>PrimaLogic</i>	<i>SyncMotion</i>
MD_CODE	40	3000	10000	10000	0
GD_CODE (R)	41	64	1000	1000	0
SD_CODE	42	0	1000	20	0
PD_CODE	43	0	6000 (fino a V3.3.1) 20000 (da V3.3.2)	20	0
UD_CODE	44	0	0	0	0
L_CODE (R)	145	0	3000	0	0
E_CODE	46	Da 0 a 8000 (configurate in AMP)	Da 0 a 8000 (configurate in AMP)	0	0
SN_CODE	47	25	25	0	0
H_CODE	48	100	128	0	0
SYMO_D_CODE	101	0	10000	0	6143
SYMORET_D_CODE (R)	104	0	3000	0	5117

Process

[in] Identificativo (numero da 1 a 20 per Serie MC o numero da 1 a 24 per OPENControl) del processo al quale appartiene la variabile.

E' preso in considerazione solo se il tipo di variabile indicata con il parametro Code appartiene ad un processo. Per gli altri tipi di variabile può valere 0.

Index

[in] indice della prima variabile da leggere

NumVar



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

[in] Numero di variabili da leggere. Questo numero sommato all'indice Index non deve superare il massimo numero di variabili ammesso per il tipo (Code) selezionato.

pValue

[out] Puntatore ad un array di variabili di formato double che conterranno i valori delle variabili lette. L'array deve comprendere come minimo un numero di elementi pari a NumVar.

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

WriteVarWord_C

Scrive le variabili del sistema che hanno formato WORD (16 bit).

```
WORD WriteVarWord_C (  
    WORD    UserSession,  
    WORD    Code,  
    WORD    Process,  
    WORD    Index,  
    WORD    NumVar,  
    WORD    *pValue  
    DWORD   *pErrClass,  
    DWORD   *pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

Code

[in] Codice identificativo della variabile. Per i possibili valori si veda la funzione ReadVarWord_C.

Process

[in] Identificativo (numero da 1 a 20 per Serie MC o numero da 1 a 24 per OPENControl) del processo al quale appartiene la variabile.

E' preso in considerazione solo se il tipo di variabile indicata con il parametro Code appartiene ad un processo. Per gli altri tipi di variabile può valere 0.

Index

[in] indice della prima variabile da scrivere

NumVar

[in] Numero di variabili da scrivere. Questo numero sommato all'indice Index non deve superare il massimo numero di variabili ammesso per il tipo (Code) selezionato.

pValue

[in] Puntatore ad un array di variabili di formato WORD (16 bit). L'array deve comprendere come minimo un numero di elementi pari a NumVar e contenere i valori da assegnare alle variabili.

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".



Documentazione CndexLinkUser DLL Interfaccia al server Cndex



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

WriteVarWordBit_C

Scrive un singolo bit di una variabili del sistema che ha formato WORD (16 bit).

```
WORD WriteVarWordBit_C (  
    WORD    UserSession,  
    WORD    Code,  
    WORD    Process,  
    WORD    Index,  
    WORD    BitIndex,  
    WORD    BitValue,  
    DWORD   *pErrClass,  
    DWORD   *pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

Code

[in] Codice identificativo della variabile. Per i possibili valori si veda la funzione ReadVarWord_C.

Process

[in] Identificativo (numero da 1 a 20 per Serie MC o numero da 1 a 24 per OPENControl) del processo al quale appartiene la variabile. E' preso in considerazione solo se il tipo di variabile indicata con il parametro Code appartiene ad un processo. Per gli altri tipi di variabile può valere 0.

Index

[in] indice della variabile da scrivere

BitIndex

[in] indice, all'interno della variabile, del bit da scrivere.

BitValue

[in] Valore da assegnare al bit, zero o uno.

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

WriteVarDouble_C

Scrive le variabili del sistema che hanno double (floating point 64 bit).

```
WORD WriteVarDouble_C (  
    WORD        UserSession,  
    WORD        Code,  
    WORD        Process,  
    WORD        Index,  
    WORD        NumVar,  
    double *    pValue  
    DWORD *    pErrClass,  
    DWORD *    pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

Code

[in] Codice identificativo della variabile. Per i possibili valori si veda la funzione ReadVarDouble_C.

Process

[in] Identificativo (numero da 1 a 20) del processo al quale appartiene la variabile.

E' preso in considerazione solo se il tipo di variabile indicata con il parametro Code appartiene ad un processo. Per gli altri tipi di variabile può valere 0.

Index

[in] indice della prima variabile da scrivere

NumVar

[in] Numero di variabili da scrivere. Questo numero sommato all'indice Index non deve superare il massimo numero di variabili ammesso per il tipo (Code) selezionato.

pValue

[in] Puntatore ad un array di variabili di formato double. L'array deve comprendere come minimo un numero di elementi pari a NumVar e contenere i valori da assegnare alle variabili.

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

ReadVarText_C

Legge le variabili ASCII (testo) del sistema.

```
WORD ReadVarText_C (  
    WORD        UserSession,  
    WORD        Code,  
    WORD        Process,  
    WORD        Index,  
    WORD        Size,  
    BYTE *      Text  
    DWORD *     pErrClass,  
    DWORD *     pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

Code

[in] Codice identificativo della variabile. I possibili valori sono riportati nella prima (define) e seconda (valore) colonna della tabella che segue. Le altre colonne della tabella riportano, per ogni tipo di variabile, il numero massimo di elementi disponibili per ogni sistema OSAI x la lunghezza massima della stringa.

<i>define</i>	<i>valore</i>	<i>Serie 10 MC</i>	<i>OPENControl</i>	<i>PrimaLogic</i>	<i>SyncMotion</i>
A_CODE	45	WinPLUS: 1500 x 40 chr PLUS: non disponibile	1500 x 40 chr	250 x 128 chr	non disponibile
AA_CODE	28	non disponibile	250 x 128 chr	250 x 128 chr	non disponibile
SC_CODE	50	100	100	non disponibile	non disponibile
SYMO_A_CODE	102	non disponibile	250 x 128 chr	non disponibile	64 x 128 chr
LS_CODE	18	non disponibile	100 x 128 chr	250 x 128 chr	non disponibile

Process

[in] Identificativo (numero da 1 a 20) del processo al quale appartiene la variabile.

E' preso in considerazione solo se il tipo di variabile indicata con il parametro Code appartiene ad un processo. Per gli altri tipi di variabile può valere 0.

Index

[in] indice della prima variabile da leggere

Size

[in] Dimensione del buffer che riceverà i dati (vedi parametro Text).

Text

[out] Puntatore ad un array di variabili di formato BYTE che conterranno il testo letto nelle variabili. L'array deve comprendere come minimo un numero di elementi pari a Size e dovrebbe poter contenere il numero di caratteri (tipicamente 128) previsto per le stringhe del tipo selezionato.



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

WriteVarText_C

Scrive le variabili ASCII (testo) del sistema.

```
WORD WriteVarText_C (  
    WORD        UserSession,  
    WORD        Code,  
    WORD        Process,  
    WORD        Index,  
    WORD        Len,  
    BYTE *      Text,  
    DWORD *     pErrClass,  
    DWORD *     pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

Code

[in] Codice identificativo della variabile. Per i possibili valori si veda la funzione ReadVarText_C.

Process

[in] Identificativo (numero da 1 a 20) del processo al quale appartiene la variabile.
E' preso in considerazione solo se il tipo di variabile indicata con il parametro Code appartiene ad un processo. Per gli altri tipi di variabile può valere 0.

Index

[in] indice della prima variabile da leggere

Len

[in] Dimensione del buffer da scrivere (vedi parametro Text).

Text

[out] Puntatore ad un array di variabili di formato BYTE che contiene il testo da scrivere nelle variabili. L'array deve comprendere come minimo un numero di elementi pari a Len.

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.
Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

Funzioni dedicate al PLC

Queste funzioni permettono di scambiare dati con il PLC e di gestire le tabelle Assi/Utensili/Offset/Utente sincronizzandone l'accesso tramite semafori.

Queste funzioni possono essere chiamate solo dopo aver creato il server Cndex (vedi ConnectServer_C), aperto una sessione di comunicazione (vedi OpenSession_C) ed aver verificato che il CNC si trovi nella modalita' RUN (vedi BootPhaseEnquiry_C).

ATTENZIONE:

- 1) *Le funzioni per la lettura e scrittura di variabili di questo paragrafo si possono utilizzare ma sono obsolete. Al loro posto si può usare la nuova serie di funzioni descritte nel paragrafo "Gestione variabili".*
- 2) *Le funzioni dedicate alle tabelle (Utensili, Correttori, Origine e Utente) sono differenti per i sistemi Serie10 e OPENControl (vedi le tabelle delle compatibilità all'inizio di questo documento).*



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

ReadWarningMsg_C

Legge il messaggio che la logica di macchina ha richiesto di visualizzare.

```
WORD ReadWarningMsg_C (  
    WORD          UserSession,  
    LPSTR         pWarningMsg,  
    DWORD *       pErrClass,  
    DWORD *       pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

pWarningMsg

[out] Puntatore ad una stringa di 130 caratteri in cui verrà scritto il messaggio da PLC con al fondo il terminatore 0 (zero).

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione è stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

GetPLVarWord_C (funzione obsoleta)

Funzione obsoleta. E' disponibile in questa libreria per compatibilità con applicazioni precedenti. Al suo posto si può usare la funzione ReadVarWord_C.

Legge il valore delle variabili short del PLUS (MW, GW etc.)

```
WORD GetPLVarWord_C (  
    WORD UserSession,  
    WORD NumVar,  
    PLVARDESC_C4 * pVarDesc,  
    short * pValue,  
    DWORD * pErrClass,  
    DWORD * pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

NumVar

[in] Numero di variabili di cui si vuole leggere il valore

pVarDesc

[in] Puntatore ad un array di strutture di tipo PLVARDESC_C4 che specifica le variabili delle quali si vogliono ottenere i valori

pValue

[out] Puntatore ad un array di short che conterra' i valori delle variabili specificate da pVarDesc

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Ulteriori codici di errore:

Classe 4 – Errore 1026 Il numero di variabili NumVar è fuori range.

Classe 4 – Errore 1027 Uno dei codici identificativi delle variabili indicate nella struttura è sbagliato.

Classe 4 – Errore 1032 Uno degli indici delle variabili indicate nella struttura è fuori range.

Classe 4 – Errore 1033 Una delle maschere delle variabili indicate nella struttura è fuori range.

Vedi anche

ReadVarWord_C

SetPLVarWord_C

Paragrafo "Descrizione strutture" per la definizione di PLVARDESC_C4



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

SetPLVarWord_C (funzione obsoleta)

Funzione obsoleta. E' disponibile in questa libreria per compatibilità con applicazioni precedenti. Al suo posto si può usare la funzione WriteVarWord_C.

Scrive il valore delle variabili word e boolean del PLUS (MW, GW, M, G etc.)

```
WORD SetPLVarWord_C (  
    WORD                UserSession,  
    WORD                NumVar,  
    PLVARDESC_C4 * pVarDesc,  
    short *            pValue,  
    DWORD *            pErrClass,  
    DWORD *            pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

NumVar

[in] Numero di variabili di cui si vuole scrivere il valore

pVarDesc

[in] Puntatore ad un array di strutture di tipo PLVARDESC_C4 che specifica le variabili da modificare

pValue

[in] Puntatore ad un array di short che contiene i valori da impostare nelle variabili descritte da pVarDesc

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Ulteriori codici di errore:

Classe 4 – Errore 1026 Il numero di variabili NumVar è fuori range.

Classe 4 – Errore 1027 Uno dei codici identificativi delle variabili indicate nella struttura è sbagliato.

Classe 4 – Errore 1032 Uno degli indici delle variabili indicate nella struttura è fuori range.

Classe 4 – Errore 1033 Una delle maschere delle variabili indicate nella struttura è fuori range.

Vedi anche

WriteVarWord_C

GetPLVarWord_C

Paragrafo "Descrizione strutture" per la definizione di PLVARDESC_C4



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

GetPLVarDouble_C (funzione obsoleta)

Funzione obsoleta. E' disponibile in questa libreria per compatibilità con applicazioni precedenti. Al suo posto si può usare la funzione ReadVarDouble_C.

Legge il valore delle variabili double del PLUS (MD, GD etc.)

```
WORD GetPLVarDouble_C (  
    WORD UserSession,  
    WORD NumVar,  
    PLVARDESC_C4 * pVarDesc,  
    double * pValue,  
    DWORD * pErrClass,  
    DWORD * pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

NumVar

[in] Numero di variabili di cui si vuole leggere il valore

pVarDesc

[in] Puntatore ad un array di strutture di tipo PLVARDESC_C4 che specifica le variabili delle quali si vogliono ottenere i valori

pValue

[out] Puntatore ad un array di double che conterra' i valori delle variabili specificate da pVarDesc

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Ulteriori codici di errore:

Classe 4 – Errore 1026 Il numero di variabili NumVar è fuori range.

Classe 4 – Errore 1027 Uno dei codici identificativi delle variabili indicate nella struttura è sbagliato.

Classe 4 – Errore 1032 Uno degli indici delle variabili indicate nella struttura è fuori range.

Classe 4 – Errore 1033 Una delle maschere delle variabili indicate nella struttura è fuori range.

Vedi anche

ReadVarDouble_C

SetPLVarDouble_C

Paragrafo "Descrizione strutture" per la definizione di PLVARDESC_C4



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

SetPLVarDouble_C (funzione obsoleta)

Funzione obsoleta. E' disponibile in questa libreria per compatibilità con applicazioni precedenti. Al suo posto si può usare la funzione WriteVarDouble_C.

Scriva il valore delle variabili double del PLUS (MD, GD etc.)

```
WORD SetPLVarDouble_C (  
    WORD UserSession,  
    WORD NumVar,  
    PLVARDESC_C4 * pVarDesc,  
    double * pValue,  
    DWORD * pErrClass,  
    DWORD * pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

NumVar

[in] Numero di variabili di cui si vuole scrivere il valore

pVarDesc

[in] Puntatore ad un array di strutture di tipo PLVARDESC_C4 che specifica le variabili da modificare

pValue

[in] Puntatore ad un array di double che contiene i valori da impostare nelle variabili descritte da pVarDesc

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Ulteriori codici di errore:

Classe 4 – Errore 1026 Il numero di variabili NumVar è fuori range.

Classe 4 – Errore 1027 Uno dei codici identificativi delle variabili indicate nella struttura è sbagliato.

Classe 4 – Errore 1032 Uno degli indici delle variabili indicate nella struttura è fuori range.

Classe 4 – Errore 1033 Una delle maschere delle variabili indicate nella struttura è fuori range.

Vedi anche

WriteVarDouble_C

GetPLVarDouble_C

Paragrafo "Descrizione strutture" per la definizione di PLVARDESC_C4



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

GetPLVarAscii_C (funzione obsoleta)

Funzione obsoleta. E' disponibile in questa libreria per compatibilità con applicazioni precedenti. Al suo posto si può usare la funzione *ReadVarText_C*.

Legge l'array delle variabili ASCII del PLC [WinPLUS](#).

Questa funzione e' disponibile solo se la logica di macchina e' stata scritta con WinPLUS e se la variabile e' definita.

```
WORD GetPLVarAscii_C (  
    WORD          UserSession,  
    WORD          Index,  
    LPSTR         pValue,  
    DWORD *       pErrClass,  
    DWORD *       pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

Index

[in] Indice di partenza per la lettura delle variabili ASCII

pValue

[in] Puntatore ad una stringa 41 caratteri in cui verra' scritto il testo contenuto nelle variabili ASCII. Al fondo del testo e' presente il terminatore di stringa /0 (zero).

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Vedi anche

[ReadVarText_C](#)

[SetPLVarAscii_C](#)



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

SetPLVarAscii_C (funzione obsoleta)

Funzione obsoleta. E' disponibile in questa libreria per compatibilità con applicazioni precedenti. Al suo posto si può usare la funzione *WriteVarText_C*.

Scrive nell'array delle variabili ASCII del PLC WinPLUS.

Questa funzione e' disponibile solo se la logica di macchina e' stata scritta con WinPLUS e se la variabile e' definita.

```
WORD SetPLVarAscii_C (  
    WORD          UserSession,  
    WORD          Index,  
    LPSTR         pValue,  
    DWORD *       pErrClass,  
    DWORD *       pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

Index

[in] Indice di partenza per la scrittura delle variabili ASCII

pValue

[in] Puntatore ad una stringa 41 caratteri, terminata con /0 (zero), che contiene il testo da scrivere nelle variabili ASCII. Al fondo del testo e' presente il terminatore di stringa /0 (zero).

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Vedi anche

WriteVarText_C

GetPLVarAscii_C, SetTabSem_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

GetPLStreamWord_C (funzione obsoleta)

Funzione obsoleta. E' disponibile in questa libreria per compatibilità con applicazioni precedenti. Al suo posto si può usare la funzione ReadVarWord_C.

Legge array di variabili sequenziali di tipo short del PLC (MW, GW etc.)

```
WORD GetPLStreamWord_C (  
    WORD      UserSession,  
    WORD      NumVar,  
    WORD      Code,  
    WORD      StartIndex,  
    short *    pValue,  
    DWORD *    pErrClass,  
    DWORD *    pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

NumVar

[in] Numero di variabili di cui si vuole leggere il valore

Code

[in] Tipo di variabili da leggere. Vedi campo Code della struttura PLVARDESC_C4 nel paragrafo "Definizioni delle Strutture e Definizioni".

StartIndex

[in] Indice di partenza della lettura all'interno dell'array.

pValue

[out] Puntatore ad un array di short che conterrà i valori delle variabili richieste. L'array deve essere dimensionato per contenere tutte le variabili richieste (almeno NumVar elementi).

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione è stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Ulteriori codici di errore:

Classe 4 – Errore 1027 Codice identificativo dell'array di variabili sbagliato o non disponibile sul sistema connesso.

Classe 4 – Errore 1032 Indice fuori range (StartIndex + NumVar).



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

Vedi anche

[ReadVarWord_C](#)

[SetPLVarWord_C](#)

[Paragrafo "Descrizione strutture" per la definizione del campo Code della struttura PLVARDESC_C4](#)



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

SetPLStreamWord_C (funzione obsoleta)

Funzione obsoleta. E' disponibile in questa libreria per compatibilità con applicazioni precedenti. Al suo posto si può usare la funzione WriteVarWord_C.

Scrive array di variabili sequenziali di tipo short del PLC (MW, GW etc.)

```
WORD SetPLStreamWord_C (  
    WORD      UserSession,  
    WORD      NumVar,  
    WORD      Code,  
    WORD      StartIndex,  
    short *    pValue,  
    DWORD *    pErrClass,  
    DWORD *    pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

NumVar

[in] Numero di variabili da scrivere

Code

[in] Tipo di variabili da scrivere. Vedi campo Code della struttura PLVARDESC_C4 nel paragrafo "Definizioni delle Strutture e Definizioni".

StartIndex

[in] Indice di partenza della scrittura all'interno dell'array.

pValue

[out] Puntatore ad un array di short che contiene i valori da scrivere nelle variabili. L'array deve essere dimensionato per contenere tutti i valori da scrivere (almeno NumVar elementi).

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Ulteriori codici di errore:

Classe 4 – Errore 1027 Codice identificativo dell'array di variabili sbagliato o non disponibile sul sistema connesso.

Classe 4 – Errore 1032 Indice fuori range (StartIndex + NumVar).



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

Vedi anche

WriteVarWord_C

SetPLVarWord_C

Paragrafo "Descrizione strutture" per la definizione del campo Code della struttura PLVARDESC_C4



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

GetPLStreamDouble_C (funzione obsoleta)

Funzione obsoleta. E' disponibile in questa libreria per compatibilità con applicazioni precedenti. Al suo posto si può usare la funzione ReadVarDouble_C.

Legge array di variabili sequenziali di tipo double del PLC (MD, GD etc.)

```
WORD GetPLStreamWord_C (  
    WORD      UserSession,  
    WORD      NumVar,  
    WORD      Code,  
    WORD      StartIndex,  
    double *   pValue,  
    DWORD *   pErrClass,  
    DWORD *   pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

NumVar

[in] Numero di variabili di cui si vuole leggere il valore

Code

[in] Tipo di variabili da leggere. Vedi campo Code della struttura PLVARDESC_C4 nel paragrafo "Definizioni delle Strutture e Definizioni".

StartIndex

[in] Indice di partenza della lettura all'interno dell'array.

pValue

[out] Puntatore ad un array di double che conterra' i valori delle variabili richieste. L'array deve essere dimensionato per contenere tutte le variabili richieste (almeno NumVar elementi).

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Ulteriori codici di errore:

Classe 4 – Errore 1027 Codice identificativo dell'array di variabili sbagliato o non disponibile sul sistema connesso.

Classe 4 – Errore 1032 Indice fuori range (StartIndex + NumVar).



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

Vedi anche

[ReadVarDouble_C](#)

[SetPLVarDouble_C](#)

[Paragrafo "Descrizione strutture" per la definizione del campo Code della struttura PLVARDESC_C4](#)



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

SetPLStreamDouble_C (funzione obsoleta)

Funzione obsoleta. E' disponibile in questa libreria per compatibilità con applicazioni precedenti. Al suo posto si può usare la funzione WriteVarDouble_C.

Scrive array di variabili sequenziali di tipo double del PLC (MD, GD etc.)

```
WORD GetPLStreamWord_C (  
    WORD      UserSession,  
    WORD      NumVar,  
    WORD      Code,  
    WORD      StartIndex,  
    double *   pValue,  
    DWORD *   pErrClass,  
    DWORD *   pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

NumVar

[in] Numero di variabili da scrivere

Code

[in] Tipo di variabili da scrivere. Vedi campo Code della struttura PLVARDESC_C4 nel paragrafo "Definizioni delle Strutture e Definizioni".

StartIndex

[in] Indice di partenza della scrittura all'interno dell'array.

pValue

[out] Puntatore ad un array di double che contiene i valori da scrivere nelle variabili. L'array deve essere dimensionato per contenere tutti i valori da scrivere (almeno NumVar elementi).

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Ulteriori codici di errore:

Classe 4 – Errore 1027 Codice identificativo dell'array di variabili sbagliato o non disponibile sul sistema connesso.

Classe 4 – Errore 1032 Indice fuori range (StartIndex + NumVar).



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

Vedi anche

WriteVarDouble_C

SetPLVarDouble_C

Paragrafo "Descrizione strutture" per la definizione del campo Code della struttura PLVARDESC_C4



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

GetAxisTabRecord_C

Legge un record della tabella assi (origini) del sistema.

```
WORD GetAxisTabRecord_C (  
    WORD                UserSession,  
    WORD                RecordNum,  
    AXIS_TABLE_C4 * pRecordBuff ,  
    DWORD *            pErrClass,  
    DWORD *            pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

pRecordBuff

[out] Puntatore ad una struttura di tipo AXIS_TABLE_C4 dove verrà scritto il record richiesto

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Ulteriori codici di errore:

Classe 4 – Errore 4 Il numero di record RecordNum è fuori range.

Vedi anche

SetAxisTabRecord_C, SetTabSem_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

SetAxisTabRecord_C

Scrive un record della tabella assi (origini) del sistema.

```
WORD SetAxisTabRecord_C (  
    WORD                UserSession,  
    WORD                RecordNum,  
    AXIS_TABLE_C4 * pRecordBuff,  
    DWORD *             pErrClass,  
    DWORD *             pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

pRecordBuff

[in] Puntatore alla struttura di tipo AXIS_TABLE_C4 che contiene i dati da scrivere nel record

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Ulteriori codici di errore:

Classe 4 – Errore 4 Il numero di record RecordNum è fuori range.

Vedi anche

GetAxisTabRecord_C, SetTabSem_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

GetToolTabRecord_C

Legge un record della tabella utensili del sistema.

```
WORD GetToolTabRecord_C (  
    WORD                UserSession,  
    WORD                RecordNum,  
    TOOL_TABLE_C4 *pRecordBuff ,  
    DWORD *             pErrClass,  
    DWORD *             pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

pRecordBuff

[out] Puntatore ad una struttura di tipo TOOL_TABLE_C4 dove verra' scritto il record richiesto

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Ulteriori codici di errore:

Classe 4 – Errore 4 Il numero di record RecordNum è fuori range.

Vedi anche

SetToolTabRecord_C, SetTabSem_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

SetToolTabRecord_C

Scrive un record della tabella utensili del sistema.

```
WORD SetToolTabRecord_C (  
    WORD                UserSession,  
    WORD                RecordNum,  
    TOOL_TABLE_C4 *pRecordBuff,  
    DWORD *             pErrClass,  
    DWORD *             pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

pRecordBuff

[in] Puntatore alla struttura di tipo TOOL_TABLE_C4 che contiene i dati da scrivere nel record

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Ulteriori codici di errore:

Classe 4 – Errore 4 Il numero di record RecordNum è fuori range.

Vedi anche

GetAxisTabRecord_C, SetTabSem_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

GetOffsetTabRecord_C

Legge un record della tabella correttori del sistema

```
WORD GetOffsetTabRecord_C (  
    WORD                UserSession,  
    WORD                RecordNum,  
    OFFSET_TABLE_C4 *   pRecordBuff,  
    DWORD *             pErrClass,  
    DWORD *             pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

pRecordBuff

[out] Puntatore ad una struttura di tipo OFFSET_TABLE_C4 dove verra' scritto il record richiesto

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Ulteriori codici di errore:

Classe 4 – Errore 4 Il numero di record RecordNum è fuori range.

Vedi anche

SetOffsetRecord_C, SetTabSem_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

SetOffsetTabRecord_C

Scrive un record della tabella correttori del sistema.

```
WORD SetOffsetTabRecord_C (  
    WORD                UserSession,  
    WORD                RecordNum,  
    OFFSET_TABLE_C4 *   pRecordBuff,  
    DWORD *             pErrClass,  
    DWORD *             pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

pRecordBuff

[in] Puntatore alla struttura di tipo OFFSET_TABLE_C4 che contiene i dati da scrivere nel record

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Ulteriori codici di errore:

Classe 4 – Errore 4 Il numero di record RecordNum è fuori range.

Vedi anche

GetAxisTabRecord_C, SetTabSem_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

GetUserTabRecord_C

Legge un record della tabella utente del sistema.

```
WORD GetUserTabRecord_C (  
    WORD                UserSession,  
    WORD                RecordNum,  
    USER_TABLE_C4 *pRecordBuff,  
    DWORD *             pErrClass,  
    DWORD *             pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

pRecordBuff

[out] Puntatore ad una struttura di tipo USER_TABLE_C4 dove verra' scritto il record richiesto

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Ulteriori codici di errore:

Classe 4 – Errore 4 Il numero di record RecordNum è fuori range.

Vedi anche

SetUserTabRecord_C, SetTabSem_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

SetUserTabRecord_C

Scrive un record della tabella utente del sistema.

```
WORD SetUserTabRecord_C (  
    WORD                UserSession,  
    WORD                RecordNum,  
    USER_TABLE_C4 *pRecordBuff,  
    DWORD *             pErrClass,  
    DWORD *             pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

pRecordBuff

[in] Puntatore alla struttura di tipo USER_TABLE_C4 che contiene i dati da scrivere nel record

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Ulteriori codici di errore:

Classe 4 – Errore 4 Il numero di record RecordNum è fuori range.

Vedi anche

GetUserTabRecord_C, SetTabSem_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

GetTabFieldDouble_C

Legge un campo formato double di una tabella del sistema.

```
WORD GetTabFieldDouble_C (  
    WORD        UserSession,  
    WORD        TableNum,  
    WORD        RecordNum,  
    WORD        FieldNum,  
    double *     pValue,  
    DWORD *     pErrClass,  
    DWORD *     pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

TableNum

[in] Identificativo della tabella

RecordNum

[in] Numero del Record

FieldNum

[in] Identificativo del campo

pValue

[out] Puntatore alla variabile dove verrà scritto il valore del campo

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Ulteriori codici di errore:

Classe 4 – Errore 2 Il numero di record RecordNum è fuori range.

Classe 4 – Errore 3 Il numero di campo FieldNum è fuori range.

Classe 4 – Errore 4 Il formato del campo richiesto non è di tipo double.

Vedi anche

SetTabFieldDouble_C, SetTabSem_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

SetTabFieldDouble_C

Scrive un campo formato double di una tabella del sistema.

```
WORD SetTabFieldDouble_C (  
    WORD        UserSession,  
    WORD        TableNum,  
    WORD        RecordNum,  
    WORD        FieldNum,  
    double      Value,  
    DWORD *     pErrClass,  
    DWORD *     pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

TableNum

[in] Identificativo della tabella

RecordNum

[in] Numero del Record

FieldNum

[in] Identificativo del campo

Value

[in] Valore da scrivere nel campo

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Ulteriori codici di errore:

Classe 4 – Errore 2 Il numero di record RecordNum è fuori range.

Classe 4 – Errore 3 Il numero di campo FieldNum è fuori range.

Classe 4 – Errore 4 Il formato del campo indicato non è di tipo double.

Vedi anche

GetTabFieldDouble_C, SetTabSem_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

GetTabFieldShort_C

Legge un campo formato WORD di una tabella del sistema.

```
WORD GetTabFieldShort_C (  
    WORD        UserSession,  
    WORD        TableNum,  
    WORD        RecordNum,  
    WORD        FieldNum,  
    short *     pValue  
    DWORD *     pErrClass,  
    DWORD *     pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

TableNum

[in] Identificativo della tabella

RecordNum

[in] Numero del Record

FieldNum

[in] Identificativo del campo

pValue

[out] Puntatore alla variabile dove verrà scritto il valore del campo

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione è stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Ulteriori codici di errore:

Classe 4 – Errore 2 Il numero di record RecordNum è fuori range.

Classe 4 – Errore 3 Il numero di campo FieldNum è fuori range.

Classe 4 – Errore 4 Il formato del campo indicato non è di tipo short.

Vedi anche

SetTabFieldShort_C, SetTabSem_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

SetTabFieldShort_C

Scrive un campo formato WORD di una tabella del sistema.

```
WORD SetTabFieldShort_C (  
    WORD        UserSession,  
    WORD        TableNum,  
    WORD        RecordNum,  
    WORD        FieldNum,  
    short       Value  
    DWORD * pErrClass,  
    DWORD * pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

TableNum

[in] Identificativo della tabella

RecordNum

[in] Numero del Record

FieldNum

[in] Identificativo del campo

Value

[out] Valore da scrivere nel campo

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Ulteriori codici di errore:

Classe 4 – Errore 2 Il numero di record RecordNum è fuori range.

Classe 4 – Errore 3 Il numero di campo FieldNum è fuori range.

Classe 4 – Errore 4 Il formato del campo indicato non è di tipo short.

Vedi anche

GetTabFieldShort_C, SetTabSem_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

TabSearchDouble_C

Ricerca un valore in formato double nelle tabelle di sistema. Per effettuare la ricerca e' necessario specificare la tabella, il campo ed il valore da ricercare. E' possibile restringere la ricerca in una determinata gamma di record.

```
WORD TabSearchDouble_C (  
    WORD        UserSession,  
    WORD        TabNum,  
    WORD        FieldNum,  
    WORD        StartIndex,  
    WORD        StopIndex,  
    double      Value,  
    WORD *       pMatchIndex,  
    DWORD *     pErrClass,  
    DWORD *     pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

TabNum

[in] Identificativo della tabella

FieldNum

[in] Identificativo del campo

StartIndex

[in] Indice di inizio ricerca

StopIndex

[in] Indice di fine ricerca

Value

[in] Valore del campo

pMatchIndex

[out] Indice del primo valore che coincide

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Ulteriori codici di errore:



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

Classe 4 – Errore 2	Il numero di campo FieldNum è fuori range.
Classe 4 – Errore 3	StartIndex è più grande della dimensione della tabella.
Classe 4 – Errore 4	Il formato del campo richiesto non è di tipo double.
Classe 4 – Errore 5	StartIndex è più grande di StopIndex.
Classe 4 – Errore 6	La ricerca ha raggiunto StopIndex senza trovare il dato.
Classe 4 – Errore 7	(la ricerca ha raggiunto la fine della tabella senza trovare il dato)

Vedi anche

TabSearchShort_C, SetTabSem_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

TabSearchShort_C

Ricerca un valore in formato WORD nelle tabelle di sistema. Per effettuare la ricerca e' necessario specificare la tabella, il campo ed il valore da ricercare. E' possibile restringere la ricerca in una determinata gamma di record.

```
WORD TabSearchShort_C (  
    WORD        UserSession,  
    WORD        TabNum,  
    WORD        FieldNum,  
    WORD        StartIndex,  
    WORD        StopIndex,  
    WORD        Value,  
    WORD        Mask,  
    WORD *      pMatchIndex,  
    DWORD *     pErrClass,  
    DWORD *     pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

TabNum

[in] Identificativo della tabella

FieldNum

[in] Identificativo del campo

StartIndex

[in] Indice di inizio ricerca

StopIndex

[in] Indice di fine ricerca

Value

[in] Valore del campo

Mask

[in] Maschera per effettuare la ricerca solo su alcuni bit della WORD

pMatchIndex

[out] Indice del primo valore che coincide

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Ulteriori codici di errore:

Classe 4 – Errore 2	Il numero di campo FieldNum è fuori range.
Classe 4 – Errore 3	StartIndex è più grande della dimensione della tabella.
Classe 4 – Errore 4	Il formato del campo richiesto non è di tipo short.
Classe 4 – Errore 5	StartIndex è più grande di StopIndex.
Classe 4 – Errore 6	La ricerca ha raggiunto StopIndex senza trovare il dato.
Classe 4 – Errore 7	La ricerca ha raggiunto la fine della tabella senza trovare il dato.

Vedi anche

TabSearchDouble_C, SetTabSem_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

SetTabSem_C

La funzione SetTabSem imposta lo stato del semaforo di protezione delle tabelle (ASSI, UTENSILI, OFFSET o UTENTE).

Ogni tabella dispone di un semaforo per la protezione da accessi multipli contemporanei. Il semaforo di ciascuna tabella non ne impedisce l'accesso ma indica che la risorsa è già in uso.

Una tabella può risultare in uso, ad esempio, quando l'operatore del CN inserisce o modifica dei dati tramite l'applicativo TABLE EDITOR.

Ogni applicativo deve verificare di poter accedere alle tabelle tramite questa funzione e non deve modificarne i dati se risultano impegnate.

L'accesso ad una tabella dovrebbe essere fatto seguendo questa procedura:

1. Impostazione del semaforo a ROSSO per indicare alle altre applicazioni che si sta per accedere alla tabella. La funzione restituisce errore se la risorsa è già impegnata; in questo caso non si deve accedere alla tabella e bisogna adottare una opportuna politica di gestione (retry oppure segnalazione all'utente della condizione etc.)
2. Scrittura dei dati della tabella
3. Impostazione del semaforo a VERDE per indicare che l'accesso alla tabella è terminato.

```
WORD SetTabSem_C (  
    WORD          UserSession,  
    WORD          TableNum,  
    WORD          SemStatus,  
    DWORD * pErrClass,  
    DWORD * pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

TableNum

[in] Identificativo della tabella

SemStatus

[in] Stato del semaforo:

1 = ROSSO

2 = VERDE

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione è stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Ulteriori codici di errore:



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

Classe 4 – Errore 1
Classe 4 – Errore 2
Classe 4 – Errore 3
Classe 4 – Errore 4

Il valore di SemStatus non è valido.
L'identificativo di tabella TableNum è fuori range.
Lo stato del semaforo è già ROSSO.
Lo stato del semaforo è già VERDE.

Vedi anche

TabSemInfo_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

TabSemInfo_C

La funzione TabSemInfo restituisce lo stato del semaforo di protezione delle tabelle (ORIGINI, UTENSILI, CORRETTORI o UTENTE).

```
WORD TabSemInfo_C (  
    WORD          UserSession,  
    WORD *        pSemStatus,  
    DWORD * pErrClass,  
    DWORD * pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

pSemStatus

[out] Puntatore ad una variabile che viene settata con lo stato dei semafori di accesso alle tabelle, il bit a zero indica semaforo VERDE mentre a uno indica semaforo ROSSO

Bit 0: Tabella assi

Bit 1: Tabella utensili

Bit 2: Tabella offset

Bit 3: Tabella utente

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Vedi anche

SetTabSem_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

ResetSingleTableII_C

Azzerà il contenuto di una tabella (ORIGINI, UTENSILI, CORRETTORI, UTENTE, MAGAZZINI o POCKET).

```
WORD ResetSingleTableII_C (  
    WORD    UserSession,  
    WORD    TableNum,  
    DWORD * pErrClass,  
    DWORD * pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

TableNum

[in] Identificativo della tabella da azzerare:

- 0 = UTENSILI
- 1 = CORRETTORI
- 2 = ORIGINI
- 3 = UTENTE
- 4 = MAGAZZINI
- 5 = POCKET

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione è stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

LockTableII_C

Imposta a rosso lo stato del semaforo di protezione delle tabelle (ORIGINI, UTENSILI, CORRETTORI, UTENTE, MAGAZZINI o POCKET). Vedi SetTabSem_C.

```
WORD LockTableII_C (  
    WORD    UserSession,  
    WORD    TableNum,  
    DWORD * pErrClass,  
    DWORD * pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

TableNum

[in] Identificativo della tabella (vedi funzione ResetSingleTableII_C).

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Ulteriori codici di errore:

Classe 25 – Errore 14 L'identificativo di tabella TableNum è fuori range.

Classe 25 – Errore 17 Lo stato del semaforo e già ROSSO.

Vedi anche

UnLockTableII_C, SetTabSem_C.



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

UnLockTableII_C

Imposta a verde lo stato del semaforo di protezione delle tabelle (ORIGINI, UTENSILI, CORRETTORI, UTENTE, MAGAZZINI o POCKET). Vedi SetTabSem_C.

```
WORD UnLockTableII_C (  
    WORD    UserSession,  
    WORD    TableNum,  
    DWORD * pErrClass,  
    DWORD * pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

TableNum

[in] Identificativo della tabella (vedi funzione ResetSingleTableII_C).

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione è stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Ulteriori codici di errore:

Classe 25 – Errore 14 L'identificativo di tabella TableNum è fuori range.

Vedi anche

LockTableII_C, SetTabSem_C.



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

GetOriginTabRecordII_C

Legge un record della tabella origini del sistema.

```
WORD GetOriginTabRecordII_C (  
    WORD                UserSession,  
    WORD                RecordNum,  
    ORIGIN_TABLE_II_C4 * pRecordBuff ,  
    DWORD *             pErrClass,  
    DWORD *             pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

RecordNum

[in] indice base 1 del record che deve essere letto dalla tabella delle origini.

pRecordBuff

[out] Puntatore ad una struttura di tipo ORIGIN_TABLE_II_C4 dove verra' scritto il record richiesto

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Ulteriori codici di errore:

Classe 4 – Errore 4 Il numero di record RecordNum è fuori range.

Vedi anche

SetOriginTabRecordII_C, SetTabSem_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

SetOriginTabRecordII_C

Scrive un record della tabella origini del sistema.

```
WORD SetOriginTabRecordII_C (  
    WORD                UserSession,  
    WORD                RecordNum,  
    ORIGIN_TABLE_II_C4 * pRecordBuff,  
    DWORD *             pErrClass,  
    DWORD *             pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

RecordNum

[in] indice base 1 del record che deve essere modificato nella tabella origini.

pRecordBuff

[in] Puntatore alla struttura di tipo ORIGIN_TABLE_II_C4 che contiene i dati da scrivere nel record

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione è stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Ulteriori codici di errore:

Classe 4 – Errore 4 Il numero di record RecordNum è fuori range.

Vedi anche

GetOriginTabRecordII_C, SetTabSem_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

GetToolTabRecordII_C

Legge un record della tabella utensili del sistema.

```
WORD GetToolTabRecordII_C (  
    WORD                UserSession,  
    WORD                RecordNum,  
    TOOL_TABLE_II_C4 *  pRecordBuff,  
    DWORD *             pErrClass,  
    DWORD *             pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

RecordNum

[in] indice base 1 del record che deve essere letto dalla tabella utensili.

pRecordBuff

[out] Puntatore ad una struttura di tipo TOOL_TABLE_II_C4 dove verrà scritto il record richiesto

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Ulteriori codici di errore:

Classe 4 – Errore 4 Il numero di record RecordNum è fuori range.

Vedi anche

SetToolTabRecordII_C, SetTabSem_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

SetToolTabRecordII_C

Scrive un record della tabella utensili del sistema.

```
WORD SetToolTabRecordII_C (  
    WORD                UserSession,  
    WORD                RecordNum,  
    TOOL_TABLE_II_C4 *  pRecordBuff,  
    DWORD *             pErrClass,  
    DWORD *             pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

RecordNum

[in] indice base 1 del record che deve essere modificato nella tabella utensili.

pRecordBuff

[in] Puntatore alla struttura di tipo TOOL_TABLE_II_C4 che contiene i dati da scrivere nel record

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Ulteriori codici di errore:

Classe 4 – Errore 4 Il numero di record RecordNum è fuori range.

Vedi anche

GetToolTabRecordII_C, SetTabSem_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

GetOffsetTabRecordII_C

Legge un record della tabella correttori del sistema.

```
WORD GetOffsetTabRecordII_C (  
    WORD UserSession,  
    WORD RecordNum,  
    OFFSET_TABLE_II_C4 * pRecordBuff,  
    DWORD * pErrClass,  
    DWORD * pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

RecordNum

[in] indice base 1 del record che deve essere letto dalla tabella correttori.

pRecordBuff

[out] Puntatore ad una struttura di tipo OFFSET_TABLE_II_C4 dove verrà scritto il record richiesto

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Ulteriori codici di errore:

Classe 4 – Errore 4 Il numero di record RecordNum è fuori range.

Vedi anche

SetOffsetRecordII_C, SetTabSem_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

SetOffsetTabRecordII_C

Scrive un record della tabella correttori del sistema.

```
WORD SetOffsetTabRecordII_C (  
    WORD                UserSession,  
    WORD                RecordNum,  
    OFFSET_TABLE_II_C4 * pRecordBuff,  
    DWORD *             pErrClass,  
    DWORD *             pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

RecordNum

[in] indice base 1 del record che deve essere modificato nella tabella correttori.

pRecordBuff

[in] Puntatore alla struttura di tipo OFFSET_TABLE_II_C4 che contiene i dati da scrivere nel record

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Ulteriori codici di errore:

Classe 4 – Errore 4 Il numero di record RecordNum è fuori range.

Vedi anche

GetOffsetTabRecordII_C, SetTabSem_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

GetUserTabRecordII_C

Legge un record della tabella utente del sistema.

```
WORD GetUserTabRecordII_C (  
    WORD                UserSession,  
    WORD                RecordNum,  
    USER_TABLE_II_C4 *  pRecordBuff,  
    DWORD *             pErrClass,  
    DWORD *             pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

RecordNum

[in] indice base 1 del record che deve essere letto dalla tabella utente.

pRecordBuff

[out] Puntatore ad una struttura di tipo USER_TABLE_II_C4 dove verra' scritto il record richiesto

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Ulteriori codici di errore:

Classe 4 – Errore 4 Il numero di record RecordNum è fuori range.

Vedi anche

SetUserTabRecordII_C, SetTabSem_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

SetUserTabRecordII_C

Scrive un record della tabella utente del sistema.

```
WORD SetUserTabRecordII_C (  
    WORD                UserSession,  
    WORD                RecordNum,  
    USER_TABLE_II_C4 *  pRecordBuff,  
    DWORD *             pErrClass,  
    DWORD *             pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

RecordNum

[in] indice base 1 del record che deve essere modificato nella tabella utente.

pRecordBuff

[in] Puntatore alla struttura di tipo USER_TABLE_II_C4 che contiene i dati da scrivere nel record

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Ulteriori codici di errore:

Classe 4 – Errore 4 Il numero di record RecordNum è fuori range.

Vedi anche

GetUserTabRecordII_C, SetTabSem_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

GetMagazineTabRecordII_C

Legge un record della tabella magazzini del sistema.

```
WORD GetMagazineTabRecordII_C (  
    WORD                UserSession,  
    WORD                RecordNum,  
    MAGAZINE_TABLE_II_C4 * pRecordBuff,  
    DWORD *             pErrClass,  
    DWORD *             pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

RecordNum

[in] indice base 1 del record che deve essere letto dalla tabellamagazzini.

pRecordBuff

[out] Puntatore ad una struttura di tipo MAGAZINE_TABLE_II_C4 dove verrà scritto il record richiesto

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Vedi anche

SetMagazineTabRecordII_C, SetTabSem_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

SetMagazineTabRecordII_C

Scrive un record della tabella magazzini del sistema.

```
WORD SetMagazineTabRecordII_C (  
    WORD                UserSession,  
    WORD                RecordNum,  
    MAGAZINE_TABLE_II_C4 * pRecordBuff,  
    DWORD *             pErrClass,  
    DWORD *             pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

RecordNum

[in] indice base 1 del record che deve essere modificato nella tabellamagazzini.

pRecordBuff

[in] Puntatore alla struttura di tipo MAGAZINE_TABLE_II_C4 che contiene i dati da scrivere nel record

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Vedi anche

GetMagazineTabRecordII_C, SetTabSem_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

GetPocketTabRecordII_C

Legge un record della tabella pocket del sistema.

```
WORD GetPocketTabRecordII_C (  
    WORD UserSession,  
    WORD RecordNum,  
    POCKET_TABLE_II_C4 * pRecordBuff,  
    DWORD * pErrClass,  
    DWORD * pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

RecordNum

[in] indice base 1 del record che deve essere letto dalla tabellapocket.

pRecordBuff

[out] Puntatore ad una struttura di tipo POCKET_TABLE_II_C4 dove verrà scritto il record richiesto

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Vedi anche

SetPocketTabRecordII_C, SetTabSem_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

SetPocketTabRecordII_C

Scrive un record della tabella pocket del sistema.

```
WORD SetPocketTabRecordII_C (  
    WORD UserSession,  
    WORD RecordNum,  
    POCKET_TABLE_II_C4 * pRecordBuff,  
    DWORD * pErrClass,  
    DWORD * pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

RecordNum

[in] indice base 1 del record che deve essere modificato nella tabellapocket.

pRecordBuff

[in] Puntatore alla struttura di tipo POCKET_TABLE_II_C4 che contiene i dati da scrivere nel record

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Vedi anche

GetPocketTabRecordII_C, SetTabSem_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

ToolSearch_C

Ricerca la posizione di un utensile in un magazzino.

```
WORD ToolSearch_C (  
    char *                ToolCode,  
    WORD                 Magazine,  
    WORD *               ToolIndex,  
    WORD *               Pocket,  
    DWORD *              pErrClass,  
    DWORD *              pErrNum  
);
```

Parametri

ToolCode

[in] Identificatore dell'utensile da ricercare.

Magazine

[in] Indice del magazzino in cui effettuare la ricerca.

ToolIndex

[out] Indice dell'utensile nella tabella Utensili.

Pocket

[out] Indice della Pocket dove si trova l'utensile.

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Vedi anche

ToolSearchLife_C, ToolSearchDir_C, ToolSearchLifeDir_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

ToolSearchLife_C

Ricerca nel magazzino l'utensile la cui vita residua è più vicina possibile, ma comunque superiore, a quella specificata.

```
WORD ToolSearchLife_C (  
    char *                ToolCode,  
    WORD                 Magazine,  
    double               LifeVal,  
    WORD *               ToolIndex,  
    WORD *               Pocket,  
    DWORD *              pErrClass,  
    DWORD *              pErrNum  
);
```

Parametri

ToolCode

[in] Identificatore dell'utensile da ricercare.

Magazine

[in] Indice del magazzino in cui effettuare la ricerca.

LifeVal

[in] vita residua minima dell'utensile.

ToolIndex

[out] Indice dell'utensile nella tabella Utensili.

Pocket

[out] Indice della Pocket dove si trova l'utensile.

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione è stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Vedi anche

ToolSearch_C, ToolSearchDir_C, ToolSearchLifeDir_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

ToolSearchDir_C

Ricerca la posizione di un utensile in un magazzino. Permette di specificare il numero di pocket da cui iniziare la ricerca e la direzione di ricerca.

```
WORD ToolSearchDir_C (  
    char *          ToolCode,  
    WORD           Magazine,  
    WORD           StartPocket,  
    WORD           Direction,  
    WORD *         ToolIndex,  
    WORD *         Pocket,  
    DWORD *        pErrClass,  
    DWORD *        pErrNum  
);
```

Parametri

ToolCode

[in] Identificatore dell'utensile da ricercare.

Magazine

[in] Indice del magazzino in cui effettuare la ricerca.

StartPocket

[in] Posizione iniziale per la ricerca.

Direction

[in] Direzione di ricerca. I valori possibili sono:

0 = Ricerca in entrambe le direzioni

1 = Ricerca in direzione positiva

2 = Ricerca in direzione negativa

ToolIndex

[out] Indice dell'utensile nella tabella Utensili.

Pocket

[out] Indice della Pocket dove si trova l'utensile.

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

Vedi anche

[ToolSearch_C](#), [ToolSearchLife_C](#), [ToolSearchLifeDir_C](#)



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

ToolSearchLifeDir_C

Cerca nel magazzino dato l'utensile con vita rimasta maggiore uguale al valore richiesto e vicino alla posizione corrente.

```
WORD ToolSearchLifeDir_C (  
    char *          ToolCode,  
    WORD            Magazine,  
    double          LifeVal,  
    WORD            Direction,  
    WORD *          ToolIndex,  
    WORD *          Pocket,  
    DWORD *         pErrClass,  
    DWORD *         pErrNum  
);
```

Parametri

ToolCode

[in] Identificatore dell'utensile da ricercare.

Magazine

[in] Indice del magazzino in cui effettuare la ricerca.

LifeVal

[in] Vita residua minima dell'utensile.

Direction

[in] Direzione di ricerca. . I valori possibili sono:

0 = Ricerca in entrambe le direzioni

1 = Ricerca in direzione positiva

2 = Ricerca in direzione negativa

ToolIndex

[out] Indice dell'utensile nella tabella Utensili.

Pocket

[out] Indice della Pocket dove si trova l'utensile.

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

Vedi anche

[ToolSearch_C](#), [ToolSearchLife_C](#), [ToolSearchDir_C](#)



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

ToolSameFamily_C

Ricerca in un magazzino tutti gli utensili che appartengono della famiglia data. Trova sia gli utensili in magazzino che quelli fuori.

```
WORD ToolSameFamily_C (  
    char *                ToolCode,  
    WORD                 Magazine,  
    WORD *               ToolInMagazine,  
    WORD *               ToolOutMagazine,  
    DWORD *              pErrClass,  
    DWORD *              pErrNum  
);
```

Parametri

ToolCode

[in] Identificatore dell'utensile da ricercare.

Magazine

[in] Indice del magazzino in cui effettuare la ricerca.

Direction

[in] Direzione di ricerca.

ToolInMagazine

[out] Posizione dell'utensile in magazzino che appartiene alla famiglia.

ToolOutMagazine

[out] Posizione dell'utensile fuori magazzino che appartiene alla famiglia.

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Vedi anche

ToolSearch_C, ToolSearchLife_C, ToolSearchDir_C, ToolSearchLifeDir_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

ToolLoad_C

Inserisce un utensile in un magazzino, verificando la compatibilità tra l'utensile e la pocket selezionata.

```
WORD ToolLoad_C (  
    WORD ToolIndex,  
    WORD Magazine,  
    WORD Pocket,  
    DWORD * pErrClass,  
    DWORD * pErrNum  
);
```

Parametri

ToolIndex

[in] Indice dell'utensile nella tabella Utensili.

Magazine

[in] Indice del magazzino.

Pocket

[in] Indice della Pocket dove inserire l'utensile.

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Vedi anche

ToolUnload_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

ToolUnload_C

Rimuove un utensile da un magazzino, specificando se l'utensile è nel mandrino, nel cambio utensile o nel magazzino.

```
WORD ToolUnload_C (  
    WORD ToolIndex,  
    WORD Magazine,  
    WORD Pocket,  
    WORD Position,  
    DWORD * pErrClass,  
    DWORD * pErrNum  
);
```

Parametri

ToolIndex

[in] Indice dell'utensile nella tabella Utensili.

Magazine

[in] Indice del magazzino.

Pocket

[in] Indice della Pocket dove inserire l'utensile.

Position

[in] codice che identifica la posizione dell'utensile:

0 = Fuori magazzino

1 = In magazzino

2 = Nel mandrino

3 = Ne cambio utensile

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

Vedi anche
ToolUnload_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

PocketSearch_C

Ricerca nel magazzino la prima pocket che può accogliere l'utensile specificato.

```
WORD PocketSearch_C (  
    WORD ToolIndex,  
    WORD Magazine,  
    WORD StartPocket,  
    WORD Direction,  
    WORD * Pocket,  
    DWORD * pErrClass,  
    DWORD * pErrNum  
);
```

Parametri

ToolIndex

[in] Indice dell'utensile nella tabella Utensili.

Magazine

[in] Indice del magazzino.

StartPocket

[in] Posizione iniziale per la ricerca.

Direction

[in] Direzione di ricerca. I valori possibili sono:

0 = Ricerca in entrambe le direzioni

1 = Ricerca in direzione positiva

2 = Ricerca in direzione negativa

Pocket

[out] Indice della Pocket dove inserire l'utensile.

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione è stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Vedi anche

PocketClosest_C, PocketNearest_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

PocketClosest_C

Date due pocket restituisce quella più vicina ad una terza (il riferimento). Restituisce anche direzione e distanza.

```
WORD PocketClosest_C (  
    WORD ToolIndex,  
    WORD Magazine,  
    WORD CurrentPocket,  
    WORD Pocket1,  
    WORD Pocket2,  
    WORD * Direction,  
    WORD * Pocket,  
    WORD * HorizontalDistance,  
    WORD * VerticalDistance,  
    DWORD * pErrClass,  
    DWORD * pErrNum  
);
```

Parametri

ToolIndex

[in] Indice dell'utensile nella tabella Utensili.

Magazine

[in] Indice del magazzino.

CurrentPocket

[in] Pocket di riferimento.

Pocket1

[in] Indice della prima Pocket.

Pocket2

[in] Indice della seconda Pocket.

Direction

[out] Direzione di ricerca.

Pocket

[out] Indice della pocket più vicina.

HorizontalDistance

[in] distanza orizzontale tra le pocket indicizzate dai campi CurrentPocket e Pocket.

VerticalDistance

[in] distanza verticale tra le pocket indicizzate dai campi CurrentPocket e Pocket. Questo valore è rilevante soltanto nel caso di magazzini di tipo planare.

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Vedi anche

PocketSearch_C, PocketNearest_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

PocketNearest_C

Ricerca nel magazzino la pocket più vicina che può accogliere l'utensile specificato.

```
WORD PocketNearest_C (  
    WORD ToolIndex,  
    WORD Magazine,  
    WORD CurrentPocket,  
    WORD * Pocket,  
    DWORD * pErrClass,  
    DWORD * pErrNum  
);
```

Parametri

ToolIndex

[in] Indice dell'utensile nella tabella Utensili.

Magazine

[in] Indice del magazzino.

CurrentPocket

[in] Indice della pocket di riferimento.

Pocket

[out] Indice della pocket più vicina.

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Vedi anche

PocketSearch_C, PocketClosest_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

ToolLife_C

Aggiorna il campo di tool life di un utensile. Se in conseguenza dell'aggiornamento cambia anche lo stato del tool aggiorna anche quest'ultimo.

```
WORD ToolLife_C (  
    WORD                ToolIndex,  
    double              Value,  
    double *            RemLife,  
    DWORD *             pErrClass,  
    DWORD *             pErrNum  
);
```

Parametri

ToolIndex

[in] Indice dell'utensile nella tabella Utensili.

Value

[in] Valore sottraendo della vita residua dell'utensile.

RemLife

[out] Vita residua dell'utensile dopo l'aggiornamento.

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

LoadTableFromFile_C

Carica in memoria una tabella contenuta in un file.

```
WORD LoadTableFromFile_C (  
    LPWSTR          FilePath,  
    LPWSTR          AuxFilePath,  
    WORD            TableType,  
    WORD            Param1,  
    WORD            Param2,  
    DWORD *         pErrClass,  
    DWORD *         pErrNum  
);
```

Parametri

FilePath

[in] Stringa Unicode contenente il percorso del file nel file system locale del CNC.

AuxFilePath

[in] Stringa Unicode contenente il percorso del file ausiliario nel file system locale del CNC. Questo parametro viene utilizzato soltanto nel caso di caricamento della tabella Utensili. Se viene specificato deve corrispondere al nome del file che contiene la tabella dei Correttori associata alla tabella Utensili in caricamento. La tabella Correttori verrà caricata contestualmente alla tabella Utensili. Se non si desidera aggiornare la tabella Correttori al momento del caricamento della tabella Utensili questo parametro deve essere vuoto.

TableType

[in] Identificativo della tabella che verrà caricata:

- 0 = UTENSILI
- 1 = CORRETTORI
- 2 = ORIGINI
- 3 = UTENTE
- 4 = MAGAZZINI

Param1

[in] Parametro ausiliario 1. Viene utilizzato nel caso di caricamento della tabella Utensili o della tabella Origini.

Tabella Utensili: vengono aggiornati soltanto gli utensili appartenenti al magazzino di indice *Param1*. Nel caso in cui i magazzini non siano disponibili questo parametro non è rilevante e deve essere '0'.

Tabella Origini: vengono aggiornati i valori di origine soltanto per gli assi che appartengono al processo *Param1*.

Param2

[in] Parametro ausiliario 2. Viene utilizzato soltanto nel caso di caricamento di tabella Utensili con magazzini disponibili.

I valori accettati sono:

- 0 = le pocket non vengono aggiornate in conseguenza del caricamento della tabella Utensili



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

1 = le pocket vengono aggiornate in conseguenza del caricamento della tabella Utensili.
In questo caso se per un Utensile il campo Pocket è diverso da '0' e il campo stato è 'In magazzino' la pocket corrispondente viene segnata come occupata.

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

SaveTables_C

Salva tutte le tabelle utente.

```
WORD SaveTables_C (  
    WORD        UserSession,  
    LPWSTR      LocalDir,  
    DWORD *     pErrClass,  
    DWORD *     pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

LocalDir

[in] Percorso locale al controllo in cui verranno salvati i files delle tabelle.

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Ulteriori codici di errore:

Classe 25 – Errore 14 Richiesta per tipologia di tabella non prevista

Classe 25 – Errore 19 Errore nel salvataggio delle tabelle

Vedi anche

SaveSingleTable_C, RestoreSingleTable_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

SaveSingleTable_C

Salva la tabella utente specificata in input.

```
WORD SaveSingleTable_C (  
    WORD                UserSession,  
    LPWSTR              LocalDir,  
    TABLE_TYPE_II      TableType  
    DWORD *             pErrClass,  
    DWORD *             pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

LocalDir

[in] Percorso locale al controllo in cui verranno salvati i files delle tabelle.

TableType

[in] Identificativo del tipo di tabella da salvare.

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Ulteriori codici di errore:

Classe 25 – Errore 14 Richiesta per tipologia di tabella non prevista

Classe 25 – Errore 19 Errore nel salvataggio della tabella

Vedi anche

SaveTables_C, RestoreSingleTable_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

RestoreSingleTable_C

Ripristina la tabella utente specificata in input.

```
WORD RestoreSingleTables_C (  
    WORD                UserSession,  
    LPWSTR              LocalDir,  
    TABLE_TYPE_II      TableType  
    DWORD *             pErrClass,  
    DWORD *             pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

LocalDir

[in] Percorso locale al controllo in cui è salvato il file della tabella da ripristinare.

TableType

[in] Identificativo del tipo di tabella da salvare.

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Ulteriori codici di errore:

Classe 25 – Errore 14 Richiesta per tipologia di tabella non prevista

Classe 25 – Errore 18 Errore nel ripristino della tabella

Vedi anche

SaveTables_C, SaveSingleTable_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

SaveBackupMemory_C

Salva il contenuto di tutta la dual port ram in un file situato nel file system del CNC.

```
WORD SaveBackupMemory_C (  
    WORD        UserSession,  
    LPWSTR       FileName,  
    DWORD *      pErrClass,  
    DWORD *      pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

FileName

[in] Percorso completo del file in cui verrà salvato il contenuto della memoria ritentiva.

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Ulteriori codici di errore:

Classe 25 – Errore 30 Errore nel salvataggio della Memoria Ritentiva

Classe 25 – Errore 31 Dual Port ram corrotta

Classe 25 – Errore 33 Errore di allocazione memoria durante ripristino della Memoria Ritentiva

Vedi anche

RestoreBackupMemory_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

RestoreBackupMemory_C

Ripristina il contenuto di tutta la dual port ram partendo da un file situato nel file system del CNC.

```
WORD RestoreBackupMemory_C (  
    WORD      UserSession,  
    LPWSTR    FileName,  
    DWORD     RestoreMask,  
    DWORD *   pErrClass,  
    DWORD *   pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

FileName

[in] Percorso completo del file dal quale ripristinare la memoria ritentiva. Il file deve essere situato nel file system del CNC.

RestoreMask

[in] Maschera delle aree di memoria ritentiva da sovrascrivere. Può essere una combinazione qualsiasi dei seguenti valori:

Simbolo	Valore	Descrizione
RESET GW	0x00000001	Resetta memoria GW
RESET GD	0x00000002	Resetta memoria GD
RESET L	0x00000004	Resetta memoria L
RESET LS	0x00000008	Resetta memoria LS
RESET PLCM	0x00000010	Resetta memoria retain PLC
RESET PROC	0x00000020	Resetta memoria per processo
RESET ALL	0x1000003f	Resetta tutta la dualport

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Ulteriori codici di errore:

Classe 25 – Errore 30 Errore nel salvataggio della Memoria Ritentiva

Classe 25 – Errore 31 Dual Port ram corrotta

Classe 25 – Errore 33 Errore di allocazione memoria durante ripristino della Memoria Ritentiva

Classe 25 – Errore 32 Errore nel ripristino della Memoria Ritentiva



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

Vedi anche

SaveBackupMemory_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

Funzioni dedicate al Processo

Queste funzioni permettono di scambiare dati con i Processi del CNC, inviare comandi ed impostare i parametri di lavorazione.

Queste funzioni possono essere chiamate solo dopo aver creato il server Cndex (vedi ConnectServer_C), aperto una sessione di comunicazione (vedi OpenSession_C) ed aver verificato che il CNC si trovi nella modalita' RUN (vedi BootPhaseEnquiry_C).



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

Cycle_C

Invia al processo il comando di inizio/fine esecuzione comando (inizio ciclo di lavorazione, movimento assi manuale, esecuzione comando MDI etc.).

```
WORD Cycle_C (  
    WORD      UserSession,  
    WORD      ProcNum,  
    WORD      Cmd  
    DWORD *   pErrClass,  
    DWORD *   pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

ProcNum

[in] Identificatore del processo (numero da 1 a 24).

Cmd

[in] Valore che determina se il comando e' Cycle Start o Cycle Stop:

1 : cycle start

0 : cycle stop

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Note

Questo comando è asincrono, la funzione non aspetta nè che il comando finisca nè che entri in esecuzione. Per aspettare che il comando entri in esecuzione utilizzare la funzione SyncrCycle_C.

Vedi anche

SyncroCycle_C, Hold_C, Reset_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

SyncroCycle_C

Invia al processo il comando d'inizio esecuzione comando (inizio ciclo di lavorazione, movimento assi manuale, esecuzione comando MDI etc.).

```
WORD SyncroCycle_C (  
    WORD      UserSession,  
    WORD      ProcNum,  
    DWORD *   pErrClass,  
    DWORD *   pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

ProcNum

[in] Identificatore del processo (numero da 1 a 24).

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione è stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Note

A differenza della funzione Cycle_C che è asincrona, questa funzione sincronizza l'avvio del comando vale a dire che attende che il sistema abbia messo in esecuzione il comando prima di restituire il controllo all'applicazione.

Vedi anche

Cycle_C, Hold_C, Reset_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

Reset_C

Invia al processo il comando di reset (interruzione dell'operazione in corso e ripristino dei valori predefiniti)

```
WORD Reset_C (  
    WORD UserSession,  
    WORD ProcNum  
    DWORD * pErrClass,  
    DWORD * pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

ProcNum

[in] Identificatore del processo (numero da 1 a 24).

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Vedi anche

Cycle_C, Hold_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

Hold_C

Invia al processo il comando di hold (sospensione e ripresa del ciclo di lavorazione).

```
WORD Hold_C (  
    WORD      UserSession,  
    WORD      ProcNum,  
    WORD      Cmd  
    DWORD *   pErrClass,  
    DWORD *   pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

ProcNum

[in] Identificatore del processo (numero da 1 a 24).

Cmd

[in] Valore che determina se il comando e' Hold On o Hold Off:

1 : Hold On

0 : Hold Off

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Ulteriori codici di errore:

Classe 4 – Errore 107 Uno degli assi non è sul profilo.

Vedi anche

Cycle_C, Reset_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

SetFeedManOver_C

Imposta il valore del Feed Manual Override (variazione percentuale della velocità per i movimenti manuali) e la direzione dei movimenti manuali.

```
WORD SetFeedManOver_C (  
    WORD      UserSession,  
    WORD      ProcNum,  
    WORD      FeedManOver,  
    WORD      Direction,  
    DWORD *   pErrClass,  
    DWORD *   pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

ProcNum

[in] Identificatore del processo (numero da 1 a 24).

FeedManOver

[in] Valore percentuale da impostare. Il numero deve essere moltiplicato per 100 (10000 indica 100%, 7500 indica 75% etc.)

Direction

[in] Determina la direzione dei movimenti manuali

0: direzione "positiva"

1: direzione "negativa"

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione è stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Vedi anche

SetFeedRateOver_C, SetFeedRapidOver_C, SetSpeedRateOver_C, GetNcInfo1_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

SetFeedRateOver_C

Imposta il valore del Feed Rate Override (variazione percentuale della velocità per i movimenti in continuo).

```
WORD SetFeedRateOver_C (  
    WORD UserSession,  
    WORD ProcNum,  
    WORD FeedRateOver,  
    DWORD * pErrClass,  
    DWORD * pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

ProcNum

[in] Identificatore del processo (numero da 1 a 24).

FeedRateOver

[in] Valore percentuale da impostare. Il numero deve essere moltiplicato per 100 (10000 indica 100%, 7500 indica 75% etc.)

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione è stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Vedi anche

SetFeedRapidOver_C, SetFeedManOver_C, SetSpeedRateOver_C, GetNcInfo1_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

SetFeedRapidOver_C

Imposta il valore del Feed Rapid Override (variazione percentuale della velocità di rapido).
Il Rapid Override è attivo solo quando è abilitato il controllo della velocità di rapido.

```
WORD SetFeedRapidOver_C (  
    WORD UserSession,  
    WORD ProcNum,  
    WORD FeedRapidOver,  
    DWORD * pErrClass,  
    DWORD * pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

ProcNum

[in] Identificatore del processo (numero da 1 a 24).

FeedRapidOver

[in] Valore percentuale da impostare. Il numero deve essere moltiplicato per 100 (10000 indica 100%, 7500 indica 75% etc.)

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione è stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Vedi anche

SetFeedRateOver_C, SetFeedManOver_C, SetSpeedRateOver_C, GetNcInfo1_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

SetSpeedRateOver_C

Imposta il valore dello Speed Override (variazione percentuale della velocità del mandrino).

```
WORD SetSpeedRateOver_C (  
    WORD      UserSession,  
    WORD      ProcNum,  
    WORD      SpeedRateOver,  
    DWORD *   pErrClass,  
    DWORD *   pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

ProcNum

[in] Identificatore del processo (numero da 1 a 24).

SpeedRateOver

[in] Valore percentuale da impostare. Il numero deve essere moltiplicato per 100 (10000 indica 100%, 7500 indica 75% etc.)

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione è stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Vedi anche

SetFeedRateOver_C, SetFeedRapidOver_C, SetFeedManOver_C, GetNcInfo1_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

SetManMovDirection_C

Imposta la direzione dei movimenti manuali. L'impostazione della direzione di movimento tramite questa funzione equivale all'impostazione effettuata tramite il parametro Direction della funzione SetFeedManOver_C. A differenza di quest'ultima la funzione SetManMovDirection_C permette di modificare la direzione di movimento senza variare il valore del Feed Manual Override

```
WORD SetFeedManOver_C (  
    WORD      UserSession,  
    WORD      ProcNum,  
    WORD      Direction,  
    DWORD *   pErrClass,  
    DWORD *   pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

ProcNum

[in] Identificatore del processo (numero da 1 a 24).

Direction

[in] Determina la direzione dei movimenti manuali

0: direzione "positiva"

1: direzione "negativa"

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Vedi anche

SetFeedManOver_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

GetVarJOG_C

Legge il valore dell'incremento per il Modo Manuale Incrementale (movimenti manuali incrementali degli assi).

```
WORD GetVarJOG_C (  
    WORD      UserSession,  
    WORD      ProcNum,  
    double *   Value,  
    DWORD *   pErrClass,  
    DWORD *   pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

ProcNum

[in] Identificatore del processo (numero da 1 a 24).

Value

[out] Puntatore alla variabile dove verra' scritto il valore attuale del JOG.

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Vedi anche

GetNcInfo1_C, SetVarJOG_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

SetVarJOG_C

Imposta il valore dell'incremento per il Modo Manuale Incrementale (movimenti manuali incrementali degli assi).

```
WORD SetVarJOG_C (  
    WORD      UserSession,  
    WORD      ProcNum,  
    double    Value,  
    DWORD *   pErrClass,  
    DWORD *   pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

ProcNum

[in] Identificatore del processo (numero da 1 a 24).

Value

[in] Valore di JOG da impostare

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Vedi anche

GetVarJOG_C, GetNcInfo1_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

SetVarUAS_C

Attiva o disattiva la modalità di test per la prova programmi (DryRun).

```
WORD SetVarUAS_C (  
    WORD      UserSession,  
    WORD      ProcNum,  
    WORD      Mode,  
    DWORD *   pErrClass,  
    DWORD *   pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

ProcNum

[in] Identificatore del processo (numero da 1 a 24).

Mode

[in] Selettore per attivare o disattivare il DryRun. I possibili valori sono:

0 = disattivazione

1 = attivazione

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

La modalità di test non può essere attivata se e' attiva la Ricerca Memorizzata. In questo caso la funzione restituisce il codice di errore ERR_UAS_DISABLED_WHEN_RCM_ON.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

GetVarRCM_C

Legge il flag RCM che indica lo stato della Ricerca Memorizzata.

```
WORD GetVarRCM_C (  
    WORD UserSession,  
    WORD ProcNum,  
    BYTE * pValue,  
    DWORD * pErrClass,  
    DWORD * pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

ProcNum

[in] Identificatore del processo (numero da 1 a 24).

pValue

[out] Puntatore alla variabile dove verrà scritto il valore attuale del flag di RCM.
Vale 0 se RCM è disattivo, 1 se è attivo.

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione è stata eseguita con successo, zero in caso di errore.
Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Vedi anche

SetVarRCM_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

SetVarRCM_C

Attiva o disattiva il flag di RCM vale a dire lo stato della Ricerca Memorizzata.

```
WORD SetVarRCM_C (  
    WORD      UserSession,  
    WORD      ProcNum,  
    WORD      Mode,  
    DWORD *   pErrClass,  
    DWORD *   pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

ProcNum

[in] Identificatore del processo (numero da 1 a 24).

Mode

[in] Valore da impostare. 0 disattiva l'RCM, 1 lo attiva.

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Vedi anche

SetVarRCM_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

LoadPTech_C

Effettua la copia di un set di valori all'interno della variabile matriciale \$PTECH .

```
WORD LoadPTech_C (  
    WORD                UserSession,  
    WORD                ProcNum,  
    LPSTR               PPPathName,  
    PTECH_WRITE_MODE    WriteMode,  
    WORD                Sheet,  
    WORD                Line,  
    WORD                Column,  
    WORD                NumVar,  
    DOUBLE *            pData,  
    DWORD *             pErrClass,  
    DWORD *             pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

ProcNum

[in] Identificatore del processo (numero da 1 a 24).

PPPathName

[in] Pathname del part program per il quale caricare i valori della variabile \$PTECH. Se il campo è vuoto viene aggiornata la variabile \$PTECH associata al part program attivo, altrimenti viene aggiornata la variabile \$PTECH associata al programma precaricato specificato da PPPathName.

WriteMode

[in] identifica la modalità di accesso in scrittura della variabile. Le modalità disponibili sono:

PTECH_WRITE_SHEET_C scrittura di un "foglio" (matrice bidimensionale) della variabile \$PTECH

PTECH_WRITE_LINE_C scrittura di una riga di un "foglio" della variabile \$PTECH

PTECH_WRITE_ELEMENT_C scrittura di un elemento di una riga di un "foglio" della variabile \$PTECH

Sheet

[in] numero del "foglio" della variabile \$PTECH nel quale verranno caricati i valori.

Line

[in] numero della riga nella quale verranno caricati i valori. Questo parametro è valido nelle modalità **PTECH_WRITE_LINE_C** e **PTECH_WRITE_ELEMENT_C**.

Column

[in] colonna nella quale verrà caricato il valore. Questo parametro è valido nella modalità **PTECH_WRITE_ELEMENT_C**.

NumVar



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

[in] dimensione totale in byte dei dati inviati. Serve a verificare che non ci siano overflow durante la fase di caricamento dei dati.

pData

[in] puntatore alla zona di memoria contenente i dati da caricare.

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Vedi anche

GetPTechSizes_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

GetPTechSizes_C

Restituisce le tre dimensioni della variabile \$PTECH

```
WORD GetPTechSizes_C (  
    WORD      UserSession,  
    WORD      ProcNum,  
    WORD *    Ind1,  
    WORD *    Ind2,  
    WORD *    Ind3,  
    DWORD *   pErrClass,  
    DWORD *   pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

ProcNum

[in] Identificatore del processo (numero da 1 a 24).

Ind1

[out] numero di "fogli" della variabile \$PTECH

Ind2

[out] numero di righe di un "foglio" della variabile \$PTECH

Ind3

[out] numero di colonne di una riga della variabile \$PTECH

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Vedi anche

LoadPTech_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

SetMdiString_C

Imposta il comando da eseguire a fronte di un cycle start nella modalita' MDI del processo.
Il comando impostato viene messo in esecuzione dalla funzione Cycle_C dopo aver selezionato la modalita' MDI tramite la funzione SetProcessMode_C.

```
WORD SetMdiString_C (  
    WORD      UserSession,  
    WORD      ProcNum,  
    LPSTR     pMdiString,  
    DWORD *   pErrClass,  
    DWORD *   pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

ProcNum

[in] Identificatore del processo (numero da 1 a 24).

pMdiString

[in] Puntatore ad una stringa che contiene il comando

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Vedi anche

Cycle_C, SyncroCycle_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

SetProcessMode_C

Imposta la modalita' di lavoro (MDI, Manuale, Esecuzione Part Program etc.) del Processo.

```
WORD SetProcessMode_C (  
    WORD      UserSession,  
    WORD      ProcNum,  
    WORD      Mode,  
    DWORD *   pErrClass,  
    DWORD *   pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

ProcNum

[in] Identificatore del processo (numero da 1 a 24).

Mode

[in] Valore corrispondente al modo del processo:

- 1 : MDI
- 2 : AUTO
- 3 : STEP
- 4 : JOG MANUALE
- 5 : JOG INCREMENTALE
- 6 : RITORNO SUL PROFILO
- 7 : HOME
- 8 : VOLANTINO

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Ulteriori codici di errore:

Classe 4 – Errore 120 Il valore del mode Mode è fuori range.

Vedi anche

GetNcInfo1_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

SelectProcess_C

Imposta il processo di lavoro, cioè il processo selezionato, sul CNC.

La modifica del processo selezionato sul CNC non ha interazioni con le funzioni di questa libreria, il processo di riferimento per queste funzioni è sempre indicato come parametro delle singole funzioni.

```
WORD SelectProcess_C (  
    WORD      UserSession,  
    WORD      ProcNum,  
    DWORD *   pErrClass,  
    DWORD *   pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

ProcNum

[in] Identificatore del processo da selezionare sul CNC (numero da 1 a 24).

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione è stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Vedi anche

GetSelectedProcess_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

GetSelectedProcess_C

Legge il numero del processo selezionato sul CNC.

```
WORD GetSelectedProcess_C (  
    WORD UserSession,  
    WORD * pProcNum,  
    DWORD * pErrClass,  
    DWORD * pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

pProcNum

[in] Identificatore del processo da selezionare sul CNC (numero da 1 a 24).

pProcNum

[out] Puntatore ad una variabile dove verra' scritto il numero del processo selezionato sul CNC.

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Vedi anche

SelectProcess_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

SelectProcAxis_C

Selezione dell'asse del processo per i movimenti nelle modalita' manuali.

```
WORD SelectProcAxis_C (  
    WORD      UserSession,  
    WORD      ProcNum,  
    BYTE      AxisName,  
    DWORD *   pErrClass,  
    DWORD *   pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

ProcNum

[in] Identificatore del processo (numero da 1 a 24).

AxisName

[in] Nome dell'asse

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Ulteriori codici di errore:

Classe 4 – Errore 121 Il nome dell'asse AxisName è sbagliato.

Vedi anche

GetNcInfo1_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

SelectPartProgram_C

Seleziona, per un processo, il part program da eseguire. Può anche essere usata per deselegionare il part program corrente.

```
WORD SelectPartProgram_C (  
    WORD      UserSession,  
    WORD      ProcNum,  
    LPSTR     pPPName,  
    DWORD *   pErrClass,  
    DWORD *   pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

ProcNum

[in] Identificatore del processo (numero da 1 a 24).

pPPName

[in] Puntatore alla stringa che contiene il nome path-name del part program. Il path-name deve fare riferimento ad un drive fisico del CNC. Se la stringa è vuota, la funzione deselegiona il part program corrente.

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Ulteriori codici di errore:

Classe 4 – Errore 1	Errore di sintassi.
Classe 4 – Errore 8	Errore di formato.
Classe 4 – Errore 9	Simbolo non definito.
Classe 4 – Errore 53	Label duplicata.
Classe 4 – Errore 55	Label troppo lunga.
Classe 4 – Errore 56	Overflow della tabella soubroutine.
Classe 4 – Errore 57	Overflow della tabella label.
Classe 4 – Errore 65	Errore durante la gestione del file part-program.
Classe 4 – Errore 66	File part-program non trovato.

Vedi anche

SelectPartProgramFromDrive_C, GetNcInfo1_C, ManagePartProgram_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

SelectPartProgramFromDrive_C

Seleziona, per un processo, il part program da eseguire. Può anche essere usata per deselezionare il part program corrente.

```
WORD SelectPartProgramFromDrive_C (  
    WORD        UserSession,  
    WORD        ProcNum,  
    LPSTR       pPPName,  
    DWORD *     pErrClass,  
    DWORD *     pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

ProcNum

[in] Identificatore del processo (numero da 1 a 24).

pPPName

[in] Puntatore alla stringa che contiene il path-name del part program. Il path-name deve fare riferimento ad un drive logico configurato con l'applicazione FileBrowser. Se la stringa è vuota, la funzione deseleziona il part program corrente.

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione è stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Ulteriori codici di errore:

Classe 4 – Errore 1	Errore di sintassi.
Classe 4 – Errore 8	Errore di formato.
Classe 4 – Errore 9	Simbolo non definito.
Classe 4 – Errore 53	Label duplicata.
Classe 4 – Errore 55	Label troppo lunga.
Classe 4 – Errore 56	Overflow della tabella soubroutine.
Classe 4 – Errore 57	Overflow della tabella label.
Classe 4 – Errore 65	Errore durante la gestione del file part-program.
Classe 4 – Errore 66	File part-program non trovato.

Vedi anche

SelectPartProgram_C, GetNcInfo1_C, ManagePartProgram_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

ManagePartProgram_C

Seleziona, per un processo, l'operazione da eseguire e il part program sul quale eseguirla.

```
WORD ManagePartProgram_C (  
    WORD        UserSession,  
    WORD        ProcNum,  
    WORD        Mode,  
    LPSTR       pPPName,  
    LPSTR       ErrString,  
    DWORD *     pErrClass,  
    DWORD *     pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

ProcNum

[in] Identificatore del processo (numero da 1 a 24).

Mode

[in] Operazione richiesta. I valori possibili per le operazioni da eseguire sono i seguenti:

REL_SPG	disattiva part program. Equivale alla SelectPartProgramFromDrive_C o alla SelectPartProgram_C chiamate con pPPName uguale a stringa vuota
SPG_EXE	attiva part program. Equivale alla SelectPartProgram_C
SPG_PASSANTE	attiva la modalità passante. Equivale alla DncInit_C
SPG_DRIVE	attiva part program il cui nome contiene un drive logico. Equivale alla SelectPartProgramFromDrive_C
SPG_NO_ANA	attiva part program, senza preanalisi
SPG_NO_ANA_DRIVE	attiva part program il cui nome contiene un drive logico, senza preanalisi
SPG_SWAP	effettua lo swap tra il programma corrente e un programma precaricato
SPG_SWAP_DRIVE	effettua lo swap tra il programma corrente e un programma precaricato
PRELOAD	recarica un partprogram
PRELOAD_DRIVE	precarica un partprogram il cui nome contiene un drive logico
UNLOAD	rimuove un part program caricato
UNLOAD_DRIVE	rimuove un part program caricato il cui nome contiene un drive logico
UNLOAD_TUTTI	rimuove tutti i part program caricati

pPPName

[in] Puntatore alla stringa che contiene il path-name del part program. Il path-name deve essere coerente con l'operazione richiesta. Nel caso cioè l'operazione sia del tipo con drive logico il path name deve fare riferimento ad un drive logico configurato con l'applicazione FileBrowser.



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

ErrString

[out] Stringa contenente eventuali informazioni riguardo l'errore avvenuto

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Ulteriori codici di errore:

Classe 23 – Errore 1	Errore di sintassi.
Classe 23 – Errore 6	Errore di formato.
Classe 23 – Errore 56	Simbolo non definito.
Classe 23 – Errore 155	Label duplicata.
Classe 23 – Errore 157	Label troppo lunga.
Classe 23 – Errore 158	Overflow della tabella soubroutine.
Classe 23 – Errore 159	Overflow della tabella label.
Classe 23 – Errore 103	Errore durante la gestione del file part-program.
Classe 23 – Errore 100	File part-program non trovato.

Vedi anche

SelectPartProgram_C, SelectPartProgramFromDrive_C, GetNcInfo1_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

GetActivePartProgram_C

Legge il nome del programma e del sottoprogramma selezionati ed indica il livello di sottoprogramma raggiunto.

```
WORD GetActivePartProgram_C (  
    WORD        UserSession,  
    WORD        ProcNum,  
    WORD *      pLevel,  
    LPSTR       pMain,  
    LPSTR       pSub,  
    DWORD *     pErrClass,  
    DWORD *     pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

ProcNum

[in] Identificatore del processo (numero da 1 a 24).

pLevel

[in] Puntatore alla variabile dove sarà scritto il livello di nesting (annidamento di chiamate) raggiunto dal sottoprogramma attivo. Il livello di nesting vale 1 per il programma principale e può arrivare fino a 6.

pMain

[out] Puntatore ad una stringa di 130 caratteri nella quale sarà scritto il nome del programma principale selezionato. Il testo ha terminatore 0.

pSub

[out] Puntatore ad una stringa di 130 caratteri nella quale sarà scritto il nome del sottoprogramma selezionato. Il testo ha terminatore 0.

pErrClass

[out] Puntatore alla variabile dove sarà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove sarà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

GetActivePartProgramFullPath_C

Legge il path completo, compreso del nome, del programma e del sottoprogramma selezionati ed indica il livello di sottoprogramma raggiunto.

```
WORD GetActivePartProgramFullPath_C (  
    WORD        UserSession,  
    WORD        ProcNum,  
    WORD *      pLevel,  
    LPSTR       pMain,  
    LPSTR       pSub,  
    DWORD *     pErrClass,  
    DWORD *     pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

ProcNum

[in] Identificatore del processo (numero da 1 a 24).

pLevel

[in] Puntatore alla variabile dove sarà scritto il livello di nesting (annidamento di chiamate) raggiunto dal sottoprogramma attivo. Il livello di nesting vale 1 per il programma principale e può arrivare fino a 6.

pMain

[out] Puntatore ad una stringa di 130 caratteri nella quale sarà scritto il path completo, con nome, del programma principale selezionato. Il testo ha terminatore 0.

pSub

[out] Puntatore ad una stringa di 130 caratteri nella quale sarà scritto il path completo, con nome, del sotto-programma selezionato. Il testo ha terminatore 0.

pErrClass

[out] Puntatore alla variabile dove sarà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove sarà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione è stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

GetPartProgramLines _C

Legge fino a 8 linee del programma in esecuzione. La linea 2 (pPPLine2) e' quella in esecuzione quando il CNC è nello stato RUN o quella da eseguire quando il CNC è nello stato IDLE.

```
WORD GetPartProgramLines_C (  
    WORD      UserSession,  
    WORD      ProcNum,  
    LPSTR     pPPLine1,  
    LPSTR     pPPLine2,  
    LPSTR     pPPLine3,  
    LPSTR     pPPLine4,  
    LPSTR     pPPLine5,  
    LPSTR     pPPLine6,  
    LPSTR     pPPLine7,  
    LPSTR     pPPLine8,  
    DWORD *   pErrClass,  
    DWORD *   pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

ProcNum

[in] Identificatore del processo (numero da 1 a 24).

pPPLine1, PPLine2... PPLine8

[out] Puntatori a stringhe di 130 caratteri nella quali saranno scritte le linee di programma. Ogni linea ha terminatore 0.

pErrClass

[out] Puntatore alla variabile dove sarà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove sarà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

GetAxOriginNum_C

Legge il numero ed il tipo di origine impostata per gli assi di un processo.

```
WORD GetAxOriginNum_C (  
    WORD UserSession,  
    WORD ProcNum,  
    WORD * pAxesNum,  
    AX_ORIG_NUM_C4 * pAxOriginNum,  
    DWORD * pErrClass,  
    DWORD * pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

ProcNum

[in] Identificatore del processo (numero da 1 a 24).

pAxesNum

[in,out] Puntatore ad una variabile:

[In] in input deve essere inizializzata con il numero di elementi dell'array puntato da pAxOriginNum.

[Out] in output conterrà il numero di assi letti, ovvero di elementi significativi dell'array.

pAxOriginNum

[out] Puntatore ad un array di strutture di tipo AX_ORIG_NUM_C4 nelle quali saranno scritti i dati. Il numero di elementi dell'array deve essere almeno pari al numero specificato con pAxesNum. Il numero massimo di assi per processo e' 9 for 10/Series, 12 for OPENControl. Il campo "OriginFlag" della struttura indica il tipo di origine attiva sull'asse.

pErrClass

[out] Puntatore alla variabile dove sarà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove sarà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Vedi anche

Struttura AX_ORIG_NUM_C4 nel paragrafo "Descrizione delle Strutture"



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

GetAxesPosition_C

Restituisce informazioni circa la posizione degli assi interpolati e ausiliari del processo

```
WORD GetAxesPosition_C (  
    WORD UserSession,  
    WORD ProcNum,  
    BYTE AxisName,  
    WORD Select,  
    WORD * pNumAxis,  
    GETINTDATA_C4 * pIntPos,  
    DWORD * pErrClass,  
    DWORD * pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

ProcNum

[in] Identificatore del processo (numero da 1 a 24).

AxisName

[in] Nome asse.

Deve indicare il nome ASCII dell'asse quando si richiede la posizione di un asse solo. Deve valere 0 per richiedere la posizione di piu' assi.

Select

[in] Valore corrispondente al tipo di dati che si intendono leggere:

- 1 : Programmed position (la posizione asse comprende tutti gli offset, origini etc)
Corrisponde alle quote PROGRAMMED visualizzate dal CNC.
- 2 : Interpolated position (e' l'output del calcolo non comprende gli offset, origini etc.).
Sommato al TotalOffset corrisponde alle quote WORK visualizzate dal CNC.
- 3 : Transducer position (e' la posizione letta dal trasduttore, non comprende gli offset, origini etc.). Sommando il campo Position ed il campo TotalOffset si ottengono le quote MACHINE visualizzate dal CNC.
- 4 : Following error (e' l'errore di posizione calcolato come differenza tra la posizione calcolata e quella fisica)
- 5 : Distance to go (e' la differenza tra la quota da raggiungere e quella attuale)
- 6 : Interpolated position (e' l'output del calcolo non comprende gli offset, origini etc.).
Corrisponde alle quote ABSOLUTE visualizzate dal CNC.

pNumAxis

[in,out] Puntatore ad una variabile:

[In] inizializzata con il numero di elementi dell'array.

[Out] che contiene il numero di assi configurati, ovvero di elementi significativi dell'array.

pIntPos

[out] Puntatore ad un array GETINTDATA_C4

pErrClass



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

GetNcInfo1_C

Restituisce un insieme di parametri del processo

```
WORD GetNcInfo1_C (  
    WORD UserSession,  
    WORD ProcNum,  
    GETINFO1DATA_C4 * pGetInfo,  
    DWORD * pErrClass,  
    DWORD * pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

ProcNum

[in] Identificatore del processo (numero da 1 a 24).

pGetInfo

[out] Puntatore alla struttura di tipo GETINFO1DATA_C4 che conterra' i dati

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Note

Tra i dati letti da questa funzione ci sono i nomi utensile in uso e programmato.

Per i CNC Serie10 i nomi utensile sono delle stringhe numeriche che nella struttura dati letta da questa funzione sono rappresentati da variabili con formato double.

Il CNC OPENControl può usare come nome utensile sia un numero sia un testo alfanumerico.

Se si utilizza questa funzione per leggere i nomi utensile di un CNC OPENControl, nei campi **actual_tool** e **progr_tool** ci saranno i nomi degli utensili convertiti in floating point. Se il nome è alfanumerico i campi conterranno dati inesatti. Per leggere i nomi degli utensili sempre in modo corretto sia per Serie10 che per OPENControl utilizzare la funzione GetToolNames_C.

Vedi anche

GetToolNames_C

Struttura GETINFO1DATA_C4 nel paragrafo "Descrizione delle Strutture"



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

GetNcInfo2_C

Legge alcuni parametri relativi al processo

```
WORD GetNcInfo2_C (  
    WORD UserSession,  
    WORD ProcNum,  
    GETINFO2DATA_C4 * pGetInfo,  
    DWORD * pErrClass,  
    DWORD * pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

ProcNum

[in] Identificatore del processo (numero da 1 a 24).

pGetInfo

[out] Puntatore alla struttura di tipo GETINFO2DATA_C4 che conterra' i dati

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Vedi anche

Struttura GETINFO2DATA_C4 nel paragrafo "Descrizione delle Strutture"

GetNcInfo1_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

GetMarkerInfo_C

Legge alcuni parametri relativi a parametri esecutivi del part program in esecuzione nel processo

```
WORD GetMarkerInfo_C (  
    WORD UserSession,  
    WORD ProcNum,  
    BYTE Request,  
    MARKER_INFO_C4 ptMarkerInfo,  
    DWORD * pErrClass,  
    DWORD * pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

ProcNum

[in] Identificatore del processo (numero da 1 a 24).

Request

[in] flag che specifica se le informazioni lette si riferiscano al punto più vicino (**1**) o all'ente attualmente in esecuzione (**0**).

ptMarkerInfo

[out] Puntatore alla struttura di tipo MARKER_INFO_C4 che conterrà i dati

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Vedi anche

Struttura MARKER_INFO_C4 nel paragrafo "Descrizione delle Strutture"



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

GetToolNames_C

Legge i nomi delle Tool in uso e programmata.

```
WORD GetNcInfo2_C (  
    WORD        UserSession,  
    WORD        ProcNum,  
    LPSTR       pTool,  
    LPSTR       pProgTool,  
    DWORD *     pErrClass,  
    DWORD *     pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

ProcNum

[in] Identificatore del processo (numero da 1 a 24).

pTool

[out] Puntatore ad una stringa di caratteri che conterrà il nome dell'utensile in uso.
La stringa deve essere composta di almeno 34 caratteri.

pProgTool

[out] Puntatore ad una stringa di caratteri che conterrà il nome dell'utensile programmato.
La stringa deve essere composta di almeno 34 caratteri.

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione è stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

GetProcessStatus_C

Restituisce informazioni circa il modo e lo stato del processo

```
WORD GetProcessStatus_C (  
    WORD        UserSession,  
    WORD        ProcNum,  
    PROCDATA_C4 * pNewMode,  
    DWORD *     pErrClass,  
    DWORD *     pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

ProcNum

[in] Identificatore del processo (numero da 1 a 24).

pNewMode

[out] Puntatore alla struttura di tipo PROCDATA_C4 che conterrà i dati

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione è stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Vedi anche

Struttura PROCDATA_C4 nel paragrafo "Descrizione delle Strutture"



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

GetBlkNum_C

Legge il numero di part program attivi (main program e relative subroutines) ed il blocco di part program attivo (in esecuzione) sul CNC remoto. Per leggere numeri di blocco validi i part program devono essere numerati.

```
WORD GetBlkNum_C (  
    WORD UserSession,  
    WORD ProcNum,  
    GETBLKNUMDATA_C4 * lpGetBlkNum,  
    DWORD * pErrClass,  
    DWORD * pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

ProcNum

[in] Identificatore del processo (numero da 1 a 24).

lpGetBlkNum

[out] Puntatore alla struttura di tipo GETBLKNUMDATA_C4 che conterrà i dati.

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione è stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Vedi anche

Struttura GETBLKNUMDATA_C4 nel paragrafo "Descrizione delle Strutture"



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

GetVarE_C (funzione obsoleta)

Funzione obsoleta. E' disponibile in questa libreria per compatibilità con applicazioni precedenti. Al suo posto si può usare la funzione ReadVarDouble_C.

Legge il valore delle variabili di tipo E del processo

```
WORD GetVarE_C (
    WORD      UserSession,
    WORD      ProcNum,
    WORD      Index,
    WORD *    pNum,
    double *  pValue,
    DWORD *   pErrClass,
    DWORD *   pErrNum
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

ProcNum

[in] Identificatore del processo (numero da 1 a 24).

Index

[in] Indice della prima variabile da leggere

pNum

[in,out] Puntatore ad una variabile:

[In] inizializzata con il numero di variabili che si vogliono leggere (MAX 22).

[Out] che conterra' il numero di variabili effettivamente lette.

pValue

[out] Puntatore all'array di variabili double che conterra' i valori delle variabili lette.

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Vedi anche

SetVarE_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

SetVarE_C (funzione obsoleta)

Funzione obsoleta. E' disponibile in questa libreria per compatibilità con applicazioni precedenti. Al suo posto si può usare la funzione WriteVarDouble_C.

Modifica il valore delle variabili di tipo E del processo.

```
WORD SetVarE_C (  
    WORD      UserSession,  
    WORD      ProcNum,  
    WORD      Index,  
    WORD      Num,  
    double *   pValue,  
    DWORD *    pErrClass,  
    DWORD *    pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

ProcNum

[in] Identificatore del processo (numero da 1 a 24).

Index

[in] Indice della prima variabile da modificare

Num

[in] Numero di variabili da modificare (MAX 22)

pValue

[in] Puntatore ad un array che contiene i dati da assegnare alle variabili. L'array deve contenere un numero di elementi pari a Num

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Vedi anche

GetVarE_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

GetVarSN_C (funzione obsoleta)

Funzione obsoleta. E' disponibile in questa libreria per compatibilità con applicazioni precedenti. Al suo posto si può usare la funzione ReadVarDouble_C.

Legge il valore delle variabili di tipo SN del processo

```
WORD GetVarSN_C (  
    WORD UserSession,  
    WORD ProcNum,  
    WORD Index,  
    WORD * pNum,  
    double * pValue,  
    DWORD * pErrClass,  
    DWORD * pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

ProcNum

[in] Identificatore del processo (numero da 1 a 24).

Index

[in] Indice della prima variabile da leggere

pNum

[in,out] Puntatore ad una variabile:

[In] inizializzata con il numero di variabili che si vogliono leggere (MAX 16).

[Out] che conterra' il numero di variabili effettivamente lette.

pValue

[out] Puntatore all'array che conterra' i valori delle variabili lette.

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Vedi anche

SetVarSN_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

SetVarSN_C (funzione obsoleta)

Funzione obsoleta. E' disponibile in questa libreria per compatibilità con applicazioni precedenti. Al suo posto si può usare la funzione WriteVarDouble_C.

Setta il valore delle variabili di tipo SN del processo

```
WORD SetVarSN_C (  
    WORD      UserSession,  
    WORD      ProcNum,  
    WORD      Index,  
    WORD      Num,  
    double *   pValue,  
    DWORD *    pErrClass,  
    DWORD *    pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

ProcNum

[in] Identificatore del processo (numero da 1 a 24).

Index

[in] Indice della prima variabile da modificare.

Num

[in] Numero di variabili da modificare (MAX 16)

pValue

[in] Puntatore ad un array che contiene i dati da assegnare alle variabili. L'array deve contenere un numero di elementi pari a Num

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Vedi anche

GetVarSN_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

GetVarSC_C (funzione obsoleta)

Funzione obsoleta. E' disponibile in questa libreria per compatibilità con applicazioni precedenti. Al suo posto si può usare la funzione ReadVarText_C.

Legge il valore delle variabili di tipo SC del processo

```
WORD GetVarSC_C (  
    WORD UserSession,  
    WORD ProcNum,  
    WORD Index,  
    WORD * pNum,  
    BYTE * pValue,  
    DWORD * pErrClass,  
    DWORD * pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

ProcNum

[in] Identificatore del processo (numero da 1 a 24).

Index

[in] Indice della prima variabile da leggere

pNum

[in,out] Puntatore ad una variabile:

[In] inizializzata con il numero di variabili che si vogliono leggere (MAX).

[Out] che conterra' il numero di variabili effettivamente lette.

pValue

[out] Puntatore all'array che conterra' i valori delle variabili lette.

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Vedi anche

SetVarSC_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

SetVarSC_C (funzione obsoleta)

Funzione obsoleta. E' disponibile in questa libreria per compatibilità con applicazioni precedenti. Al suo posto si può usare la funzione WriteVarText_C.

Modifica il valore delle variabili di tipo SC del sistema

```
WORD SetVarSC_C (  
    WORD UserSession,  
    WORD ProcNum,  
    WORD Index,  
    WORD Num,  
    BYTE * pValue,  
    DWORD * pErrClass,  
    DWORD * pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

ProcNum

[in] Identificatore del processo (numero da 1 a 24).

Index

[in] Indice della prima variabile da leggere.

Num

[in] Numero di variabili da leggere.

pValue

[in] Puntatore ad un array che contiene i dati da assegnare alle variabili. L'array deve contenere un numero di elementi pari a Num.

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Vedi anche

GetVarSC_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

ReadErrMsg_C

Legge i messaggi di errore di processo o di sistema. I messaggi sono nella lingua installata sul CNC.

```
WORD ReadErrMsg_C (  
    WORD UserSession,  
    WORD ProcNum,  
    ERR_MSG_C4 * pSysErrMsg,  
    DWORD * pErrClass,  
    DWORD * pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

ProcNum

[in] Identificatore del processo (numero da 1 a 24).

pErrMsg

[out] Puntatore ad una struttura che viene compilata con il testo relativo all' eventuale errore di processo o all' eventuale errore di sistema (prioritario rispetto a quello di processo).

La struttura ERR_MSG_C4 contiene 4 array di 40 caratteri ciascuno. IL CNC restituisce il testo negli array così come è visualizzato sullo schermo cioè in ogni array una riga di testo (da 1 a 4).

Gli array non utilizzati sono azzerati.

Le stringhe di testo, quando presenti, NON hanno terminatore cioè il testo di una linea occupa sempre i 40 caratteri del singolo array.

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione è stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Vedi anche

Struttura ERR_MSG_C4 nel paragrafo "Descrizione delle Strutture"



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

ReadPartProgramMsg_C

Legge il testo relativo ai messaggi da part program (istruzione DIS)

```
WORD ReadPartProgramMsg_C (  
    WORD        UserSession,  
    WORD        ProcNum,  
    LPSTR        pPartProgramMsg,  
    DWORD *      pErrClass,  
    DWORD *      pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

ProcNum

[in] Identificatore del processo (numero da 1 a 24).

pPartProgramMsg

[out] Puntatore ad una stringa che conterra' il testo del messaggio terminata con /0 (zero). La stringa deve essere preallocata e lunga almeno 130 caratteri.

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione è stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

ReadCurrentErrorMsg_C

Compatibilità

OPENcontrol

Legge il messaggio di errore di processo attivo.

```
WORD ReadCurrentErrorMsg_C (  
    WORD        UserSession,  
    WORD        Process,  
    MSG_ERROR_C4* pData,  
    DWORD*      pErrClass,  
    DWORD*      pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

Process

[in] Identificatore del processo (numero da 1 a 24).

pData

[out] Puntatore ad una struttura che viene compilata con le informazioni relative all' eventuale errore di processo.

NOTA: Il campo FormatTxt non contiene testo tradotto bensì le informazioni necessarie al software OPENcontrol per generare la traduzione in lingua del messaggio.

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione è stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Vedi anche

Struttura MSG_ERROR_C4 nel paragrafo "Descrizione delle Strutture"



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

ReadCurrentEmergMsg_C

Compatibilità

OPENcontrol

Legge il messaggio di emergenza.

```
WORD ReadCurrentEmergMsg_C (  
    WORD UserSession,  
    WORD Process,  
    MSG_EMERGENCY_C4* pData,  
    DWORD* pErrClass,  
    DWORD* pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

Process

[in] Identificatore del processo (numero da 1 a 24).

pData

[out] Puntatore ad una struttura che viene compilata con le informazioni relative all' eventuale emergenza.

NOTA: Il campo FormatTxt non contiene testo tradotto bensì le informazioni necessarie al software OPENcontrol per generare la traduzione in lingua del messaggio.

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione è stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Vedi anche

Struttura MSG_EMERGENCY_C4 nel paragrafo "Descrizione delle Strutture"



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

ReadCurrentAnomalyMsg_C

Compatibilità

OPENcontrol

Legge il messaggio di system anomaly.

```
WORD ReadCurrentAnomalyMsg_C (  
    WORD UserSession,  
    MSG_ANOMALY_C4* pData,  
    DWORD* pErrClass,  
    DWORD* pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

pData

[out] Puntatore ad una struttura che viene compilata con le informazioni relative all' eventuale system anomaly.

NOTA: Il campo FormatTxt non contiene testo tradotto bensì le informazioni necessarie al software OPENcontrol per generare la traduzione in lingua del messaggio.

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione è stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Vedi anche

Struttura MSG_ANOMALY_C4 nel paragrafo "Descrizione delle Strutture"



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

GetGCode_C

Restituisce lo stato di attivazione delle funzioni G del processo.

```
WORD GetGCode_C (  
    WORD      UserSession,  
    WORD      ProcNum,  
    WORD *    pGCode,  
    DWORD *   pErrClass,  
    DWORD *   pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

ProcNum

[in] Identificatore del processo (numero da 1 a 24).

pGCode

[out] Puntatore ad un array di 14 WORD che conterrà lo stato delle funzioni G

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

GetMCode_C

Restituisce la situazione delle M attive del processo

```
WORD GetMCode_C (  
    WORD        UserSession,  
    WORD        ProcNum,  
    WORD *      pMCode,  
    DWORD *     pErrClass,  
    DWORD *     pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

ProcNum

[in] Identificatore del processo (numero da 1 a 24).

pMCode

[out] Puntatore ad un array di 16 WORD che conterrà le M attive

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Vedi anche

GetNcInfo2_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

SkipPProgBlock_C

Sposta il cursore del programma per fare in modo che uno o più blocchi di programma non siano eseguiti. Il CNC accetta questo comando solo se c'è un programma selezionato ma non in esecuzione (CNC non in RUN), ed è impostato il modo semi-automatico.

```
WORD SkipPProgBlock_C (  
    WORD      UserSession,  
    WORD      ProcNum,  
    WORD      NumOfBlocks,  
    WORD      Direction,  
    WORD *    pMCode,  
    DWORD *   pErrClass,  
    DWORD *   pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

ProcNum

[in] Identificatore del processo (numero da 1 a 24).

NumOfBlocks

[in] Numero di blocchi di programma da saltare per posizionare il cursore del programma.

Direction

[in] Direzione di spostamento del cursore del programma. 0 avanti (verso fine programma), 1 indietro (verso inizio programma).

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione è stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Vedi anche

GetNcInfo2_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

Ese_C

Richiede l'esecuzione di un programma fino alla linea indicata.

```
WORD GetMCode_C (  
    WORD      UserSession,  
    WORD      ProcNum,  
    DWORD     BlockNum,  
    DWORD *   pErrClass,  
    DWORD *   pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

ProcNum

[in] Identificatore del processo (numero da 1 a 24).

BlockNum

[in] Numero del blocco di programma dove deve terminare l'esecuzione del programma. Il programma deve essere numerato o deve almeno esistere il numero di blocco della linea di programma da raggiungere.

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

EseEx_C

Compatibilità

OPENcontrol

Richiede l'esecuzione del programma fino al verificarsi della condizione di arresto. Estende la funzione Ese_C consentendo l'utilizzo di condizioni di arresto multiple.

```
WORD EseEx_C (  
    WORD        UserSession,  
    WORD        ProcNum,  
    RCM_TO_MODE Mode,  
    DWORD       Mask,  
    DWORD       BlockNum,  
    DWORD       BrkVal,  
    LPWSTR      Label,  
    DWORD       LineNum,  
    DWORD*      pErrClass,  
    DWORD*      pErrNum  
);
```

Parameters

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

ProcNum

[in] Identificatore del processo (numero da 1 a 24).

Mode

[in] modalità di valutazione della condizione di arresto:

RCM_TO: il part program si ferma non appena una delle condizioni di arresto specificate nel parametro Mask diventa vera. L'esecuzione riprende dalla linea successiva a quella di arresto (vedi "Osservazioni").

RCM_NEXT: Il part program si ferma al blocco, linea o label specificata. L'esecuzione riprende dal successivo blocco, linea, o label (vedi "Osservazioni").

Mask

[in] Maschera di bit che seleziona le condizioni di arresto da applicare. Solamente le condizioni il cui bit è a 1 vengono prese in considerazione.

Bit	Condizione
0	Numero di blocco
1	Numero di linea
2	Label
3	Valore di \$BRK

BlockNum

[in] Numero del blocco di programma dove deve terminare l'esecuzione del programma. Il programma deve essere numerato o deve almeno esistere il numero di blocco della linea di programma da raggiungere.



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

Label

[in] Puntatore ad un array C di wide characters (WCHAR) contenente il nome della label in corrispondenza della quale l'esecuzione del part program deve fermarsi. Il valore di Label non può essere NULL.

LineNum

[in] numero della linea in corrispondenza della quale l'esecuzione del part program deve fermarsi.

BrkVal

[in] Valore della variabile di processo \$BRK. L'esecuzione del part program si ferma non appena \$BRK è uguale al valore specificato.

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Osservazioni

In modalità RCM_TO, l'esecuzione del part program si ferma non appena si verifica una delle condizioni di arresto.

L'esecuzione riprende quando il processo riceve un comando di CYCLE, ripartendo dalla prima linea valida dopo il punto di arresto.

In modalità RCM_NEXT è possibile abilitare solamente una delle condizioni di arresto. L'esecuzione del part program si ferma non appena la condizione di arresto si verifica.

L'esecuzione riprende quando il processo riceve un comando di CYCLE, riavviandosi come segue:

Con una condizione di tipo "blocco", l'esecuzione riprende dal blocco successivo al punto di arresto, ovvero dalla successiva linea che inizia con un numero di sequenza (Nxx).

Con una condizione di tipo "linea", l'esecuzione riprende dalla prima linea valida dopo il punto di arresto.

Con una condizione di tipo "label", l'esecuzione riprende dalla prima label dopo il punto di arresto.

Con una condizione di tipo "valore di \$BRK", l'esecuzione riprende dalla prima linea in cui il valore di \$BRK viene modificato assegnando un valore che invalida la condizione di arresto.

Valore di ritorno

Diverso da zero se la funzione è stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

AxesRef_C

Forza la condizione di asse riferito.

ATTENZIONE: questa funzione dichiara riferito l'asse ovunque si trovi (anche se non e' sul micro di zero), muovendo l'asse si rischia di portarlo in oltrecorsa o sugli ingombri e danneggiare la macchina utensile.

```
WORD AxesRef_C (  
    WORD      UserSession,  
    WORD      ProcNum,  
    WORD      NumAx,  
    LPSTR      pAxisName,  
    DWORD *    pErrClass,  
    DWORD *    pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

ProcNum

[in] Identificatore del processo (numero da 1 a 24).

pAxisName

[in] Puntatore ad una stringa che contiene l'elenco dei nomi degli assi da riferire terminata con /0 (zero).

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

SetProcVarWord_C

Permette di scrivere delle variabili di tipo WORD appartenenti al processo.

Alcune delle variabili attivano o disattivano funzioni e modalità del controllo e non sono normalmente modificabili da part program e possono richiedere che il processo si trovi in stati particolari affinché la scrittura vada a buon fine.

```
WORD SetProcVarWord_C (  
    WORD UserSession,  
    WORD ProcNum,  
    PROC_WORD_VAR_TYPE_C VarType,  
    WORD Value,  
    DWORD * pErrClass,  
    DWORD * pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

ProcNum

[in] Identificatore del processo (numero da 1 a 24).

VarType

[in] Identificatore della variabile processo che si desidera scrivere. Gli identificatori disponibili sono:

Identificatore	Variabile	Descrizione
1	RCM	Attivazione ricerca memorizzata
2	ARM	Modalità normalizzazione movimenti circolari
3	DPS	Disabilitazione scroll part program
4	RAP	Attivazione ritorno automatico sul profilo
5	TRP	Percentuale velocità di ritorno in maschiatura
6	VFF	Valore percentuale del Velocity Feed Forward
7	UVR	Usa velocità di rapido
8	DLA	Disabilitazione della look-ahead
9	MBR	Abilitazione Multi Block Retrace
10	URL	Usa override per velocità di rapido
11	DSB	Disabilitazione dei blocchi barrati
12	USO	Abilitazione stop opzionale su codici M
13	ERR	Abilitazione gestione errori da part program
14	HMP	Abilitazione multi-homing
15	UAS	Attivazione modalità assi sconnessi
16	MBA	Abilitazione emissione funzioni ausiliarie in MBR
17	REM	Modalità ritorno sul profilo
18	CYCTIME	Abilitazione modalità stima tempi

Value

[in] Valore che dovrà assumere la variabile



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione è stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

GetProcVarWord_C

Permette di leggere alcune delle variabili di tipo WORD appartenenti al processo.

```
WORD GetProcVarWord_C (  
    WORD UserSession,  
    WORD ProcNum,  
    PROC_WORD_VAR_TYPE_C VarType,  
    WORD * pValue,  
    DWORD * pErrClass,  
    DWORD * pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

ProcNum

[in] Identificatore del processo (numero da 1 a 24).

VarType

[in] Identificatore della variabile processo che si desidera leggere. Gli identificatori disponibili sono:

Identificatore	Variabile	Descrizione
1	RCM	Attivazione ricerca memorizzata
2	ARM	Modalità normalizzazione movimenti circolari
3	DPS	Disabilitazione scroll part program
4	RAP	Attivazione ritorno automatico sul profilo
5	TRP	Percentuale velocità di ritorno in maschiatura
6	VFF	Valore percentuale del Velocity Feed Forward
7	UVR	Usa velocità di rapido
8	DLA	Disabilitazione della look-ahead
9	MBR	Abilitazione Multi Block Retrace
10	URL	Usa override per velocità di rapido
11	DSB	Disabilitazione dei blocchi barrati
12	USO	Abilitazione stop opzionale su codici M
13	ERR	Abilitazione gestione errori da part program
14	HMP	Abilitazione multi-homing
15	UAS	Attivazione modalità assi sconnessi
16	MBA	Abilitazione emissione funzioni ausiliarie in MBR
17	REM	Modalità ritorno sul profilo
18	CYCTIME	Abilitazione modalità stima tempi

pValue

[out] Puntatore a una variabile di formato WORD (16 bit) che conterrà il valore della variabile letta.

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione è stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

Funzioni per la “Modalità passante”

Le funzioni di questo capitolo sono RISERVATE.

Queste funzioni possono essere chiamate solo dopo aver creato il server Cndex (vedi ConnectServer_C), aperto una sessione di comunicazione (vedi OpenSession_C) ed aver verificato che il CNC si trovi nella modalità RUN (vedi BootPhaseEnquiry_C).

Tramite la “Modalità Passante” e’ possibile eseguire un Part Program inviato run-time da un applicativo esterno (invece che da file memorizzato su CN).

Puo’ essere gestito tramite la Modalità passante solo il programma **MAIN**; le subroutines **NON** possono essere scaricate run-time. Il main puo’ invece richiamare delle subroutines residenti sul CN in formato file.

La dimensione massima di un programma “Passante”, cioe’ la quantita’ massima di dati trasferiti, e’ di 2 Gbyte.

Il CN memorizza i dati ricevuti via rete in un buffer circolare di 65000 caratteri; quest’area è da intendere come una “finestra” aperta sull’intero programma “Passante”. Il buffer circolare viene scaricato dal CN durante l’esecuzione delle istruzioni in esso contenute in modo da potere caricare ulteriori dati; i dati scaricati (perché relativi a istruzioni già eseguite) non sono più accessibili.

Durante l’esecuzione del programma Passante, il raggiungimento della fine dell’area attiva (ovvero il suo completo svuotamento) provocherà l’attesa di ulteriori dati a meno che non sia stata già segnalata la “fine” del programma, in questo caso il CN termina l’esecuzione del programma.

Una volta terminata l’esecuzione (oppure quando il CN è in IDLE) sarà possibile chiudere la Modalità passante.

Il comando di Reset non chiude la modalità passante, bensì svuota l’area attiva e riposiziona ad inizio programma.



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

DncInit_C

Funzione riservata.

Attiva la Modalità passante.

```
WORD DncInit_C (  
    WORD        UserSession,  
    WORD        ProcNum,  
    LPSTR       pPathName ,  
    DWORD *     pErrClass,  
    DWORD *     pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

ProcNum

[in] Identificatore del processo (numero da 1 a 24).

pPathName

[in] Puntatore alla stringa che contiene il nome da visualizzare come Programma attivo

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Ulteriori codici di errore:

Classe 4 – Errore 160 Comando non disponibile nello stato attuale.



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

DncData_C

Funzione riservata.

Invia le linee di Part Program da eseguire in Modalità Passante.

```
WORD DncData_C (  
    WORD      UserSession,  
    WORD      ProcNum,  
    LPSTR      pData,  
    DWORD *    pErrClass,  
    DWORD *    pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

ProcNum

[in] Identificatore del processo (numero da 1 a 24).

pData

[in] Puntatore all'area che contiene le righe di programma da eseguire (MAX 178 caratteri). I dati devono essere terminati con il carattere /0 (zero). In un blocco dati si possono specificare piu' linee di part program separate con i caratteri CR-LF (Carriage Return – Line Feed).

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Ulteriori codici di errore:

Classe 4 – Errore 11 Funzione non permessa (modalità non attiva)

Classe 4 – Errore 10 Memoria piena

Classe 4 – Errore 65 Programma passante troppo grande (2Gbyte)



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

DncEof_C

Funzione riservata.

Informa il CN che le linee di Part Program sono state tutte trasferite (fine Part Program).

```
WORD DncEof_C (  
    WORD        UserSession,  
    WORD        ProcNum,  
    DWORD *     pErrClass,  
    DWORD *     pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

ProcNum

[in] Identificatore del processo (numero da 1 a 24).

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

DncStop_C

Funzione riservata.

Disattiva la Modalità passante.

```
WORD DncStop_C (  
    WORD UserSession,  
    WORD ProcNum,  
    DWORD * pErrClass,  
    DWORD * pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

ProcNum

[in] Identificatore del processo (numero da 1 a 24).

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione è stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Ulteriori codici di errore:

Classe 4 – Errore 160 Comando non disponibile nello stato attuale.



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

Funzioni generiche

Queste funzioni possono essere chiamate solo dopo aver creato il server Cndex (vedi ConnectServer_C), aperto una sessione di comunicazione (vedi OpenSession_C) ed aver verificato che il CNC si trovi nella modalita' RUN (vedi BootPhaseEnquiry_C).



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

GetAxesInfo3_C

Legge il numero di processo, il nome asse, il tipo asse ed il tipo di interfaccia per tutti gli assi configurati.

```
WORD GetAxesInfo3_C (  
    WORD        UserSession,  
    WORD        AxisID,  
    WORD        *pAxesNumber,  
    WORD        *pAxOwnerList,  
    BYTE        *pAxNameList,  
    WORD        *pAxType,  
    WORD        *pAxInterface,  
    DWORD * pErrClass,  
    DWORD * pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

AxisID

[in] ID dell'asse di cui si richiedono le informazioni. Inserire 0xFFFF per avere le informazioni su tutti gli assi. In OPENControl questo parametro viene ignorato.

pAxesNumber

[in] Dimensione dei buffer assi. Deve essere almeno 32 per Serie10 e 64 per OPENControl.

[out] Numero di assi presenti.

pAxOwnerList

[out] Vettore di WORD contenente l'ID del processo owner di ogni asse presente. L'i-esimo elemento del vettore contiene l'ID del processo owner dell'asse con ID i + 1. L'ID di processo 0 identifica la logica di macchina. Se l'asse non è presente il valore è 0xFFFF.

pAxNameList

[out] Vettore di BYTE contenente il codice ASCII corrispondente al nome di ogni asse presente. L'i-esimo elemento del vettore contiene il nome dell'asse con ID i + 1. Se l'asse non è presente il valore è 0.

pAxType

[out] Vettore di WORD contenente la maschera di bit che definisce il tipo asse di ogni asse presente. L'i-esimo elemento del vettore contiene il tipo asse dell'asse con ID i + 1. Se l'asse non è presente il valore è 0.

pAxInterface

[out] Vettore di WORD contenente il tipo di interfaccia di ogni asse presente. L'i-esimo elemento del vettore contiene il tipo di interfaccia dell'asse con ID i + 1. Se l'asse non è presente il valore è 0.



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

GetCodeNumber_C

Legge la matricola del CNC e l'identificatore della versione software installata.

```
WORD GetCodeNumber_C (  
    WORD        UserSession,  
    LPSTR       pCodeNumber,  
    LPSTR       pSwVersion,  
    DWORD *     pErrClass,  
    DWORD *     pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

pWarningMsg

[out] Puntatore ad una stringa di almeno 20 caratteri in cui verra' scritta la matricola del controllo numerico.

pSwVersion

[out] Puntatore ad una stringa di almeno 20 caratteri in cui verra' scritta la versione del software installato sul controllo numerico.

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

GetOptions_C

Legge la mappa dei flag delle opzioni abilitate, la mappa dei flag delle abilitazioni e l'identificatore della versione software installata.

```
WORD GetOptions_C (  
    WORD UserSession,  
    BYTE * pOption,  
    BYTE * pSecurLevel,  
    LPSTR pSwVersion,  
    DWORD * pErrClass,  
    DWORD * pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

pOption

[out] Puntatore ad una stringa di almeno 19 byte in cui sarà scritta la mappa delle opzioni abilitate sul CNC. Per l'utilizzo di questo array riservato e' necessario richiedere ad OSAI la documentazione di dettaglio della mappa.

pSecurLevel

[out] Puntatore ad una stringa di almeno 12 byte in cui sarà scritta la mappa delle funzioni abilitate sul CNC. I primi due byte indicano il livello selezionato tramite l'applicazione Security. I restanti 10 byte sono la mappa dei flag impostati tramite Security per il livello selezionato.

pSwVersion

[out] Puntatore ad una stringa di almeno 20 caratteri in cui sarà scritta la versione del software installato sul controllo numerico.

pErrClass

[out] Puntatore alla variabile dove sarà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove sarà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

GetDateTime_C

Richiede la data e l'ora impostata sul CNC.

```
WORD GetDateTime_C (  
    WORD UserSession,  
    WORD * pYear,  
    WORD * pMonth,  
    WORD * pDay,  
    WORD * pHour,  
    WORD * pMinute,  
    WORD * pSecond  
    DWORD * pErrClass,  
    DWORD * pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

pYear

[out] Puntatore alla variabile dove verrà scritto l'anno.

pMonth

[out] Puntatore alla variabile dove verrà scritto il mese.

pDay

[out] Puntatore alla variabile dove verrà scritto il giorno.

pHour

[out] Puntatore alla variabile dove verrà scritto l'ora.

pMinute

[out] Puntatore alla variabile dove verranno scritti i minuti.

pSecond

[out] Puntatore alla variabile dove verranno scritti i secondi.

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione è stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

Vedi anche
[SetDateTime_C](#)



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

SetDateTime_C

Modifica la data e l'ora del CNC.

```
WORD GetDateTime_C (  
    WORD          UserSession,  
    WORD          Year,  
    WORD          Month,  
    WORD          Day,  
    WORD          Hour,  
    WORD          Minute,  
    WORD          Second  
    DWORD * pErrClass,  
    DWORD * pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

Year

[in] Anno che verrà impostato sul CNC.

Month

[in] Mese che verrà impostato sul CNC.

Day

[in] Giorno che verrà impostato sul CNC.

Hour

[in] Ora che verrà impostato sul CNC.

Minute

[in] Minuti che verranno impostati sul CNC.

Second

[in] Secondi che verranno impostati sul CNC.

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione è stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

Vedi anche
[GetDateTime_C](#)



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

GRead_C

Legge un area di memoria condivisa del CNC. Le aree di memoria condivise del sistema si indirizzano tramite il loro identificatore. Per utilizzare questa funzione e' necessario conoscere l'identificativo dell'area di sistema che si intende leggere e la sua struttura. Queste informazioni sono distribuite da OSAI solamente su richiesta.

```
WORD GRead_C (  
    WORD        UserSession,  
    WORD        MemId,  
    WORD        Offset,  
    WORD        Length,  
    byte *      pBuffer,  
    WORD *      pBufferLength,  
    DWORD *     pErrClass,  
    DWORD *     pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

MemId

[in] Identificativo della memoria da leggere

Offset

[in] Offset a partire dal quale viene letta la memoria

Length

[in] Puntatore ad una variabile che determina la lunghezza dei dati da leggere.

pBuffer

[in] Puntatore al buffer dati

pBufferLength

[in] Lunghezza del buffer dati

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

Vedi anche
GWrite_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

GWrite_C

Scrive un area di memoria condivisa del CNC. Le aree di memoria condivise del sistema si indirizzano tramite il loro identificatore. Per utilizzare questa funzione e' necessario conoscere l'identificativo dell'area di sistema che si intende scrivere e la sua struttura. Queste informazioni sono distribuite da OSAI solamente su richiesta.

```
WORD GWrite_C (  
    WORD        UserSession,  
    WORD        MemId,  
    WORD        Offset,  
    WORD        Length,  
    byte *      pBuffer,  
    WORD *      pBufferLength,  
    DWORD *     pErrClass,  
    DWORD *     pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

MemId

[in] Identificativo della memoria da scrivere

Offset

[in] Offset a partire dal quale viene scritta la memoria

pLength

[in, out] Puntatore ad una variabile che:

[inp] determina la lunghezza dei dati da scrivere.

[out] informa del numero di byte effettivamente scritti.

pBuffer

[in] Puntatore al buffer dati

BufferLength

[in] Lunghezza del buffer dati

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

Vedi anche
GRead_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

SetIpAddress_C

Permette di modificare l'indirizzo IP di un controllo numerico. Il sistema viene riavviato automaticamente dopo la modifica.

```
WORD SetIpAddress_C (  
    WORD        UserSession,  
    LPTSTR      IpAddress,  
    LPTSTR      SubNetMask,  
    LPTSTR      DefaultGateway,  
    LPTSTR      Dns,  
    LPTSTR      Wins,  
    LPTSTR      Net,  
    DWORD * pErrClass,  
    DWORD * pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

IpAddress

[in] Nuovo indirizzo IP del controllo.

SubNetMask

[in] Nuova subnetmask.

DefaultGateway

[in] Nuovo default gateway.

Dns

[in] Nuovo Dns.

Wins

[in] Nuovo Wins.

Net

[in] Identificativo della scheda di rete di cui si vuole cambiare l'indirizzo IP(0: Prima scheda
1:Seconda scheda).

pErrClass

[out] Puntatore alla variabile dove sarà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove sarà scritto il numero dell'eventuale errore.



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Ulteriori codici di errore:

Classe 17 Errore 265 Errore durante la modifica dell'indirizzo IP



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

GetSerialNumber_C

Legge la matricola del CNC.

```
WORD GetSerialNumber_C (
    WORD          UserSession,
    LPSTR          pSerialNumber,
    DWORD * pErrClass,
    DWORD * pErrNum
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

pSerialNumber

[out] Puntatore ad una stringa di almeno 20 caratteri in cui verra' scritta la matricola del controllo numerico.

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

ReadRemapDefinitions_C

Legge dal CN l'array delle definizioni di I/O remap

```
WORD ReadRemapDefinitions_C (  
    WORD                UserSession,  
    int                 ArraySize,  
    REMAP_DEF_C4 *      pArray,  
    int                 Type,  
    int *               pDefinitionCnt,  
    DWORD *             pErrClass,  
    DWORD *             pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

ArraySize

[in] Dimensione dell'array puntato da pArray

pArray

[out] Puntatore ad un array di strutture di tipo REMAP_DEF_C4 nelle quali saranno scritte le definizioni di I/O remap

Type

[in] se vale 0 la funzione legge le definizioni di remap degli input; se vale 1 la funzione legge le definizioni di remap degli output

pDefinitionCnt

[out] numero di definizioni lette dalla funzione, ovvero numero di elementi significativi dell'array.

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Vedi anche

WriteRemapDefinitions_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

WriteRemapDefinitions_C

Scrive sul CN l'array delle definizioni di I/O remap

```
WORD WriteRemapDefinitions_C (  
    WORD                UserSession,  
    int                 ArraySize,  
    REMAP_DEF_C4 *      pArray,  
    int                 Type,  
    int                 Mode,  
    DWORD *             pErrClass,  
    DWORD *             pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

ArraySize

[in] Dimensione dell'array puntato da pArray

pArray

[in] Puntatore ad un array di strutture di tipo REMAP_DEF_C4 che contengono le definizioni di I/O remap

Type

[in] se vale 0 la funzione scrive le definizioni di remap degli input; se vale 1 la funzione scrive le definizioni di remap degli output

Mode

[in] può assumere questi valori:

- 0: sovrascrive tutte le definizioni
- 1: aggiunge una definizione
- 2: rimuove una definizione

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Vedi anche

ReadRemapDefinitions_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

GetServoPar_C

Legge il valore del parametro di un asse

```
WORD GetServoPar_C (  
    WORD UserSession,  
    WORD AxisId,  
    WORD ParId,  
    double * pValue,  
    DWORD * pErrClass,  
    DWORD * pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

AxisId

[in] Id dell'asse

ParId

[in] Id del parametro di cui desidera leggere il valore

pValue

[out] Valore del parametro letto dalla funzione

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Vedi anche

SetServoPar_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

SetServoPar_C

Scrive il valore del parametro di un asse

```
WORD SetServoPar_C (  
    WORD UserSession,  
    WORD AxisId,  
    WORD ParId,  
    double Value,  
    DWORD * pErrClass,  
    DWORD * pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

AxisId

[in] Id dell'asse

ParId

[in] Id del parametro di cui desidera scrivere il valore

Value

[in] Valore del parametro da scrivere

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Vedi anche

GetServoPar_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

GetCNCRegKey_C

Legge una chiave del registry del CNC.

```
WORD GetCNCRegKey_C (  
    WORD        UserSession,  
    LPWSTR       Key,  
    LPWSTR       SubKey,  
    DWORD *      pValueLen,  
    BYTE *        pValue,  
    DWORD *      pErrClass,  
    DWORD *      pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

Key

[in] percorso della chiave da leggere, relativo a HKEY_LOCAL_MACHINE \Osai. Il percorso completo viene calcolato come segue:

HKEY_LOCAL_MACHINE\Osai\<Key>

Pertanto non è possibile leggere chiavi di registro che si trovano fuori dal percorso HKEY_LOCAL_MACHINE \Osai.

SubKey

[in] nome della sottochiave da leggere.

pValueLen

[in, out] puntatore ad un intero a 32 bit senza segno contenente la lunghezza del vettore cui punta pValue. All'uscita dalla chiamata a funzione l'intero contiene la lunghezza effettiva del valore letto. Se il vettore non è abbastanza grande da contenere il valore letto la funzione restituisce errore classe 17 numero 221.

pValue

[in] Puntatore ad un vettore di byte in cui verrà copiato il valore letto dal registry.

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

Funzioni per la gestione del file system

Le funzioni di questo paragrafo, permettono di gestire in modo uniforme il file system dei diversi sistemi OSAI (Serie10, OPENControl, SyncMotion, Top5, PC)

Il file system remoto dei sistemi OSAI basa il suo funzionamento sui **drive logici**. I drive logici sono delle directory identificate da un nome logico.

In tutte le funzioni di questo paragrafo dove è richiesto un path-name bisogna specificare un path-name riferito ad un drive logico.

E' possibile p.e. definire un drive logico con nome "Programs" che identifica la directory "F:\My-Programs" di un determinato CNC Serie10.

Le funzioni del file system che utilizzano un path-name faranno riferimento al drive logico "Programs" di quel CNC Serie10 e non all'indirizzo fisico "[F:\MyPrograms](#)".

Ipotizzando una sub-directory F:\MyPrograms\dir1 (riferita all'esempio precedente) che contiene il file Example.txt, il path-name da utilizzare nelle funzioni per indicare il file sarà Programs\dir1\Examples.txt cioè il nome drive seguito dalla sub-directory e dal nome file (notare che NON è specificata alcuna barra prima del nome drive logico).

I drive logici sono tipicamente configurati dall'OEM tramite l'applicazione FileBrowser della human interface WinNBI (si veda l'Help dell'applicazione FileBrowser per la definizione dei drive logici).

E' anche possibile gestire la configurazione dei drive logici tramite alcune funzioni descritte in questo paragrafo (vedi le funzioni LogFSAddDrive_C, LogFSRemoveDrive_C).

Esistono inoltre, su tutti i target OSAI (tranne i PC), dei drive logici predefiniti ma non visibili tramite la human interface WinNBI (FileBrowser ecc.). Questi drive sono definiti **Hidden logical drive** (drive logici nascosti) e possono essere utilizzati solamente via software tramite le funzioni di questo paragrafo.

La configurazione dei drive logici nascosti non si può modificare. I drive logici nascosti attualmente definiti sono i seguenti:

per i sistemi OPENControl, SyncMotion, Top5
"DEVICE" che indica la directory "\SSD"

per i sistemi S10 (in modalità di boot **RUN** oppure **EMERGENZA** o **SERVICE**)

"**SYS**" che indica la directory "C:\"

"**PRJ**" che indica la directory "D:\"

"**USR**" che indica la directory "E:\"

"**UPP**" che indica la directory "F:\"

Le funzioni possono essere chiamate solo dopo aver aperto una sessione di comunicazione (vedi OpenSession_C) ed aver verificato che il CNC si trovi nella modalità RUN, EMERGENZA o SERVICE (vedi BootPhaseEnquiry_C).

Fanno eccezione le due funzioni di trasferimento file (LogFSTransferFile_C e LogFSTransferFileW_C) che provvedono autonomamente alla apertura delle sessioni di comunicazione.



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

LogFSTransferFile_C

Copia un file. Il file può anche essere copiato in una directory appartenente ad un altro file system, ad es. ad un altro CNC. Se l'applicazione utilizza stringhe in formato Unicode è consigliabile utilizzare la funzione LogFSTransferFileW_C.

```
WORD LogFSTransferFile_C (  
    LPCSTR      SourceTargetName,  
    LPCSTR      SourceFilePath,  
    LPCSTR      DestTargetName,  
    LPCSTR      DestFilePath,  
    DWORD * pErrClass,  
    DWORD * pErrNum  
);
```

Parametri

SourceTargetName

[in] Nome di rete del Target dalla quale prelevare il file (per la sintassi del nome si veda il campo RemoteName della funzione OpenSession_C).

SourceFilePath

[in] Path-name completo del file da leggere.

DestTargetName

[in] Nome di rete del Target sul quale copiare il file (per la sintassi del nome si veda il campo RemoteName della funzione OpenSession_C).

DestFilePath

[in] Path-name completo della directory dove scrivere il file. Deve comprendere il nome file di destinazione.

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Vedi anche

LogFSTransferFileW_C, LogFSCopyFile_C



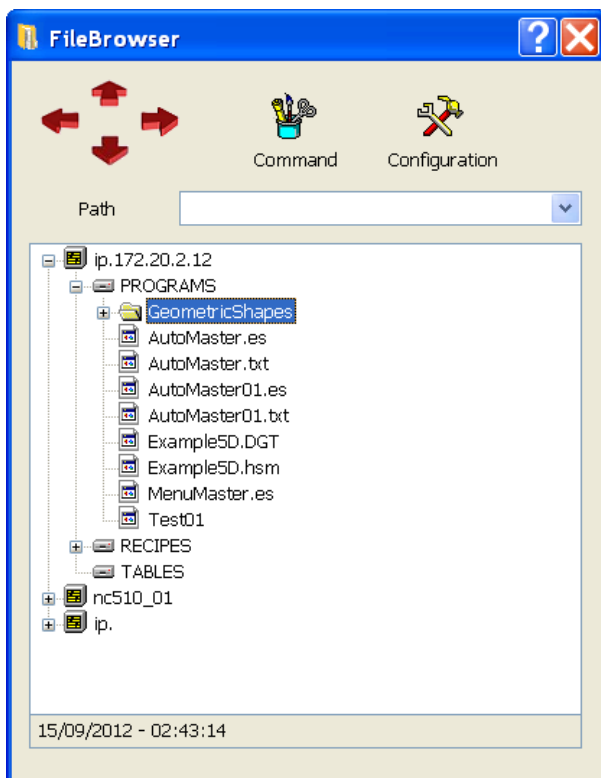
Documentazione CndexLinkUser DLL Interfaccia al server Cndex

```
//Questo code sample in C# copia il file di testo "square.txt" dalla directory
//locale "C:\Programs" al drive logico remoto "PROGRAMS" nella directory
// "GeometricShapes"

//Crea un nuovo oggetto Cndex
Cndex cndexobj = new Cndex();

//A differenza delle altre funzioni della libreria user, LogFSFileTransfer_C
//e LogFSFileTransferW_C non richiedono un numero di sessione.
if (cndexobj.LogFSTransferFileW_C(
    null, //null indica il sistema locale
    "C:\\Programs\\Square.txt", // path nel formato del file system locale
    "IP.172.20.2.12", //Indirizzo IP del sistema remoto
    //Il prefisso "IP." identifica un OPENcontrol o un PC
    "PROGRAMS\\GeometricShapes\\Square.txt", //Il path remoto deve usare la
    //sintassi <logical drive>\\<sottodirectory>.
    out m_ErrClass, out m_ErrNum) > 0)
    MessageBox.Show("File transfer successful.");
else
{
    String msg = string.Format("File transfer error {0}/{1:X}h",
        m_ErrClass, m_ErrNum);
    MessageBox.Show(msg);
}
```

Il drive logico deve essere configurato in File Browser o Process Controller come mostrato in figura. Per maggiori dettagli si consulti la documentazione del File Browser.





Documentazione CndexLinkUser DLL Interfaccia al server Cndex

LogFSTransferFileW_C

Questa funzione equivale a LogFSTransferFile_C, ma utilizza il formato Unicode (WCHAR) per i nomi dei target ed i percorsi all'interno del file system.

```
WORD LogFSTransferFileW_C (  
    LPCWSTR SourceTargetName,  
    LPCWSTR SourceFilePath,  
    LPCWSTR DestTargetName,  
    LPCWSTR DestFilePath,  
    DWORD * pErrClass,  
    DWORD * pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

SourceTargetName

[in] Nome di rete del Target dalla quale prelevare il file (per la sintassi del nome si veda il campo RemoteName della funzione OpenSession_C).

SourceFilePath

[in] Percorso completo del file da leggere.

DestTargetName

[in] Nome di rete del Target sul quale copiare il file (per la sintassi del nome si veda il campo RemoteName della funzione OpenSession_C).

DestFilePath

[in] Percorso completo della directory dove scrivere il file. Deve comprendere il nome file di destinazione.

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Vedi anche

LogFSTransferFile_C, LogFSCopyFile_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

LogFSSetSecurityLevel_C

Imposta il livello di sicurezza da utilizzare per il file system.

```
WORD LogFSSetSecurityLevel_C (  
    WORD UserSession,  
    LPCWSTR Password,  
    SECURITY_LEVEL_C SecurityLevel,  
    SECURITY_LEVEL_C * pOldSecurityLevel,  
    DWORD * pErrClass,  
    DWORD * pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

Password

[in] Password (OEM) che permette la modifica del livello di sicurezza attivo. Attualmente non è utilizzato, con le prossime release questo campo verrà attivato. Un applicativo che utilizza questa funzione dovrebbe richiedere all'utente il livello e la password da inserire nei rispettivi parametri.

SecurityLevel

[in] Livello di sicurezza da impostare. I livelli previsti sono i seguenti:

SECURITY_LEV_ADMIN	= 1
SECURITY_LEV_SERVICE	= 2
SECURITY_LEV_OEM_ADMIN	= 3
SECURITY_LEV_OEM_SERVICE	= 4
SECURITY_LEV_USER_ADMIN	= 5
SECURITY_LEV_USER_SERVICE	= 6

Il livello 1 e 2 sono riservati ad OSAI. Gli altri quattro livelli potranno essere utilizzati dallo OEM/end-user.

Ad oggi è necessario richiedere il livello SECURITY_LEV_OEM_ADMIN per poter inserire o rimuovere drive logici dalla configurazione del target.

pOldSecurityLevel

[out] Livello di sicurezza attivo prima della chiamata di questa funzione.

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione è stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Vedi anche

LogFSGetSecurityLevel_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

LogFSGetSecurityLevel_C

Legge il livello di sicurezza impostato per il file system.

```
WORD LogFSGetSecurityLevel_C (  
    WORD UserSession,  
    SECURITY_LEVEL_C * pSecurityLevel,  
    DWORD * pErrClass,  
    DWORD * pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

pSecurityLevel

[out] Puntatore alla variabile nella quale verrà scritto il livello di sicurezza attualmente impostato sul sistema. Vedi la funzione LogFSSetSecurityLevel_C per ulteriori dettagli.

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Vedi anche

LogFSSetSecurityLevel_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

LogFSLongFileNames_C

Questa funzione permette di verificare se il CNC gestisce in modo nativo i nomi lunghi per i file. I sistemi OPENControl e SyncMotion gestiscono i nomi lunghi in modo nativo mentre la Serie10 implementa la gestione dei nomi lunghi tramite le funzioni descritte nel paragrafo "Funzioni per la gestione dei nomi file lunghi".

```
WORD LogFSLongFileNames_C (  
    WORD          UserSession,  
    WORD *        pUseLongFileNames,  
    DWORD *       pErrClass,  
    DWORD *       pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

pUseLongFileNames

[out] Puntatore alla variabile nella quale verrà scritto il dato. Se il dato è 0 il sistema non gestisce i nomi lunghi. In questo caso bisognerà utilizzare nomi file con formato DOS.

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione è stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

LogFSGetNumDrive_C

Legge il numero di drive utente e drive nascosti attualmente attivi (vedi introduzione di questo paragrafo).

```
WORD LogFSGetNumDrive_C (  
    WORD UserSession,  
    WORD * pNumHiddenDrive,  
    WORD * pNumUserDrive,  
    DWORD * pErrClass,  
    DWORD * pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

pNumHiddenDrive

[out] Puntatore alla variabile dove verrà scritta il numeri di hidden logical drive (drive logic nascosti) attivi.

pNumUserDrive

[out] Puntatore alla variabile dove verrà scritta il numeri di user logical drive (drive logici utente) attivi.

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Vedi anche

LogFSGetHiddenDriveList_C, LogFSGetDriveList_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

LogFSGetDriveList_C

Legge, uno per volta, i nomi dei drive logici utente attivi (vedi introduzione di questo paragrafo).

```
WORD LogFSGetDriveList_C (  
    WORD        UserSession,  
    LPWSTR       DriveName,  
    WORD         index,  
    DWORD *      pErrClass,  
    DWORD *      pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

DriveName

[out] Puntatore alla variabile dove verrà scritto il nome del user logical drive.

index

[in] Indice del drive del quale si vuole leggere il nome. Non deve eguagliare o superare il numero di user logical drive letto con la funzione LogFSGetNumDrive_C.

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Vedi anche

LogFSGetHiddenDriveList_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

LogFSGetHiddenDriveList_C

Legge i nomi dei drive logici nascosti attivi (vedi introduzione di questo paragrafo).

```
WORD LogFSGetHiddenDriveList_C (  
    WORD        UserSession,  
    LPWSTR      DriveName,  
    WORD        index,  
    DWORD *     pErrClass,  
    DWORD *     pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

DriveName

[out] Puntatore alla variabile dove verrà scritto il nome del hidden logical drive.

index

[in] Indice del drive del quale si vuole leggere il nome. Non deve eguagliare o superare il numero di hidden logical drive letto con la funzione LogFSGetNumDrive_C.

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Vedi anche

LogFSGetDriveList_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

LogFSGetDrivePath_C

Legge il path-name fisico associato ad un drive configurato.

```
WORD LogFSGetDrivePath_C (  
    WORD UserSession,  
    LPCWSTR DriveName,  
    LPWSTR DrivePath,  
    SECURITY_LEVEL_C SecurityLevel,  
    DWORD * pErrClass,  
    DWORD * pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

DriveName

[in] Nome del drive logico del quale si richiede il path-name fisico configurato.

DrivePath

[out] Variabile dove verrà scritto il path-name fisico configurato per il drive logico specificato.

SecurityLevel

[in] Livello di sicurezza a cui fa riferimento la richiesta. Attualmente è necessario passare il valore SECURITY_LEV_OEM_ADMIN. Lo stesso livello deve essere stato precedentemente impostato come livello di sicurezza attivo (vedi la funzione LogFSSetSecurityLevel_C per ulteriori dettagli).

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Vedi anche

LogFSSetSecurityLevel_C.



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

LogFSAddDrive_C

Aggiunge un drive logico alla configurazione dei drive.

Il nuovo drive è disponibile solo dopo aver attivato la nuova configurazione tramite la funzione LogFSReloadDriveList_C.

La funzione restituisce errore se si richiede di aggiungere un drive logico su un target di tipo PC. I drive logici su PC possono essere configurati solo tramite l'applicazione OSAI FileBrowser (appartenente al prodotto WinNBI) attivata localmente al PC.

```
WORD LogFSAddDrive_C (  
    WORD                UserSession,  
    LPCWSTR             DriveName,  
    LPCWSTR             PathName,  
    BOOL                Temporary,  
    SECURITY_LEVEL_C     SecurityLevel,  
    DWORD *             pErrClass,  
    DWORD *             pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

DriveName

[in] Nome del drive logico da aggiungere nella configurazione.

PathName

[out] Path-name della directory fisica da associare al drive logico.

Temporary

[in] Parametro non utilizzato. Passare FALSE (0).

SecurityLevel

[in] Livello di sicurezza a cui fa riferimento la richiesta. Attualmente è necessario passare il valore SECURITY_LEV_OEM_ADMIN. Lo stesso livello deve essere stato precedentemente impostato come livello di sicurezza attivo (vedi la funzione LogFSSetSecurityLevel_C per ulteriori dettagli).

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione è stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Vedi anche



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

LogFSRemoveDrive_C, LogFSReloadDriveList_C, LogFSSetSecurityLevel



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

LogFSRemoveDrive_C

Rimuove un drive dalla configurazione dei drive.

Il drive viene rimosso dalla lista dei drive attivi solo dopo che è stata attivata la nuova configurazione tramite la funzione LogFSReloadDriveList_C.

```
WORD LogFSRemoveDrive_C (  
    WORD          UserSession,  
    LPCWSTR       DriveName,  
    SECURITY_LEVEL_C SecurityLevel,  
    DWORD *       pErrClass,  
    DWORD *       pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

DriveName

[in] Nome del drive logico da rimuovere dalla configurazione.

SecurityLevel

[in] Livello di sicurezza a cui fa riferimento la richiesta. Attualmente è necessario passare il valore SECURITY_LEV_OEM_ADMIN. Lo stesso livello deve essere stato precedentemente impostato come livello di sicurezza attivo (vedi la funzione LogFSSetSecurityLevel_C per ulteriori dettagli).

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione è stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Vedi anche

LogFSAddDrive_C, LogFSReloadDriveList_C, LogFSSetSecurityLevel



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

LogFSReloadDriveList_C

Attiva la configurazione modificata tramite le funzioni LogFSAddDrive_C e LogFSRemoveDrive_C.

```
WORD LogFSReloadDriveList_C (  
    WORD        UserSession,  
    DWORD * pErrClass,  
    DWORD * pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Vedi anche

LogFSAddDrive_C, LogFSRemoveDrive_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

LogFSCreateDir_C

Crea una nuova directory.

```
WORD LogFSCreateDir_C (  
    WORD          UserSession,  
    LPCWSTR PathName,  
    DWORD * pErrClass,  
    DWORD * pErrNum  
);
```

Parametri

UserSession
[in] Identificatore della sessione di comunicazione con il controllo numerico.

PathName
[in] Path-name completo della directory da creare.

pErrClass
[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum
[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.
Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

LogFSCreateFile_C

Crea un nuovo file.

```
WORD LogFSCreateFile_C (  
    WORD        UserSession,  
    LPCWSTR     PathName,  
    DWORD *     pErrClass,  
    DWORD *     pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

PathName

[in] Path-name completo del file da creare.

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

LogFSGetFileSize_C

Legge la dimensione di un file.

```
WORD LogFSGetFileSize_C (  
    WORD        UserSession,  
    LPCWSTR     PathName,  
    UINT *      pSize,  
    DWORD *     pErrClass,  
    DWORD *     pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

PathName

[in] Path-name completo del file.

pSize

[out] Puntatore alla variabile dove verrà scritta la dimensione del file in byte.

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

LogFSGetFileAttrib_C

Legge gli attributi di un file.

```
WORD LogFSGetFileAttrib_C (  
    WORD UserSession,  
    LPCWSTR PathName,  
    DWORD * pAttrib,  
    DWORD * pErrClass,  
    DWORD * pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

PathName

[in] Path-name completo del file.

pAttrib

[out] Puntatore alla variabile dove verranno scritti gli attributi del file.

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Per il significato dei singoli bit di attributo si riporta qui la documentazione MSDN Microsoft della funzione GetFileAttributes.

The attributes can be one or more of the following values.

Attribute	Meaning
FILE_ATTRIBUTE_ARCHIVE (0x00000020)	A file or directory that is an archive file or directory. Applications use this attribute to mark files for backup or removal.
FILE_ATTRIBUTE_COMPRESSED (0x00000800)	A file or directory that is compressed. For a file, all of the data in the file is compressed. For a directory, compression is the default for



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

FILE_ATTRIBUTE_DEVICE (0x00000040)	newly created files and subdirectories.
FILE_ATTRIBUTE_DIRECTORY (0x00000010)	Reserved; do not use. The handle that identifies a directory. A file or directory that is encrypted.
FILE_ATTRIBUTE_ENCRYPTED (0x00004000)	For a file, all data streams in the file are encrypted. For a directory, encryption is the default for newly created files and subdirectories.
FILE_ATTRIBUTE_HIDDEN (0x00000002)	The file or directory is hidden. It is not included in an ordinary directory listing.
FILE_ATTRIBUTE_NORMAL (0x00000080)	A file or directory that does not have other attributes set.
FILE_ATTRIBUTE_NOT_CONTENT_INDEXED (0x00002000)	This attribute is valid only when used alone. The file is not to be indexed by the content indexing service. The data of a file is not available immediately.
FILE_ATTRIBUTE_OFFLINE (0x00001000)	This attribute indicates that the file data is physically moved to offline storage. This attribute is used by Remote Storage, which is the hierarchical storage management software. Applications should not arbitrarily change this attribute. A file or directory that is read-only.
FILE_ATTRIBUTE_READONLY (0x00000001)	For a file, applications can read the file, but cannot write to it or delete it.
FILE_ATTRIBUTE_REPARSE_POINT (0x00000400)	For a directory, applications cannot delete it.
FILE_ATTRIBUTE_SPARSE_FILE (0x00000200)	A file or directory that has an associated reparse point. A file that is a sparse file.
FILE_ATTRIBUTE_SYSTEM (0x00000004)	A file or directory that the operating system uses a part of, or uses exclusively. A file that is being used for temporary storage.
FILE_ATTRIBUTE_TEMPORARY (0x00000100)	File systems avoid writing data back to mass storage if sufficient cache memory is available, because typically, an application deletes a temporary file after the handle is closed. In that scenario, the system can entirely avoid writing the data. Otherwise, the data is written after the handle is closed.
FILE_ATTRIBUTE_VIRTUAL	A file is a virtual file.



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

Vedi anche

`LogFSSetFileAttrib_C`, `LogFSChangeFileAttrib`



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

LogFSSetFileAttrib_C

Imposta gli attributi di un file.

```
WORD LogFSSetFileAttrib_C (  
    WORD        UserSession,  
    LPCWSTR     PathName,  
    DWORD       Attrib,  
    DWORD *     pErrClass,  
    DWORD *     pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

PathName

[in] Path-name completo del file.

Attrib

[in] Attributi da impostare per il file specificato da path-name. Per i possibili valori vedi LogFSGetFileAttrib_C.

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Vedi anche

LogFSGetFileAttrib_C, LogFSChangeFileAttrib



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

LogFSChangeFileAttrib_C

Modifica gli attributi di un file.

```
WORD LogFSChangeFileAttrib_C (  
    WORD          UserSession,  
    LPCWSTR PathName,  
    DWORD          Add,  
    DWORD          Remove,  
    DWORD * pErrClass,  
    DWORD * pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

PathName

[in] Path-name completo del file.

Add

[in] Attributi da aggiungere al file specificato da path-name. Per i possibili valori vedi LogFSGetFileAttrib_C.

Remove

[in] Attributi da rimuovere dal file specificato da path-name. Per i possibili valori vedi LogFSGetFileAttrib_C.

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Vedi anche

LogFSGetFileAttrib_C, LogFSSetFileAttrib_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

LogFSFindFirst_C

La funzione LogFSFindFirst_C cerca, in una directory, una sotto-directory o un file con un nome che corrispondente al nome specificato.

Se il nome file contiene dei caratteri jolly (p.e. "*") la funzione permette di ricercare liste di file o sotto-directory.

Se la funzione è eseguita senza errori, oltre ai dati relativi alla directory/file trovata, restituisce un identificatore (HANDLE) di ricerca che può essere usato per continuare la ricerca tramite la funzione LogFSFindNext_C.

Al termine del ciclo di ricerca l'handle deve essere chiuso tramite la funzione LogFSFindClose;

```
WORD LogFSFindFirst_C (  
    WORD UserSession,  
    LPCWSTR pFileName,  
    FILE_FIND_DATA_C4 * pFindData,  
    HANDLE * pHFinder,  
    DWORD * pErrClass,  
    DWORD * pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

pFileName

[in] Path-name completo del primo file o directory da cercare.

pFindData

[out] Puntatore ad una struttura di tipo FILE_FIND_DATA_C4 dove verranno scritte le informazioni relative all'elemento ricercato.

pHFinder

[out] Puntatore all'HANDLE della ricerca. Se la funzione ha terminato l'esecuzione senza errori, l'handle può essere usato nella funzione LogFSFindNext_C per cercare l'elemento successivo. Se non è stato trovato alcun file, il valore dell'handle è 0xFFFFFFFF (INVALID_HANDLE_VALUE).

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione è stata eseguita con successo, zero in caso di errore. Se la ricerca termina senza aver trovato alcun file non viene segnalato errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

Vedi anche

LogFSFindNext_C, LogFSFindClose_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

LogFSFindNext_C

Questa funzione continua la ricerca file iniziata da una precedente chiamata alla funzione LogFSFindFirst_C (vedi descrizione funzione LogFSFindFirst_C).

```
WORD LogFSFindNext_C (  
    WORD                UserSession,  
    FILE_FIND_DATA_C4 * pFindData,  
    HANDLE              hFinder,  
    BOOL *              pFound,  
    DWORD *             pErrClass,  
    DWORD *             pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

pFindData

[out] Puntatore ad una struttura di tipo FILE_FIND_DATA_C4 dove verranno scritte le informazioni relative all'elemento trovato.

hFinder

[out] HANDLE della ricerca che è stato restituito da una precedente chiamata alla funzione LogFSFindFirst_C.

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione è stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Vedi anche

LogFSFindFirst_C, LogFSFindClose_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

LogFSFindClose_C

Chiude una sessione di ricerca file aperta con la funzione LogFSFindFirst_C (vedi descrizione funzione LogFSFindFirst_C).

```
WORD LogFSFindClose_C (  
    WORD        UserSession,  
    HANDLE       hFinder,  
    DWORD *      pErrClass,  
    DWORD *      pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

hFinder

[out] HANDLE della ricerca che è stato restituito da una precedente chiamata alla funzione LogFSFindFirst_C. Una volta chiamata questa funzione l'HANDLE non è più valido e non deve essere riutilizzato.

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione è stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Vedi anche

LogFSFindFirst_C, LogFSFindNext_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

LogFSRemoveFile_C

Cancella un file.

```
WORD LogFSRemoveFile_C (  
    WORD        UserSession,  
    LPCWSTR     PathName,  
    LPCWSTR     FileName,  
    DWORD *     pErrClass,  
    DWORD *     pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

PathName

[in] Path-name completo del file da cancellare.

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

LogFSRemoveDir_C

Cancella una directory e tutto il suo contenuto.

```
WORD LogFSRemoveDir_C (  
    WORD    UserSession,  
    LPCWSTR PathName,  
    DWORD * pErrClass,  
    DWORD * pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

PathName

[in] Path-name completo della directory da rimuovere.

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

LogFSRename_C

Modifica il nome di un file o di una directory.

```
WORD LogFSRename_C (  
    WORD    UserSession,  
    LPCWSTR PathName,  
    LPCWSTR NewPathName,  
    DWORD * pErrClass,  
    DWORD * pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

PathName

[in] Path-name completo del file od directory da rinominare.

NewPathName

[in] Nuovo nome da assegnare al file indicato da PathName.

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

LogFSCopyFile_C

Copia un file in una directory appartenente al medesimo file system. Questa funzione è più veloce delle funzioni LogFSTransferFile_C e LogFSTransferFileW_C perchè sfrutta la connessione con il CNC già disponibile ma non può copiare un file in una directory appartenente ad un altro file system, come accade ad esempio nella copia di un file tra due CNC o tra un CNC ed un PC.

```
WORD LogFSCopyFile_C (  
    WORD    UserSession,  
    LPCWSTR PathName,  
    LPCWSTR NewPathName,  
    BOOL     FailIfExists,  
    DWORD *  pErrClass,  
    DWORD *  pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

PathName

[in] Path-name completo del file da copiare.

NewPathName

[in] Path-name completo del file di destinazione.

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Vedi anche

LogFSTransferFile_C

LogFSTransferFileW_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

LogFSGetInfo_C

Restituisce la dimensione e lo spazio libero del disco sul quale è definito un drive logico.

```
WORD LogFSGetInfo_C (  
    WORD    UserSession,  
    LPCWSTR PathName,  
    DWORD   Selector,  
    WORD    ItemCounter,  
    DWORD   pBuffer  
    DWORD * pErrClass,  
    DWORD * pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

PathName

[in] Path-name che contiene il nome drive logico. Il path-name può indicare il solo drive o anche una directory e un file. La funzione estrae il nome drive logico dall'inizio del path-name.

Selector

[in] Selettore comando. Deve valere 1.

ItemCounter

[in] Numero di elementi presenti nell'array pBuffer. Vedi anche parametro pBuffer

pBuffer

[in] Array dove verranno scritti i dati. Devono essere presenti un numero di elementi almeno pari ad ItemCounter. Gli elementi possono essere 2 o 4. Specificando 2 verrà restituito lo spazio libero su disco. Specificando 4 verrà restituita anche la dimensione totale del disco.

Gli elementi dell'array conterranno i seguenti dati.

Buffer[0] = parte bassa spazio libero su disco

Buffer[1] = parte alta spazio libero su disco

Buffer[2] = parte bassa spazio totale disco

Buffer[3] = parte alta spazio totale disco

Per calcolare lo spazio libero (o la dimensione disco) si può utilizzare una variabile di appoggio di tipo `double` ed applicare la formula:

```
double Space = (double)LowValue +  
                ((double)HighValue * (double)0x100000000);
```

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione è stata eseguita con successo, zero in caso di errore.



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

Funzioni per la gestione dei nomi file lunghi

Queste funzioni possono essere chiamate solo dopo aver creato il server Cndex (vedi ConnectServer_C), aperto una sessione di comunicazione (vedi OpenSession_C) ed aver verificato che il CNC si trovi nella modalità RUN (vedi BootPhaseEnquiry_C).

In un sistema Serie10 i file con nome "lungo" (più di 8 caratteri) possono essere creati solamente utilizzando l'interfaccia PartProgram del CNC. Il sistema mette a disposizione quattro directory logiche, denominate rispettivamente PROGRAM, USER, SYSTEM e OEM, nelle quali inserire i file creati tramite l'interfaccia. Il sistema utilizza un metodo proprietario per gestire i nomi di file lunghi, per cui non è possibile trasferire file con nome lungo da un PC attraverso la connessione di un disco remoto.

Mediante le funzioni descritte in questo paragrafo è possibile creare o eliminare file con nome lungo da un PC remoto, esattamente come se si usasse l'interfaccia del CNC.

Ogni file con nome lungo viene in realtà salvato come file DOS, con un nome generato automaticamente, nella directory UPP del disco F. Il sistema associa il nome DOS del file al nome lungo che viene visualizzato dall'interfaccia tramite una tabella cross-reference.

La funzione PPInsertName_C crea una nuova entry in questa tabella. La funzione PPDeleteName_C elimina dalla tabella una entry creata in precedenza.

La funzione PPGetLogicalName_C restituisce il nome file logico, o "lungo", una volta noto il nome fisico del file DOS. La funzione PPGetPhysicalName_C restituisce il nome file fisico quando è noto il nome logico.

La funzione PPGetLogicalDir_C restituisce il numero della directory logica che contiene il file. Per la codifica delle directory si consulti il paragrafo "Descrizione delle strutture e definizioni".

La funzione PPUpdate_C aggiorna le informazioni sul file visualizzate dall'interfaccia del CNC. In generale le uniche informazioni che necessitano un'aggiornamento sono SIZE, DATE e TIME.



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

PPInsertName_C

Crea una entry nella tabella cross-reference dei nomi di file lunghi. Il CNC genera automaticamente il nome fisico del file e crea il file con lunghezza 0 nella directory UPP del disco F.

```
WORD PPInsertName_C (  
    WORD          UserSession,  
    LPSTR         LogicalName,  
    LPSTR         LogicalExt,  
    WORD          LogicalDir,  
    LPSTR         PhysicalName,  
    DWORD *       pErrClass,  
    DWORD *       pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

LogicalName

[in] Stringa con carattere terminatore nullo contenente il nome file logico, senza estensione.
Lunghezza massima 49 caratteri, terminatore incluso.

LogicalExt

[in] Stringa con carattere terminatore nullo contenente l'estensione del nome file logico.
Lunghezza massima 4 caratteri, terminatore incluso.

LogicalDir

[in] Identificatore della directory logica in cui inserire il file.

PhysicalName

[out] Stringa con carattere terminatore nullo contenente il nome file fisico generato dal CNC. Il nome file fisico non ha estensione. Lunghezza massima 9 caratteri, terminatore incluso.

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.
Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Vedi anche

PPDeleteName_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

PPDeleteName_C

Elimina dalla tabella cross-reference dei nomi di file lunghi una entry creata in precedenza. È necessario conoscere il nome logico del file, l'estensione e la directory logica che lo contiene. Il CNC cancella automaticamente anche il file fisico.

```
WORD PPDeleteName_C (  
    WORD          UserSession,  
    LPSTR         LogicalName,  
    LPSTR         LogicalExt,  
    WORD          LogicalDir,  
    WORD *        NumberOfNamesDeleted,  
    DWORD * pErrClass,  
    DWORD * pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

LogicalName

[in] Stringa con carattere terminatore nullo contenente il nome file logico, senza estensione.
Lunghezza massima 49 caratteri, terminatore incluso.

LogicalExt

[in] Stringa con carattere terminatore nullo contenente l'estensione del nome file logico.
Lunghezza massima 4 caratteri, terminatore incluso.

LogicalDir

[in] Identificatore della directory logica che contiene il file.

NumberOfNamesDeleted

[out] Numero di file eliminati. Questa funzione non accetta caratteri jolly, per cui questo campo può contenere i valori 0 o 1.

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione è stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Vedi anche

PPInsertName_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

PPGetPLogicalName_C

Restituisce il nome logico e l'estensione del file. È necessario conoscere il nome del file fisico.

```
WORD PPGetLogicalName_C (  
    WORD        UserSession,  
    LPSTR       PhysicalName,  
    LPSTR       LogicalName,  
    LPSTR       LogicalExt,  
    WORD        LogicalDir,  
    DWORD *     pErrClass,  
    DWORD *     pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

PhysicalName

[in] Stringa con carattere terminatore nullo contenente il nome del file fisico associato al nome file logico. Il nome file fisico non ha estensione. Lunghezza massima 9 caratteri, terminatore incluso.

LogicalName

[out] Stringa con carattere terminatore nullo contenente il nome file logico, senza estensione. Lunghezza massima 49 caratteri, terminatore incluso.

LogicalExt

[out] Stringa con carattere terminatore nullo contenente l'estensione del nome file logico. Lunghezza massima 4 caratteri, terminatore incluso.

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione è stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Vedi anche

PPGetPhysicalName_C, PPGetLogicalDir_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

PPGetPhysicalName_C

Restituisce il nome del file fisico. È necessario conoscere il nome logico, l'estensione e la directory logica che contiene il file.

```
WORD PPGetPhysicalName_C (  
    WORD        UserSession,  
    LPSTR       LogicalName,  
    LPSTR       LogicalExt,  
    WORD        LogicalDir,  
    LPSTR       PhysicalName,  
    DWORD *     pErrClass,  
    DWORD *     pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

LogicalName

[in] Stringa con carattere terminatore nullo contenente il nome file logico, senza estensione.
Lunghezza massima 49 caratteri, terminatore incluso.

LogicalExt

[in] Stringa con carattere terminatore nullo contenente l'estensione del nome file logico.
Lunghezza massima 4 caratteri, terminatore incluso.

LogicalDir

[in] Identificatore della directory logica che contiene il file.

PhysicalName

[out] Stringa con carattere terminatore nullo contenente il nome del file fisico associato al nome file logico. Il nome file fisico non ha estensione. Lunghezza massima 9 caratteri, terminatore incluso.

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione è stata eseguita con successo, zero in caso di errore.
Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Vedi anche

PPGetLogicalName_C, PPGetLogicalDir_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

PPGetLogicalDir_C

Restituisce la directory logica che contiene il file. È necessario conoscere il nome logico del file e l'estensione.

```
WORD PPGetLogicalDir_C (  
    WORD        UserSession,  
    LPSTR       LogicalName,  
    LPSTR       LogicalExt,  
    WORD *      LogicalDir,  
    DWORD *     pErrClass,  
    DWORD *     pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

LogicalName

[in] Stringa con carattere terminatore nullo contenente il nome file logico, senza estensione.
Lunghezza massima 49 caratteri, terminatore incluso.

LogicalExt

[in] Stringa con carattere terminatore nullo contenente l'estensione del nome file logico.
Lunghezza massima 4 caratteri, terminatore incluso.

LogicalDir

[out] Identificatore della directory logica che contiene il file.

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione è stata eseguita con successo, zero in caso di errore.
Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Vedi anche

PPGetLogicalName_C, PPGetPhysicalName_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

PPUpdate_C

Aggiorna le informazioni sul file visualizzate dall'interfaccia Part Program del CNC. Di solito l'aggiornamento è richiesto dopo che il file è stato modificato al di fuori dell'interfaccia Part Program.

```
WORD PPUpdate_C (  
    WORD          UserSession,  
    LPSTR         LogicalName,  
    LPSTR         LogicalExt,  
    WORD          LogicalDir,  
    DWORD * pErrClass,  
    DWORD * pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

LogicalName

[in] Stringa con carattere terminatore nullo contenente il nome file logico, senza estensione.
Lunghezza massima 49 caratteri, terminatore incluso.

LogicalExt

[in] Stringa con carattere terminatore nullo contenente l'estensione del nome file logico.
Lunghezza massima 4 caratteri, terminatore incluso.

LogicalDir

[in] Identificatore della directory logica che contiene il file.

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore.

Valore di ritorno

Diverso da zero se la funzione è stata eseguita con successo, zero in caso di errore.
Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Vedi anche

PPInsertName_C, PPDeleteName_C



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

Funzioni per la gestione CANOPEN HILSCHER

Questo capitolo contiene le funzioni dedicate alla gestione dei dispositivi posti su BUS CANOpen e collegati alla scheda master Hilscher. Per informazioni più approfondite si consulti il manuale Libreria WinPLUS capitolo 10.



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

CANInit_C

La funzione permette di impostare il tempo di rinfresco dei dati a disposizione della logica. Tali dati sono quelli di trasmissione e ricezione tramite PDO (Process Data Object).

Inoltre la funzione permette di informare la scheda circa la mappatura in memoria dei dati di ricezione e trasmissione nonché dei dati di diagnostica della scheda master e dei nodi.

La funzione deve essere chiamata prima di ogni altra operazione da effettuare sui nodi presenti sul BUS CANopen. **Il sistema inizierà a rinfrescare i dati di Input/Output soltanto dopo la chiamata della funzione CAN_INIT.**

```
WORD CANInit_C (  
    WORD          UserSession,  
    WORD          DevNum,  
    WORD          TickNr,  
    WORD          Inp1,  
    WORD          Inp2,  
    WORD          Out1,  
    WORD          Out2,  
    WORD          OfftDiagn,  
    DWORD * pErrClass,  
    DWORD * pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

DevNum

[in] Indica il numero della scheda master CANOpen su cui attivare i parametri.

TickNr

[in] Indica il tempo di rinfresco dei dati. Il tempo deve essere impostato come multiplo di 10ms. Se viene impostato un valore non multiplo di 10ms il sistema arrotonda il valore al multiplo superiore più vicino. (Esempio: Valore impostato da logica uguale a 13ms → valore impostato dal sistema uguale a 20ms).

Inp1

Inp2

[in] Indicano rispettivamente il primo e l'ultimo byte relativi ai dati di input (Intervallo 0-1023). Se si desidera una rimappatura degli offset degli Input entrambi i parametri devono essere posti uguali a **zero**, mentre le informazioni di rimappatura saranno presenti nella sezione [INPUT] del file *.RMP.

Out1

Out2

[in] Indicano rispettivamente il primo e l'ultimo byte relativi ai dati di output (Intervallo 0-1023). Se si desidera una rimappatura degli offset degli Output entrambi i parametri devono essere posti uguali a zero, mentre le informazioni di rimappatura saranno presenti nella sezione [OUTPUT] del file *.RMP.

OffsDiagn



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

[in] Indica il primo byte del buffer di diagnostica. Tale buffer viene allocato dal sistema all'interno del buffer di input. A partire dal byte indicato e per una lunghezza di 48 byte viene memorizzato lo stato della scheda master e lo stato dei nodi. Per la descrizione dei dati memorizzati nel buffer di diagnostica fare riferimento alla descrizione presente nel manuale di applicazione WinPLUS.

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore. Gli errori di classe 4 sono codificati come segue:

Valore	Descrizione
0	Funzione eseguita correttamente
307	Inizializzazione già effettuata
308	L'intervallo di input o l'intervallo della diagnostica si sovrappone con l'intervallo di input di altra scheda master CANopen presente nel sistema
309	L'intervallo di input o l'intervallo della diagnostica si sovrappone con l'intervallo di diagnostica di altra scheda master CANopen presente nel sistema
310	L'intervallo di output si sovrappone con l'intervallo di output di altra scheda master CANopen presente nel sistema
314	File di remapping non trovato
400	Numero della scheda master non valido
405	Valore Offset negativo
406	Offset degli Input fuori range
407	Primo byte di input maggiore dell'ultimo byte di input
408	Offset degli Output fuori range
409	Primo byte di output maggiore dell'ultimo byte di output
410	Offset del buffer di diagnostica fuori range
411	Sovrapposizione buffer di diagnostica su intervallo di input

Valore di ritorno

Diverso da zero se la funzione è stata eseguita con successo, zero in caso di errore.
Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Vedi anche

CANBoard_C, CANNMT_C, CANSync_C, CANReadSDO_C, CANWriteSDO_C, CANGetEmergency_C, CANConsoleCfg_C.



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

CANBoard_C

La funzione invia un comando alla scheda master identificata dal parametro **DevNum**.

```
WORD CANBoard_C (  
    WORD          UserSession,  
    WORD          DevNum,  
    WORD          Cmd,  
    DWORD * pErrClass,  
    DWORD * pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

DevNum

[in] Indica il numero della scheda master CANOpen.

Cmd

[in] Indica il tipo di comando da inviare. I valori possibili sono:

Valori	Descrizione
1	RESET della scheda master
2	START della comunicazione CANopen
3	STOP della comunicazione CANopen

Il comando di RESET inizializza la DualPort della scheda master con i parametri salvati nella memoria FLASH del dispositivo. L'operazione di RESET può durare fino a 10 secondi e fa cadere ogni comunicazione CANopen, quindi i nodi remoti potranno andare nello stato di WatchDog. Per riattivare la comunicazione alla fine del comando di RESET bisogna sempre dare il comando di START.

Il comando di START ripristina la comunicazione CANopen ed effettua tutta la sequenza di "Node Bootup" impostata tramite il configuratore SyCon.

Il comando di STOP ferma ogni attività CANopen sul BUS.

Nello stato di STOP e durante l'operazione di RESET i nodi remoti potranno andare nello stato di WatchDog.

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore. Gli errori di classe 4 sono codificati come segue:

Valore	Descrizione
0	Funzione eseguita correttamente
104	Tipo di comando non valido
105	Reset della scheda in corso
400	Numero della scheda master non valido
Altro	Vedi appendice B del manuale di applicazione WinPLUS

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.
Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Vedi anche

CANInit_C, CANNMT_C, CANSync_C, CANReadSDO_C, CANWriteSDO_C, CANGetEmergency_C, CANConsoleCfg_C.



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

CANSync_C

La funzione invia comandi di SYNC su tutta la rete. Tale comando è utile per aggiornare dati da/per moduli analogici. **La funzione può essere eseguita soltanto in modalità WAIT.**

```
WORD CANSync_C (  
    WORD          UserSession,  
    WORD          DevNum,  
    WORD          Timeout,  
    BYTE          Sync,  
    DWORD * pErrClass,  
    DWORD * pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

DevNum

[in] Indica il numero della scheda master CANopen.

Timeout

[in] Indica il timeout in ms (0 = nessun timeout).

Sync

[in] Modalità di esecuzione funzione (0 = NOWAIT, 1 = WAIT).

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore. Gli errori di classe 4 sono codificati come segue:

Valore	Descrizione
0	Funzione eseguita correttamente
105	Reset della scheda master in corso
106	Chip CAN della scheda master nello stato di OFFLINE
107	Chip CAN della scheda master nello stato di STOP
108	Chip CAN della scheda master nello stato di CLEAR
306	Canale ancora da inizializzare
400	Numero della scheda master non valido
404	Modalità NO WAIT non possibile

Valore di ritorno

Diverso da zero se la funzione è stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

Vedi anche

CANInit_C, CANBoard_C, CANNMT_C, CANReadSDO_C, CANWriteSDO_C, CANGetEmergency_C, CANConsoleCfg_C.



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

CANNMT_C

La funzione serve per l'esecuzione di servizi di gestione della rete (NMT: Network management). Attraverso questi servizi i nodi possono essere inizializzati, avviati, fermati e resettati.

La funzione è unicamente identificata dall'identificativo del nodo (idNode 1-127). Per l'esecuzione del servizio su tutti i nodi presenti su BUS si avrà idNode = 0.

```
WORD CANNMT_C (  
    WORD        UserSession,  
    WORD        DevNum,  
    BYTE        idNode,  
    WORD        Cmd,  
    WORD        Timeout,  
    BYTE        Sync,  
    DWORD * pErrClass,  
    DWORD * pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

DevNum

[in] Indica il numero della scheda master CANopen.

idNode

[in] Identificativo del nodo (0 = Broadcasting).

Cmd

[in] Indica il tipo di servizio da inviare. I valori possibili sono:

Valori	Descrizione
1	Avvio nodo remoto
2	Stop nodo remoto
128	Imposta stato pre-operazionale
129	Reset del nodo remoto
130	Reset della comunicazione

Timeout

[in] Indica il timeout in ms (0 = nessun timeout).

Sync

[in] Modalità di esecuzione funzione (0 = NOWAIT, 1 = WAIT).

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore. Gli errori di classe 4 sono codificati come segue:



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

Valore	Descrizione
0	Funzione eseguita correttamente
100	Comando rifiutato (Lista comandi interna piena) (solo modalità NO WAIT)
101	Comando accettato in attesa di invio (solo modalità NO WAIT)
102	Comando inviato al nodo ed in attesa di risposta (solo modalità NO WAIT)
103	Errore di Timeout
105	Reset della scheda master in corso
106	Chip CAN della scheda master nello stato di OFFLINE
107	Chip CAN della scheda master nello stato di STOP
108	Chip CAN della scheda master nello stato di CLEAR
306	Canale ancora da inizializzare
400	Numero della scheda master non valido
401	Identificativo nodo non valido
403	Comando NMT errato

Valore di ritorno

Diverso da zero se la funzione e' stata eseguita con successo, zero in caso di errore.
Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Vedi anche

CANInit_C, CANBoard_C, CANSync_C, CANReadSDO_C, CANWriteSDO_C, CANGetEmergency_C, CANConsoleCfg_C.



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

CANReadSDO_C

La funzione permette di leggere le informazioni contenute nell'oggetto di un nodo. L'oggetto è specificato tramite i parametri **Index** e **subIndex**.

```
WORD CANReadSDO_C (  
    WORD      UserSession,  
    WORD      DevNum,  
    WORD      idNode,  
    WORD      Index,  
    BYTE      SubIndex,  
    WORD      Timeout,  
    BYTE      Sync,  
    BYTE *    Data,  
    WORD *    DataLen,  
    DWORD *   pErrClass,  
    DWORD *   pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

DevNum

[in] Indica il numero della scheda master CANopen.

idNode

[in] Identificativo del nodo (0 = Broadcasting).

Index

[in] Indice oggetto.

SubIndex

[in] sottoindice oggetto.

Timeout

[in] Indica il timeout in ms (0 = nessun timeout).

Sync

[in] Modalità di esecuzione funzione (0 = NOWAIT, 1 = WAIT).

DataLen

[out] puntatore ad una word contenente la lunghezza in byte del blocco dati restituito dalla funzione.

Data

[out] Puntatore al buffer nel quale verrà copiato il blocco dati restituito dalla funzione.

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore. Gli errori di classe 4 sono codificati come segue:

Valore	Descrizione
0	Funzione eseguita correttamente
100	Comando rifiutato (Lista comandi interna piena) (solo modalità WAIT)
101	Comando accettato dal driver (solo modalità WAIT)
102	Comando inviato al nodo (solo modalità WAIT)
103	Errore di Timeout
105	Reset della scheda master in corso
106	Chip CAN della scheda master nello stato di OFFLINE
107	Chip CAN della scheda master nello stato di STOP
108	Chip CAN della scheda master nello stato di CLEAR
306	Canale ancora da inizializzare
400	Numero della scheda master non valido
401	Identificativo nodo non valido
1003	Il servizio è stato rifiutato dal nodo con il servizio di Abort SDO. La causa potrebbe essere dovuta al parametro Index e subIndex non validi oppure non si hanno i diritti di accesso al nodo. I dati in uscita alla funzione contengono il codice relativo a Abort SDO.
1017	Nessuna risposta dal nodo o nodo non presente
1019	Il nodo non si trova nello stato operativo e l'accesso viene negato. Il canale SDO è usato al momento dalla scheda master per la configurazione del nodo nella fase di avvio. Riprovare a richiedere il servizio.
1051	Il buffer di ricezione ha superato i limiti.
1053	I dati di protocollo frammentati superano i limiti del buffer di ricezione.
1054	Il servizio precedente è ancora attivo e non confermato all'applicazione.
1057	Errore di sequenza nella frammentazione del protocollo. La richiesta è abortita.
1200	Scheda master non configurata.

Valore di ritorno

Diverso da zero se la funzione è stata eseguita con successo, zero in caso di errore.
Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Vedi anche

CANInit_C, CANBoard_C, CANSync_C, CANNMT_C, CANWriteSDO_C, CANGetEmergency_C, CANConsoleCfg_C.



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

CANWriteSDO_C

La funzione permette di scrivere le informazioni contenute nel parametro *Data* nell'oggetto di un nodo.

L'oggetto è specificato tramite i parametri ***Index*** e ***SubIndex***.

Nel caso in cui il comando di scrittura venga rifiutato dal nodo con un codice di **Abort SDO**, il parametro *Data* contiene il codice di errore relativo ad Abort SDO.

```
WORD CANWriteSDO_C (  
    WORD      UserSession,  
    WORD      DevNum,  
    WORD      idNode,  
    WORD      Index,  
    BYTE      SubIndex,  
    WORD      Timeout,  
    BYTE      Sync,  
    BYTE *    Data,  
    WORD *    DataLen,  
    DWORD *   pErrClass,  
    DWORD *   pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

DevNum

[in] Indica il numero della scheda master CANopen.

idNode

[in] Identificativo del nodo (0 = Broadcasting).

Index

[in] Indice oggetto.

SubIndex

[in] sottoindice oggetto.

Timeout

[in] Indica il timeout in ms (0 = nessun timeout).

Sync

[in] Modalità di esecuzione funzione (0 = NOWAIT, 1 = WAIT).

DataLen

[in,out] puntatore ad una word contenente la lunghezza in byte del blocco dati da scrivere, oppure del codice di Abort SDO.

Data

[in,out] Puntatore al buffer contenente il blocco dati da scaricare nel nodo, oppure il codice di Abort SDO.



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore. Gli errori di classe 4 sono codificati come segue:

Valore	Descrizione
0	Funzione eseguita correttamente
100	Comando rifiutato (Lista comandi interna piena) (solo modalità WAIT)
101	Comando accettato dal driver (solo modalità WAIT)
102	Comando inviato al nodo (solo modalità WAIT)
103	Errore di Timeout
105	Reset della scheda master in corso
106	Chip CAN della scheda master nello stato di OFFLINE
107	Chip CAN della scheda master nello stato di STOP
108	Chip CAN della scheda master nello stato di CLEAR
306	Canale ancora da inizializzare
400	Numero della scheda master non valido
401	Identificativo nodo non valido
1003	Il servizio è stato rifiutato dal nodo con il servizio di Abort SDO. La causa potrebbe essere dovuta al parametro Index e SubIndex non validi oppure non si hanno i diritti di accesso al nodo. I dati in uscita alla funzione contengono il codice relativo a Abort SDO.
1017	Nessuna risposta dal nodo o nodo non presente
1019	Il nodo non si trova nello stato operativo e l'accesso viene negato. Il canale SDO è usato al momento dalla scheda master per la configurazione del nodo nella fase di avvio. Riprovare a richiedere il servizio.
1051	Il buffer di ricezione ha superato i limiti.
1053	I dati di protocollo frammentati superano i limiti del buffer di ricezione.
1054	Il servizio precedente è ancora attivo e non confermato all'applicazione.
1057	Errore di sequenza nella frammentazione del protocollo. La richiesta è abortita.
1200	Scheda master non configurata.

Valore di ritorno

Diverso da zero se la funzione è stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

Vedi anche

CANInit_C, CANBoard_C, CANSync_C, CANNMT_C, CANReadSDO_C, CANGetEmergency_C,
CANConsoleCfg_C.



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

CANGetEmergency_C

La funzione legge dal buffer interno della scheda master le emergenze relative ad un nodo.
La funzione può essere chiamata dopo aver verificato se esistono nodi in emergenza. La verifica va fatta sul buffer di diagnostica a partire dal byte OFFSETDIAGN+32 per una lunghezza di 16 byte.

```
WORD CANGetEmergency_C (  
    WORD        UserSession,  
    WORD        DevNum,  
    WORD        idNode,  
    WORD        Timeout,  
    BYTE        Sync,  
    WORD *      NodeStatus,  
    WORD *      AddInfo,  
    WORD *      ProfNum,  
    short *     NodeStateNG,  
    short *     ActualErr,  
    short *     EmcyLen,  
    EMCY_TYPE * EmcyData  
    DWORD * pErrClass,  
    DWORD * pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

DevNum

[in] Indica il numero della scheda master CANopen.

idNode

[in] Identificativo del nodo. Il valore 0 (Broadcasting) non è ammesso.

Timeout

[in] Indica il timeout in ms (0 = nessun timeout).

Sync

[in] Modalità di esecuzione funzione (0 = NOWAIT, 1 = WAIT).

NodeStatus

[out] Maschera dei flag di stato del nodo. I flag sono mappati come segue:

Bit	Descrizione
0	Il nodo non risponde.
1	Il numero di emergenze ha superato le dimensioni del buffer.
2	C'è una differenza tra i dati di configurazione della scheda master e del nodo.
3	WatchDog attivo sul nodo.
4-7	Riservati



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

AddInfo

[out] Informazioni contenute nell'Oggetto 1000H.

ProfNum

[out] Numero di profilo dell'Oggetto 1000H.

NodeStateNG

[out] Stato del nodo. Se il WatchDog è attivo per il nodo in esame il suo stato viene riportato in questa variabile. I valori sono mostrati nella tabella seguente:

Valore	Descrizione
1	Nodo disconnesso
2	Nodo in connessione
3	Nodo in preparazione
4	Nodo preparato
5	Nodo in stato operativo
127	Nodo in stato pre-operativo

ActualErr

[out] Codice di errore del nodo. I possibili valori sono mostrati nella tabella seguente, mentre fra le parentesi è indicata una possibile soluzione:

Valore	Descrizione
30	Errore di WatchDog (Controllare che il nodo sia connesso)
31	Il nodo ha cambiato il suo stato e non è più operativo (Reset del nodo)
32	Sequenza errata nella gestione del WatchDog (reset nodo)
33	Nessuna risposta in seguito ad un PDO. (Verificare che il nodo sia abilitato ai PDO).
34	Nessuna risposta del nodo durante la sua configurazione (Verificare che il nodo sia connesso ed operativo).
35	Il numero di profilo indicato in configurazione è diverso dal numero di profilo del nodo (Verificare il numero di profilo).
36	Il tipo di device indicato in configurazione è diverso dal tipo di device del nodo (Verificare i servizi supportati dal nodo).
37	E' stata ricevuta una risposta di un SDO sconosciuto (Nodo non compatibile con le specifiche CiA).
38	Il messaggio ricevuto da un SDO ha una lunghezza diversa da 8 (Nodo non compatibile con le specifiche CiA).
39	Il nodo non è gestito dalla scheda master, nodo in STOP.

EmcyLen

[out] Lunghezza array delle emergenze.

EmcyData

[out] Array delle emergenze [0..4].



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore. Gli errori di classe 4 sono codificati come segue:

Valore	Descrizione
0	Funzione eseguita correttamente
306	Canale ancora da inizializzare
400	Numero della scheda master non valido
401	Identificativo nodo non valido
100	Comando rifiutato (Lista comandi interna piena) (solo modalità WAIT)
101	Comando accettato dal driver (solo modalità WAIT)
102	Comando inviato al nodo (solo modalità WAIT)
103	Errore di Timeout
105	Reset della scheda master in corso
106	Chip CAN della scheda master nello stato di OFFLINE
107	Chip CAN della scheda master nello stato di STOP
108	Chip CAN della scheda master nello stato di CLEAR
1161	Identificativo nodo non valido

Valore di ritorno

Diverso da zero se la funzione è stata eseguita con successo, zero in caso di errore.

Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Vedi anche

CANInit_C, CANBoard_C, CANNMT_C, CANSync_C, CANReadSDO_C, CANWriteSDO_C, CANConsoleCfg_C.



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

CANConsoleCfg_C

La funzione permette di configurare una console WinMedia collegate ad una scheda master CANopen intelligente.

```
WORD CANConsoleCfg_C (  
    WORD        UserSession,  
    WORD        DevNum,  
    BYTE        idNode,  
    WORD        InpOffs,  
    WORD        OutOffs,  
    WORD        Timeout,  
    BYTE        Sync,  
    DWORD * pErrClass,  
    DWORD * pErrNum  
);
```

Parametri

UserSession

[in] Identificatore della sessione di comunicazione con il controllo numerico.

DevNum

[in] Indica il numero della scheda master CANopen.

idNode

[in] Identificativo della console.

InpOffs

[in] Byte offset di Input.

OutOffs

[in] Byte offset di Output.

Timeout

[in] Indica il timeout in ms (0 = nessun timeout).

Sync

[in] Modalità di esecuzione funzione (0 = NOWAIT, 1 = WAIT).

pErrClass

[out] Puntatore alla variabile dove verrà scritta la classe dell'eventuale errore.

pErrNum

[out] Puntatore alla variabile dove verrà scritto il numero dell'eventuale errore. Gli errori di classe 4 sono codificati come segue:

Valore	Descrizione
0	Funzione eseguita correttamente
2	Console già configurata
100	Comando rifiutato (Lista comandi interna piena)



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

Valore	Descrizione
101	Comando accettato dal driver (solo modalità NO WAIT)
102	Comando inviato al nodo (solo modalità NO WAIT)
103	Errore di Timeout
105	Reset della scheda master in corso
106	Chip CAN della scheda master nello stato di OFFLINE
107	Chip CAN della scheda master nello stato di STOP
108	Chip CAN della scheda master nello stato di CLEAR
306	Canale ancora da inizializzare
400	Numero della scheda master non valido
401	Identificativo nodo non valido
1000+X o 2000+X	La configurazione dei led non è avvenuta correttamente. Riprovare ad eseguire la funzione.

Valore di ritorno

Diverso da zero se la funzione è stata eseguita con successo, zero in caso di errore.
Per i codici di errore (Classe e Numero) fare riferimento al capitolo "Gestione degli Errori".

Vedi anche

CANInit_C, CANBoard_C, CANNMT_C, CANSync_C, CANReadSDO_C, CANWriteSDO_C,
CANGetEmergency_C.



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

Strumenti utilizzabili da riga di comando

Introduzione

WinNbiCmd è un'applicazione di tipo "console" che implementa alcune funzionalità degli applicativi BootController, ODM e Security. L'eseguibile è distribuito insieme a WinNBI e si trova nella directory di installazione.

Ogni singola operazione è identificata da un comando passato come parametro della riga di comando.

Sintassi

La sintassi per l'utilizzo è la seguente:

```
<nome eseguibile> <comando> <opzioni>
```

Il comando identifica la singola operazione, mentre il numero e il tipo delle opzioni dipende dall'operazione richiesta.

Le opzioni vengono specificate tramite la sintassi:

```
<identificatore opzione> [[<valore>] [<valore>] ...]
```

Gli identificatori hanno il formato -<lettera>. Gli identificatori sono case sensitive.

Ad esempio, il nome CNC può essere specificato nel formato:

```
-c <nome CNC o indirizzo IP>
```

Ad esempio

```
-c 172.20.2.14
```

Ovviamente lo stesso nome opzione può avere significati differenti se utilizzato con interfacce e/o comandi differenti.



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

Comandi

L'elenco dei comandi implementati ad oggi è il seguente:

<code>backupcnc</code>	Esegue il backup di una o più aree dati del controllo numerico. Il CNC deve essere in modalità SETUP.
<code>restorecnc</code>	Esegue il ripristino di una o più aree dati del controllo numerico. Il CNC deve essere in modalità SETUP.
<code>installcnc</code>	Installa una nuova release software sul controllo numerico. Il CNC deve essere in modalità SETUP.
<code>rebootcnc</code>	Riavvia il controllo numerico. È necessario passare la modalità di boot come opzione della riga di comando. Il riavvio può avvenire in modo sincrono o asincrono. In modo sincrono WinNbiCmd.exe rimane in esecuzione fino a che il CNC ha raggiunto la modalità richiesta, oppure finché non scade il timeout se specificato. La modalità sincrona viene attivata mediante l'opzione -w. In modo asincrono l'esecuzione termina non appena inviato al CNC il comando di reboot. <i>Nota: nelle versioni di OPENcontrol antecedenti a 3.4.3, quando il CNC è in modalità SETUP il comando di reboot potrebbe generare errore.</i>
<code>activateamp</code>	Rende attiva una delle configurazioni presenti nel controllo numerico. Non viene verificato se lo slot specificato contiene effettivamente un database AMP, o se l'AMP è stato compilato.
<code>importamp</code>	Importa da un file .zpo un database AMP, inserendolo in uno slot AMP di un CNC oppure nella libreria offline.
<code>exportamp</code>	Esporta un database AMP da uno slot AMP di un CNC oppure dalla libreria offline in un file .zpo.
<code>buildamp</code>	Compila una configurazione AMP. Se lo slot specificato non contiene un database AMP il comando restituisce errore.

Elenco dei comandi e relative opzioni:

```
backupcnc      -o <output file path>
                -c <cncname>
                [-a <SYSTEM OEM USER DUALPORT ALL default=ALL>]
restorecnc     -i <input file path>
                -c <cncname>
                [-a <SYSTEM OEM USER DUAL PORT ALL default=ALL>]
installcnc     -i <input file path>
                -c <cncname>
rebootcnc      -m <RUNNING|EMERGENCY|SETUP|SERVICE>
                -c <cncname>
                [-w]
                [-t <timeout>]
activateamp     -s <0-9>
                -c <cncname>
importamp      -i <input file path>
                [-n <offline amp name> | -s <0-9> -c <cncname>]
                [-u]
exportamp       -o <output file path>
                [-n <offline amp name> | -s <0-9> -c <cncname>]
buildamp       -s <0-9>
                -c <cncname>
                [-u]
```



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

Opzioni

- c <cncname>
nome di rete o indirizzo IP del CNC.
- i <input file path>
percorso completo del file di input.
- o <output file path>
percorso completo del file di output.
- a <SYSTEM OEM USER DP ALL default=ALL>
Elenco delle aree dati del CNC, separate da spazi.
'ALL' indica tutte le aree.
- m <RUNNING|EMERGENCY|SETUP|SERVICE>
CNC boot mode.
- w
Attende finché il comando non ha completato l'esecuzione. Disponibile solamente per comandi che richiedono l'esecuzione di operazioni di una certa durata da parte del CNC, come per esempio il reboot del CNC.
- t <timeout>
Timeout di esecuzione del comando in millisecondi. Quando il timeout scade l'eseguibile termina la sua esecuzione anche se l'esecuzione del comando richiesto è ancora in corso. Se il valore di timeout è pari a zero oppure non è specificato WinNbiCmd attende che l'esecuzione del comando sia completa prima di terminare.
- s <0-9>
AMP slot number in the CNC. Must be used in conjunction with the -c option.
- n <offline amp name>
Name of the AMP configuration in the ODM offline library.
- u
update AMP configuration to the latest version if necessary.

Esempi pratici

Di seguito alcuni esempi di utilizzo di WinNbiCmd.

Backup di un CNC (aree system e user)

```
winnbicmd backupcnc -c 172.20.2.14 -a SYSTEM USER -o c:\backup\mybackup
```

Restore di un CNC (aree system e oem)

```
winnbicmd restorecnc -c 172.20.2.14 -a SYSTEM OEM -i c:\backup\172.20.2.14
```

Installazione/aggiornamento release CNC:

```
winnbicmd installcnc -c 172.20.2.14 -i c:\backup\172.20.2.14
```

Riavvio del CNC in modalità Emergenza:

```
winnbicmd rebootcnc -c 172.20.2.14 -m EMERGENCY
```

Riavvio del CNC in modalità Emergenza con attesa fine riavvio (reboot sincrono):

```
winnbicmd rebootcnc -c 172.20.2.14 -m EMERGENCY -w
```

Import di una configurazione AMP nella libreria offline con nome "MyAmpConfig":

```
winnbicmd importamp -n MyAmpConfig -i c:\amp\ampcfg.zpo
```

Import di una configurazione AMP nello slot AMP2 di un CNC:

```
winnbicmd importamp -c 172.20.2.14 -s 2 -i c:\amp\ampcfg.zpo
```



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

Export di una configurazione AMP con nome "MyAmpConfig" dalla libreria offline:

```
winnbicmd exportamp -n MyAmpConfig -o c:\amp\ampcfg.zpo
```

Export di una configurazione AMP dallo slot AMP0 di un CNC:

```
winnbicmd exportamp -c 172.20.2.14 -s 0 -o c:\amp\ampcfg.zpo
```

Compilazione della configurazione AMP contenuta nello slot AMP0 di un CNC:

```
winnbicmd buildamp -c 172.20.2.14 -s 0
```

Attivazione della configurazione AMP contenuta nello slot AMP0 di un CNC:

```
winnbicmd activateamp -c 172.20.2.14 -s 0
```



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

Descrizione delle strutture e definizioni

Le strutture e le definizioni seguenti sono riportate nel file di include della DLL (CndexLinkUser.h e CndexLink.bas).

```
#define ON 1
#define OFF 0

// Modo selezionato del Processo
#define MDI 1 // Modo MDI (Manual Data Input)
#define AUTO 2 // Modo Automatico
#define SEMI 3 // Modo Semi-Automatico (Blocco-Blocco)
#define MANJOG 4 // Modo Manuale Continuo
#define INCJOG 5 // Modo Manuale Incrementale
#define PROFILE 6 // Modo Ritorno Sul Profilo
#define HOME 7 // Modo Riferimento Assi

// Stato del Processo
#define IDLE 1
#define CYCLE 2
#define HOLDA 3
#define RUNH 4
#define HRUN 5
#define ERRO 6
#define WAIT 7
#define RESET 8
#define EMERG 9
#define INPUT 10

// Sotto-Stato del Processo
#define MAS 6
#define MBR 4

// Massimo numero di assi per un Processo
#define NUM_ELEM_SEL_AXI 9

// Selettore del tipo di Posizione Assi
#define PROGPOS 1 // Posizione Programmata
#define INTPOS 2 // Posizione Interpolata
#define TRANSDPOS 3 // Quota Trasduttore
#define ERRPOS 4 // Errore di Inseguimento
// Distance To Go

#define MAIN_PROGR_NAME_LEN 55
```




Documentazione CndexLinkUser DLL Interfaccia al server Cndex

```
////////////////////////////////////
// Logic variables identifier
////////////////////////////////////
// Boolean LOGIC variables
#define I_CODE      0
#define O_CODE      1
#define M_CODE      2
#define G_CODE      3
#define S_CODE      4
#define P_CODE      7
#define U_CODE      9
// WORD LOGIC variables
#define MW_CODE     20
#define GW_CODE     21
#define SW_CODE     22
#define PW_CODE     62
#define UW_CODE     63
// DOUBLE LOGIC variables
#define MD_CODE     40
#define GD_CODE     41
#define PD_CODE     43
#define UD_CODE     44
// TEXT LOGIC variables
#define A_CODE      45
// Double PROCESS variables
#define E_CODE      46
#define SN_CODE     47
#define H_CODE      48
// TEXT PROCESS variables
#define SC_CODE     50
// SyncMotion variables
#define SYMO_W_CODE 100
#define SYMO_D_CODE 101
#define SYMO_A_CODE 102
#define SYMORET_W_CODE 103
#define SYMORET_D_CODE 104
```



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

```
// Tables identifiers
#define AXIS_TABLE_ID 1 // Identificativo della Tabella Assi
#define TOOL_TABLE_ID 2 // Identificativo della Tabella Utensili
#define OFFSET_TABLE_ID 3 // Identificativo della Tabella Offset
#define USER_TABLE_ID 4 // Identificativo della Tabella User

//Logical directory identifiers
#define PPDIR_PROGRAM 0 //PROGRAM Directory
#define PPDIR_USER 1 //USER Directory
#define PPDIR_SYSTEM 2 //SYSTEM Directory
#define PPDIR_OEM 3 //OEM Directory

#pragma pack(push, VB_requests_pack_4, 4)

struct GETINTDATA_C4
{
    BYTE AxisName; // Nome ASCII dell'asse
    BYTE mode; //
    double position; // Posizione attuale
    double TotalOffset; // Offset Totale
};

struct PROCDATA_C4
{
    WORD Mode; // Modo selezionato
    WORD Status; // Stato del Processo
    WORD SubStatus; // Sotto-Stato del Processo
};

struct GETINFO1DATA_C4
{
    BYTE status; // Stato del Processo
    BYTE substatus; // Sotto-Stato del Processo
    BYTE mode_select; // Modo selezionato
    BYTE jog_dir; // Direzione movimenti manuali (1 = negativi, 0 =
positiv1)
    WORD speed_ov; // Spindle Speed Override (moltiplicato 100)
    WORD feed_ov; // Feed Rate Override (moltiplicato 100)
    WORD rap_feed_ov; // Rapid Feed Override (moltiplicato 100)
    WORD man_feed_ov; // Manual Feed Override (moltiplicato 100)
    double jog_incr; // Valore dell'incremento per il
// modo Manuale Incrementale
    BYTE num_ax_sel; // Numero di assi selezionati
    BYTE ax_sel[NUM_ELEM_SEL_AXI]; // Array degli assi selezionati
    double actual_tool; // Attuale utensile caricato
    double progr_tool; // Attuale Utensile programmato
    WORD actual_tool_off; // Attuale numero offset caricato
    WORD progr_tool_off; // Attuale numero offset programmato
    double real_speed; // Velocita' mandrino reale
    double progr_speed; // Velocita' mandrino programmata
    double progr_feed; // Velocita' programmata il continuo
    double rapid_feed; // Velocita' di rapido
}
```



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

```
double real_feed;    // Velocita' reale
WORD feed_mis_unit;  // Unita' di misura delle velocita':
                    //0 = inch senza G95
                    //1 = mm senza G95
                    //2 = inch con G95
                    //3 = mm con G95
BYTE main_progr_name[MAIN_PROGR_NAME_LEN]; //Part Program principale attivo
BYTE dry_run;        // 1 = modalita' dry run attiva
BYTE rapid_override; // 1 = rapidi limitati attivi
BYTE disable_slashed_blk; // 1 = Disattivazione Blocchi Barrati attivo
BYTE optional_stop;  // 1 = Forzatura stop su M00 attiva
BYTE force_rapid_feed; // 1 = Forzatura velocita' di rapido attiva
BYTE auto_jog_ret;   // 1 = Ritorno automatico sul profilo e
                    // Movimenti Incrementali Automatici attivi
BYTE block_retrace;  // 1 = multi block retrace attivo
WORD last_nc_error;  // Ultimo errore di Processo. 0 = nessun errore
BYTE free[22];       // campi riservati
};

struct GETINFO2DATA_C4
{
    WORD StatusWORD; //status word for changed fields
    double Urp;      //plane rotation angle value
    double ActTool;  //actual tool number
    WORD ActOffset;  //actual tool offset number
    double ProgTool; //programmed tool number
    WORD ProgOffset; //programmed tool offset number
    WORD M_Status;   //status for highlighted M codes
    WORD M_Value[16]; //value of the 16 M codes
    BYTE Ax1Name;    //axis 1 name
    double Ax1Offset; //axis 1 tool offset value
    BYTE Ax2Name;    //axis 2 name
    double Ax2Offset; //axis 1 tool offset value
    double Radius;   //radius of the active tool
};

struct MARKER_INFO_C4
{
    short Number; // Numero di part program attivi
    unsigned long LineNum[MAX_LEVEL]; // Numero di linea dei part program attivi
    unsigned long NumBlks[MAX_LEVEL]; // Numero di blocco dei part program
attivi
    char MarkerName[MAX_LEVEL][LABEL_LEN+1]; // Marker dei part program
attivi
    unsigned char PPname[MAX_LEVEL][MAX_LEN_BLK+5]; //Nome del part program
attivo
    unsigned long breakVal; // Valore della variabile di breakpoint ($BRK)
    unsigned long free[8]; // Spazio riservato
};

struct GETBLKNUMDATA_C4
{
    WORD ppActNum; // Numero di programmi attivi
    unsigned long MainActBlk; // Numero di blocco del Part Program principale se
attivo
    unsigned long Sbr1ActBlk; // Numero di blocco della Subroutine 1 se attiva
}
```



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

```
unsigned long  Sbr2ActBlk; // Numero di blocco della Subroutine 2 se attiva
unsigned long  Sbr3ActBlk; // Numero di blocco della Subroutine 3 se attiva
unsigned long  Sbr4ActBlk; // Numero di blocco della Subroutine 4 se attiva
};

// Series10 axis table descriptor
struct AXIS_TABLE_C4
{
    WORD        ax_owner;    // Codice dell'ambiente proprietario dell'asse
    WORD        ax_name;     // Nome ASCII dell'asse
    double      ax_orig;     // valore dell'origine in uso
    double      free1;       // campo riservato
    double      ax_ofg92;    // offset G92
    double      ax_toff;     // Offset utensile attuale
    double      free2;       // campo riservato
    double      ax_offset;   // valore totale dell'offset
    double      orig1;       // origine 1
    double      orig2;       // origine 2
    double      orig3;       // origine 3
    double      orig4;       // origine 4
    double      orig5;       // origine 5
    double      orig6;       // origine 6
    double      orig7;       // origine 7
    double      orig8;       // origine 8
    double      orig9;       // origine 9
    double      orig10;      // origine 10
    WORD        riservati[2]; // campo riservato
};

// Series10 tool table descriptor
struct TOOL_TABLE_C4
{
    double      tcode;       // Numero utensile
    WORD        pocket;      // posizione nel magazzino utensili
    WORD        tfamcol;     // codice famiglia utensili
    WORD        tclass;      // classe utensile
    WORD        tstatus;     // status WORD
    WORD        tcntrl;      // control WORD
    double      maxtime;     // vita utensile totale
    double      remtime;     // vita utensile residua
    double      tuser1;      // parametro 1 utente
    double      tuser2;      // parametro 2 utente
    double      tuser3;      // parametro 3 utente
    double      tuser4;      // parametro 4 utente
    WORD        tolnfr;      // numero offset associato all'utensile
};

// Series10 offset table descriptor
struct OFFSET_TABLE_C4
{
    double      tactl1;      // lunghezza 1 utensile attuale
    double      tcmaxl1;     // massima variazione lunghezza 1 utensile
    double      tcactl1;     // attuale variazione lunghezza 1 utensile
    double      tactl2;      // lunghezza 2 utensile attuale
    double      tcmaxl2;     // massima variazione lunghezza 2 utensile
    double      tcactl2;     // attuale variazione lunghezza 2 utensile
};
```



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

```
double      tdiameter; // diametro
double      tcacdiam;  // attuale variazione diametro
WORD        torient;   // orientamento punta utensile
};

// Series10 user table descriptor
struct USER_TABLE_C4
{
    double    user1;      // variabile 1 utente
    double    user2;      // variabile 2 utente
    double    user3;      // variabile 3 utente
    double    user4;      // variabile 4 utente
};

// OPENcontrol tool table descriptor
struct TOOL_TABLE_II_C4
{
    char      ToolName [TOOL_ASCII_LEN]; // Nome utensile
    WORD      Status;    // Stato utensile
    WORD      LifeType;   // Tipo gestione vita
    double    MaxLife;    // Vita iniziale
    double    RemLife;    // Vita rimanente
    WORD      OffsNum;    // Numero di offset di default
    WORD      ExternType; // Tipo gestione esterna (..Magazzini)
    WORD      ExternInd;  // Indice per gestione esterna (..)
    double    Duser [NUM_DUSER]; // Parametri User double
    short     Suser [NUM_SUSER];  // Parametri user short
    WCHAR     Descr [DESCR_LEN];   // Commento con /0
};

struct OFFSET_VAL_II_C4
{
    double    ValOrig;    // Valore originale utensile
    double    MaxChangeVal; // Massima variazione possibile
    double    ActChangeVal; // Variazione corrente
};

// OPENcontrol offset table descriptor
struct OFFSET_TABLE_II_C4
{
    struct OFFSET_VAL_II_C4 LenVal [OFFS_AX]; // Lunghezze
    struct OFFSET_VAL_II_C4 DiaVal [DIAM_AX]; // Diametri / Raggetto
    WORD      Orient;     // Orientamento
    WORD      ExternType;  // Tipo gestione esterna (..)
    WORD      ExternInd;   // Indice per gestione esterna (..)
    double    Duser [NUM_DUSER]; // Parametri User double
    short     Suser [NUM_SUSER];  // Parametri user short
    WCHAR     Descr [DESCR_LEN]; // Commento con /0
};
```



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

```
// OPENcontrol origin table descriptor
struct ORIGIN_TABLE_II_C4
{
    double AxisVal [MAX_NUM_AX_XTEND]; // Valore origine per ogni asse
    WORD ExternType; // Tipo gestione esterna (..)
    WORD ExternInd; // Indice per gestione esterna (..)
    WCHAR Descr [DESCR_LEN]; // Commento con /0
};

// OPENcontrol user table descriptor
struct USER_TABLE_II_C4
{
    double UserVal [4]; // user variable (1 - 4)
};

struct MAGAZINE_TABLE_II_C4
{
    WORD Type; // Tipo del magazzino
    WORD NumPockets; // Numero di pockets del magazzino
    WORD RowPockets; // Numero di pocket per riga (planari)
    WORD ExternType; // Tipo gestione esterna (..)
    WORD ExternInd; // Indice per gestione esterna (..)
    double Duser [10]; // Parametri User double
    WORD Suser [10]; // Parametri user short
    WCHAR Descr [DESCR_LEN]; // Commento con /0
};

struct POCKET_TABLE_II_C4
{
    WORD Status; // Pocket free or occupied
    WORD isRandom; // assigned to a fixed tool or not
    WORD Type; // Pocket type (master, bulk ... )
    WORD Class; // Pocket class
    WORD ToolInd; // Indice del tool caricato in pocket
    WCHAR Descr [DESCR_LEN]; // Commento con /0
};

struct ERR_MSG_C4
{
    unsigned char Msg1[40]; // prima riga del messaggio
    unsigned char Msg2[40]; // seconda riga del messaggio
    unsigned char Msg3[40]; // terza riga del messaggio
    unsigned char Msg4[40]; // quarta riga del messaggio
};
```



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

```
struct PLVARDESC_C4
{
    unsigned short Code; // Codice identificativo della variabile
                        // vedi "Logic variables identifier"
                        // in questo stesso paragrafo
    unsigned short Index; // Indice della variabile
    unsigned short Bit; // Indice del Bit (0 - 15) della variabile. Questo
                        // campo e' utilizzato solo in scrittura e quando
                        // Code identifica una variabile booleana.
};

struct MSG_ERROR_C4
{
    DWORD BootID;
    SYSTEMTIME_CNDEX_C4 SystemTime; // - Istante di generazione
    DWORD UnIdSeq; // - Numero di sequenza (da accensione)
    DWORD Code_Err; // - Codice Errore
    short Process; // - Processo
    char Comando; // - Comando in errore
    char SubCom; // - Sottocomando in errore
    BYTE FormatTxt[148]; // - Dati aggiuntivi per errore
};

struct MSG_EMERGENCY_C4
{
    DWORD BootID;
    SYSTEMTIME_CNDEX_C4 SystemTime; // - Istante di generazione
    DWORD UnIdSeq; // - Numero di sequenza (da accensione)
    DWORD Code_Err; // - Codice Emergenza
    short Process; // - Processo
    short Proc_Err; // - Processi coinvolti
    BYTE FormatTxt[148]; // - Dati aggiuntivi per emergenza
};
```



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

```
struct MSG_LOGS_C4
{
    DWORD BootID;
    SYSTEMTIME_CNDEX_C4 SystemTime;    // - Istante di generazione
    DWORD UnIdSeq;                     // - Numero di sequenza (da accensione)
    DWORD Code_Log;                     // - Codice Log
    DWORD Err_info;                     // - Log con associato errore
    short Proc_Err;                     // - Processi coinvolti
    BYTE  FormatTxt[148];               // - Dati aggiuntivi per log
};
```

```
struct MSG_ANOMALY_C4
{
    DWORD BootID;
    SYSTEMTIME_CNDEX_C4 SystemTime;    // - Istante di generazione
    DWORD UnIdSeq;                     // - Numero di sequenza (da accensione)
    DWORD Code_Err;                     // - Codice Errore
    DWORD Linea;                        // - Linea
    BYTE  FormatTxt[148];               // - Dati aggiuntivi per errore
};
```

```
struct REMAP_DEF_C4
{
    unsigned char VarType;              // 0: bool; 1: byte; 2: WORD; 3: DWORD
    unsigned short PhysAddr;             // indirizzo fisico
    unsigned char PhysBit;               // bit fisico
    unsigned short LogicAddr;            // indirizzo logico
    unsigned char LogicBit;              // bit logico
    unsigned char Mode; // 0: ignore; 1: copy; 2: set to true; 3: set to false;
                        // 4: negate; 5: set to value
    unsigned long RiseTime;              // tempo di salita (us)
    unsigned long FallTime;              // tempo di discesa (us)
    unsigned long Value;                 // valore
    unsigned long IOMsgId;               // id del messaggio
    unsigned char IOClass;               // classe

    unsigned long CurrPhysValue;          // valore fisico attuale
    unsigned long CurrLogicValue;         // valore logico attuale
};
```

```
struct AX_ORIG_NUM_C4
{
    unsigned char AxisName;              //Nome asse
    unsigned char OriginNumber;          //Numero dell'origine attualmente in uso
    unsigned short OriginFlag;           //Bit mask del tipo di origine:
                                        // bit 0 - origine assoluta (attivata con UAO)
                                        // bit 1 - origine temporanea (attivata con UTO)
                                        // bit 2 - origine incrementale(attivata con UIO)
};
```




Documentazione CndexLinkUser DLL Interfaccia al server Cndex

Gestione degli errori

In tutte le funzioni gli ultimi due parametri riportano i codici degli eventuali errori:

Il primo (pErrClass) identifica la classe di appartenenza dell'errore, il secondo (pErrNum) riporta il numero dell'errore vero e proprio.



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

Classe degli errori

Gli errori sono stati suddivisi in queste classi:

Classe di errore **COM** (classe 1)

Gli errori che appartengono a questa classe sono causati dall'interfaccia COM utilizzata per comunicare con il server Cndex.

Classe di errore **SERVER** (classe 2)

Gli errori che appartengono a questa classe possono essere causati da una serie preliminare di controlli che il server Cndex esegue sui parametri delle funzioni chiamate, oppure da malfunzionamenti interni del server stesso.

Classe di errore **NETBIOS** (classe 3)

Gli errori che appartengono a questa classe sono causati da anomalie o errori nello scambio di dati tra server Cndex e controlli numerici con cui sono attive sessioni di comunicazione sulla rete locale.

Classe di errore **CNC** (classe 4)

Gli errori che appartengono a questa classe sono causati dall'elaborazione dei dati da parte dei controlli numerici Serie 10 connessi al server Cndex.

Classe di errore **FILESYS** (classe 5)

Questi errori sono generati a livello server (Cndex) da una chiamata a funzione di gestione del logical file system (LogFS).

Classe di errore **FILESYS_ERR** (classe 6)

Questi errori sono generati dal sistema operativo dove è in esecuzione il server cndex quando una chiamata a funzione del logical file system incontra un errore nell'accesso al file system fisico.

Classe di errore **CNC_BOOT** (classe 7)

Questi errori sono generati dal server Cndex a fronte di un problema durante l'avvio del controllo numerico OPENControl

Classe di errore **CNDEXLINK_DLL** (classe 9)

Contiene i codici di errore generate da CndexLink.dll.

Classe di errore **DLL_INTERFACE** (classe 10)

I codici di questa classe sono generati da CndexLink user library API, ovvero dalle funzioni descritte nel presente documento.

Classi di errore **OPENcontrol** (classi da 17 a 62)

Questi errori sono generati dal software OPENcontrol. Per una descrizione dettagliata si prega di consultare la documentazione utente di OPENcontrol.



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

Codici di errore COM (classe 1)

Questo è un elenco degli errori COM più comuni. Per i codici non inclusi nell'elenco si prega di consultare la documentazione MSDN.

CO_E_SERVER_EXEC_FAILURE (0x80080005)	Su Windows 95/98 il server non può essere lanciato automaticamente ma bisogna eseguirlo a mano, facendo attenzione che quando non c'è più nessuna applicazione attiva il server viene terminato automaticamente.
RPC_S_SERVER_UNAVAILABLE (0x800706ba)	Verificare che il CNC sia acceso e collegato alla LAN. Assicurarsi che nel registry il valore del parametro <enableremoteconnect> sia Y. Aggiungere o attivare il protocollo TCP/IP nella configurazione di rete
RPC_S_CALL_FAILED (0x800706be)	Verificare che il CNC sia acceso e collegato alla LAN. Verificare tramite BootController che il CNC sia raggiungibile.
E_ACCESSDENIED (0x80070005)	Verificare che DCOM sia abilitato sul sistema e che l'applicativo inizializzi correttamente le opzioni di sicurezza di DCOM. Per maggiori dettagli si consulti il documento DCOM_WindowsXP_Configuration.pdf.
REGDB_E_CLASSNOTREG (0x80040154)	Il server Cndex non è stato registrato nel registry di Windows. Registrarlo con questo comando da eseguire in un DOS shell : cndex.exe /regserver
CO_S_NOTALLINTERFACES (0x00080012)	La DLL proxy del server Cndex non è stata registrata nel registry di Windows. Registrarla con questo comando da eseguire in un DOS shell: regsvr32 cndexps.dll
RPC_E_DISCONNECTED (0x80010108)	Il server Cndex si è disconnesso dal client. Le disconnessioni sono solitamente dovute a errori di timeout o errori di rete.
RPC_E_SERVERFAULT (0x80010105)	Il server Cndex ha sollevato un'eccezione. Contattare il supporto tecnico Prima Electro.
RPC_X_NULL_REF_POINTER (0x800706f4)	Puntatore a riferimento nullo passato allo stub. Verificare che tutti gli argomenti di tipo puntatore passati alla funzione siano non nulli.
RPC_E_CANTCALLOUT_INEXTERNALCALL (0x80010005)	Una chiamata ad una funzione Cndex o ad una funzione Com in genere è stata inserita nell'handler del messaggio WM_PAINT. Rimuovere la funzione che genera l'errore dal message handler.
OR_INVALID_OXID (0x80070776)	Questo errore può verificarsi se si modifica la configurazione di una o più schede di rete del CNC e non si esegue il riavvio di Windows CE. Riavviare Windows CE. Se il problema persiste contattare il supporto tecnico PrimaElectro.



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

Codici di errore del server Cndex (classe 2)

Memoria insufficiente per le allocazioni dinamiche	1
Impossibile creare gli eventi per la sincronizzazione	2
Sessione abortita e non piu' utilizzabile	3
Sessione non aperta	4
Impossibile allocare un canale	5
Richiesta una funzione di un processo non esistente	6
Comando di broadcasting abortito	7
Il buffer fornito dalla funzione su cui vengono copiati i dati forniti dal CN e' troppo piccolo	8
La sessione e' gia' aperta	9
Lista di broadcasting non valida	10
Comando di realtime abortito	11
Funzione gia' attiva	12
Funzione non ancora attiva	13
Thread di ricezione terminato	14
Non e' stata ricevuta una risposta al comando entro un certo tempo	15
La release del CN a cui si e' connessi non e' compatibile con il colloquio con il server	16
Il cookie non identifica nessuna sessione di comunicazione	17
Impossibile creare il thread di realtime	18
Non ci sono piu' sessioni disponibili	19
Errore nell'acquisizione dei simboli	20
Impossibile creare un'istanza di un oggetto interno	23
Impossibile creare il thread di broadcast	24
La funzione non puo' essere eseguita in questa fase di boot del controllo numerico	25
Parametro errato	26
Buffer non valido	27
L'identificativo della sessione (UserSession) non e' valido (sessione chiusa o mai aperta)	29
L'identificativo della sessione non e' valido (valore fuori ranged) oppure la sessione e' stata chiusa automaticamente dal server Cndex	31
Errore di scrittura del registry	32
Memoria esaurita	33
Funzione non disponibile (la fase di boot non e' corretta oppure c'è stato un errore di caricamento del SW minimo per l'esecuzione del comando)	34
Time-out attesa evento generico (usata in reboot)	35
Errore durante la registrazione del thread corrente come thread OPENControl	36
Errore lanciando o eseguendo un processo (applicazione) esterno al server	37
Funzione non implementata	38
DLL non trovata	39
L'entry point di una funzione in una DLL non è stato trovato	40
La definizione della memoria di interscambio con un altro applicativo ha generato errore	41
Si è verificato un errore non previsto	42
Il numero di AMP attivo contenuto nel file BootAmp.txt non è valido	43
Errore di lettura dal registry	44
Chiave di registro non presente	45
Errore nell'apertura di un file	46
Errore di lettura da file	47
Errore di scrittura su file	48
Nome CNC troppo lungo	49



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

Codici di errore del protocollo di rete NETBIOS (classe 3)

Illegal buffer length	0x0001
Illegal command	0x0003
Command timed out	0x0005
Message incomplete, issue another command	0x0006
Illegal buffer address	0x0007
Session number out of range	0x0008
No resource available	0x0009
Session closed	0x000a
Command cancelled	0x000b
Duplicate name	0x000d
Name table full	0x000e
No deletions, name has active sessions	0x000f
Local session table full	0x0011
Remote session table full	0x0012
Illegal name number	0x0013
No callname	0x0014
Cannot put * in NCB_NAME	0x0015
Name in use on remote adapter	0x0016
Name deleted	0x0017
Session ended abnormally	0x0018
Name conflict detected	0x0019
Interface busy, IRET before retrying	0x0021
Too many commands outstanding, retry later	0x0022
Ncb_lana_num field invalid	0x0023
Command completed while cancel occurring	0x0024
Command not valid to cancel	0x0026
Name defined by another local process	0x0030
Environment undefined. RESET required	0x0034
Required OS resources exhausted	0x0035
Max number of applications exceeded	0x0036
No saps available for netbios	0x0037
Requested resources are not available	0x0038
Invalid ncb address or length > segment	0x0039
Invalid NCB DDID	0x003B
Lock of user area failed	0x003C
NETBIOS not loaded	0x003f
System error	0x0040

Gli errori in grassetto sono quelli che si verificano con maggior frequenza.

L'errore "No callname" (0x0014) si verifica quando non e' possibile attivare la connessione con un controllo numerico.

L'errore "Session ended abnormally" (0x0018) si verifica quando la sessione di comunicazione viene persa a causa di un problema di collegamento in rete o perché viene spento o riavviato via software il controllo numerico.



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

Codici di errore Serie 10 (classe 4)

Codici di errore per funzioni Real Time

Comando sconosciuto	0x0101
Nessun canale disponibile	0x0102
Tick richiesto non multiplo del tick di sistema	0x0103
Id canale errato	0x0104
Acquisizione dati ancora in corso	0x0105
Canale non configurato	0x0106
Errore su stop trigger	0x0107
Canale gia' configurato per altri tipi di dati	0x0108

Codici di errore per funzioni Dry Run

Processo non configurato	0x0200
Asse non presente nel processo	0x0201
Dry run non configurato	0x0202
Dry run gia' in esecuzione	0x0203
Dry run gia' in stop	0x0204

Codici di errore delle funzioni del Processo

Fare riferimento al Manuale di Programmazione della Serie 10.



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

Codici di errore FILESYS_CLASS (classe 5)

ERROR_OPENING_TRANSACTION 1 (0x1)	Si è verificato un errore nell'apertura di una transazione con in target per il trasferimento di un file.
ERROR_CLOSING_TRANSACTION 2 (0x2)	Si è verificato un errore nella chiusura di una transazione con in target per il trasferimento di un file.
WRONG_FILEID 3 (0x3)	File ID non valido.
WRONG_TRANSACTION 4 (0x4)	Numero di transazione non valido.
ERROR_OPENING_FILE 5 (0x5)	Errore nell'apertura di un file.
ERROR_CLOSING_FILE 6 (0x6)	Errore nella chiusura di un file.
WILDCARD_NOTALLOWED 7 (0x7)	Un percorso file contenente wildcard ("*" o "?") è stato passato ad una funzione che non supporta le wildcard.
ERROR_READING_DATA 8 (0x8)	Errore nella lettura dei dati da file.
ERROR_WRITING_DATA 9 (0x9)	Errore nella scrittura dei dati su file.
DRIVE_NOT_FOUND 10 (0xA)	Il drive logico non è presente sul target.
WRONG_PATH_NAME 12 (0xC)	Il percorso specificato non è valido.
DRIVE_NAME_ALREADY_EXISTS 13 (0xD)	Si è tentato di creare un drive logico sul target con lo stesso nome di un drive logico già esistente.
WRONG_SECURITY_LEVEL 14 (0xE)	La funzione invocata richiede un livello di sicurezza più elevato sul target.
NO_ADD_DRIVE_ON_PC 15 (0xF)	La creazione di un drive logico su un PC remote non è consentita per ragioni di sicurezza.
INSUFFICIENT_DISK_SPACE 16 (0x10)	Spazio su disco insufficiente per completare l'operazione richiesta.
FILE_NOT_FOUND 17 (0x11)	File non trovato.



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

Codici di errore FILESYS_ERRNUM_CLASS (classe 6)

Quando il file system fisico genera un errore, il server Cndex restituisce un errore di classe 6. Solamente le funzioni di gestione del file system logico restituiscono errori di classe 6. Il numero di errore corrisponde esattamente al codice di errore restituito dal sistema operativo del target, che può essere Windows CE oppure un sistema operativo Windows per desktop. Si consulti [la documentazione MSDN relativa ai codici di errore di sistema](#).



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

Codici di errore CNC_BOOT_CLASS (classe 7)

ERROR_LOADING_IPC_DLL 1 (0x1)	Errore durante il caricamento della modulo IPC del controllo.
ERROR_LOADING_TABLE_DLL 2 (0x2)	Errore durante il caricamento della DLL di gestione tabelle e del system history.
ERROR_LOADING_LOADER_DLL 3 (0x3)	Errore durante la fase di avvio del controllo numerico.
ERROR_CNC_SHUT_DOWN 4 (0x4)	
ERROR_CNC_INIT 16 (0x10)	Errore durante il caricamento delle routine necessarie all'avvio del controllo numerico.
ERROR_CNC_HALT_BOOT 19 (0x13)	Errore di connessione al controllo numerico. Controllo numerico in stato di errore del loader.



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

Codici di errore SERVER_EXCEPTION_CLASS (classe 8)

Questa classe di errore indica che si è verificata un'eccezione durante l'esecuzione della procedura remota all'interno del CNC. Il numero di errore corrisponde all'errore restituito dalla API Windows CE GetLastError(). Ad esempio, un errore di classe 8 numero 0xC0000005 indica che è stata sollevata un'eccezione di access violation durante l'esecuzione della funzione invocata.



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

Codici di errore CNDEXLINK_DLL_ERR_CLASS (classe 9)

ERR_INTERFACE_NOT_AVAILABLE 1 (0x1)	È stata chiamata una funzione non supportata dal target.
ERR_CE_USED_AS_A_NETBIOS_ROUTER 2 (0x2)	Si è tentato di connettersi ad un sistema Serie10 utilizzando un sistema windows CE come Cndex server/router. Solo un sistema che supporta NetBEUI può essere utilizzato a questo scopo.
ERR_OPENING_LOCAL_FILE 3 (0x3)	Si è verificato un errore nell'apertura di un file locale durante una chiamata ad una funzione del logical file system (LogFS).
ERR_READING_LOCAL_FILE 4 (0x4)	Si è verificato un errore nella lettura di un file locale durante una chiamata ad una funzione del logical file system (LogFS).
ERR_WRITING_LOCAL_FILE 5 (0x5)	Si è verificato un errore nella scrittura dati in un file locale durante una chiamata ad una funzione del logical file system (LogFS).
ERR_CLOSING_LOCAL_FILE 6 (0x6)	Si è verificato un errore nella chiusura di un file locale durante una chiamata ad una funzione del logical file system (LogFS).
ERR_NO_MEMORY_AVAILABLE 7 (0x7)	L'allocazione di un buffer di memoria è fallita a causa di memoria non disponibile.
ERR_READING_LOCAL_FILE_SIZE 8 (0x8)	Si è verificato un errore nella lettura della dimensione di un file locale durante una chiamata ad una funzione del logical file system (LogFS).
ERR_INSUFFICIENT_DISK_SPACE 9 (0x9)	Si è verificato un errore di disco pieno durante la copia di un file locale in una chiamata ad una funzione del logical file system (LogFS).
ERR_SYNCHRONIZATION 10 (0xA)	Si è verificato un errore durante l'attesa della chiusura di un worker thread.
ERR_THREAD_CREATION 11 (0xB)	Si è verificato un errore durante la creazione di un worker thread.
ERR_NO_MEMORY 12 (0xC)	Si è verificato un errore durante la creazione di un'area dati condivisa con un worker thread.
ERR_CNDEX_SERVER_NOT_CREATED 13 (0xD)	Il server Cndex associate all'ID di sessione corrente non è stato creato.
ERR_DURING_SERVER_LOCK 14 (0xE)	Si è verificato un errore in un tentativo di accesso esclusivo al server Cndex.
ERR_NETBIOS_NAME_TOO_LONG 15 (0xF)	Si è tentato di connettersi ad un sistema Serie10 utilizzando un nome CN più lungo di 15 caratteri.



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

Codici di errore DLL_INTERFACE_ERR_CLASS (classe 10)

ERR_SERVER_ALREADY_CREATED 1 (0x1)	Il server è stato creato più di una volta.
ERR_CREATING_SERVER_OBJECT 2 (0x2)	Si è verificato un errore durante la creazione del server Cndex.
ERR_SERVER_NOT_CREATED 3 (0x3)	E' stata chiamata una funzione senza aver creato il server Cndex.
ERR_INVALID_PARAMETER 4 (0x4)	Uno o più parametri in input alla funzione non sono validi.
ERR_OPTION_NOT_ENABLED 5 (0x5)	L'opzione A06 "CndexLink communication" per la comunicazione in rete con applicativi esterni non è abilitata sul CNC che si sta cercando di connettere.
ERR_UAS_DISABLED_WHEN_RCM_ON 6 (0x6)	Una richiesta di Dry Run non può essere accettata mentre è in corso una Ricerca Memorizzata.
ERR_OUT_OF_MEMORY 7 (0x7)	Impossibile allocare un buffer di memoria.
ERR_BUFFER_TOO_SMALL 8 (0x8)	Il buffer dati passato alla funzione è troppo piccolo per contenere i dati richiesti.



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

Codici di errore SOAP_INTERFACE_ERR_CLASS (classe 11)

La classe 11 indica errori nella comunicazione tramite protocollo SOAP rilevati dal lato client. La causa dell'errore potrebbe trovarsi sia nel client che nel server. Ad esempio, un errore di tipo SOAP_TCP_ERROR restituito da una chiamata a funzione SOAP potrebbe indicare che il CNC è stato riavviato, è stato spento, oppure si è verificato un errore non recuperabile nel CNC.

Codice	Simbolico	Descrizione
0	SOAP_OK	No error
1	SOAP_CLI_FAULT	The service returned a client fault (SOAP 1.2 Sender fault)
2	SOAP_SVR_FAULT	The service returned a server fault (SOAP 1.2 Receiver fault)
3	SOAP_TAG_MISMATCH	An XML element didn't correspond to anything expected
4	SOAP_TYPE	An XML Schema type mismatch
5	SOAP_SYNTAX_ERROR	An XML syntax error occurred on the input
6	SOAP_NO_TAG	Begin of an element expected, but not found
7	SOAP_IOB	Array index out of bounds
8	SOAP_MUSTUNDERSTAND	An element needs to be ignored that need to be understood
9	SOAP_NAMESPACE	Namespace name mismatch (validation error)
10	SOAP_USER_ERROR	User error (reserved for soap.user usage)
11	SOAP_FATAL_ERROR	Internal error
12	SOAP_FAULT	An exception raised by the service
13	SOAP_NO_METHOD	The dispatcher did not find a matching operation for a request
14	SOAP_NO_DATA	No data in HTTP message
15	SOAP_GET_METHOD	HTTP GET operation not handled, see Section 19.10
20	SOAP_EOM	Out of memory
21	SOAP_MOE	Memory overflow/corruption error (DEBUG mode)
23	SOAP_NULL	An element was null, while it is not supposed to be null
24	SOAP_DUPLICATE_ID	Element's ID duplicated (multi-ref encoding)
25	SOAP_MISSING_ID	Element ID missing for an href/ref (multi-ref encoding)
26	SOAP_HREF	Reference to object is incompatible with the object referred to
27	SOAP_UDP_ERROR	Message too large to store in UDP packet
28	SOAP_TCP_ERROR	A connection error occurred
29	SOAP_HTTP_ERROR	An HTTP error occurred
30	SOAP_SSL_ERROR	An SSL error occurred
31	SOAP_ZLIB_ERROR	A Zlib error occurred
32	SOAP_DIME_ERROR	DIME formatting error or DIME size exceeds SOAP_MAXDIME_SIZE
33	SOAP_DIME_HREF	DIME attachment has no href from SOAP body (and no DIME callbacks were defined to save the attachment)
34	SOAP_DIME_MISMATCH	DIME version/transmission error
35	SOAP_DIME_END	End of DIME attachments protocol error
36	SOAP_MIME_ERROR	MIME parsing error
37	SOAP_MIME_HREF	MIME attachment has no href from SOAP body error
38	SOAP_MIME_END	End of MIME attachments protocol error
39	SOAP_VERSIONMISMATCH	SOAP version mismatch or no SOAP message
40	SOAP_PLUGIN_ERROR	Failed to register plugin
41	SOAP_DATAENCODINGUNKNOWN	SOAP 1.2 DataEncodingUnknown fault



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

Codice	Simbolico	Descrizione
42	SOAP_REQUIRED	Attributed required validation error
43	SOAP_PROHIBITED	Attributed prohibited validation error
44	SOAP_OCCURS	Element minOccurs/maxOccurs validation error or SOAP_MAXOCCURS exceeded
45	SOAP_LENGTH	Element length validation error or SOAP_MAXLENGTH exceeded
46	SOAP_FD_EXCEEDED	Too many open sockets (for non-win32 systems not supporting poll())
47	SOAP_UTF_ERROR	An UTF-encoded message decoding error occurred
48	SOAP_NTLM_ERROR	An NTLM authentication handshake error occurred
#N/D	SOAP_LEVEL	XML nesting depth level exceeds SOAP_MAXLEVEL
EOF (-1)	SOAP_EOF	Unexpected end of file, no input, or timeout receiving data
EOF (-1)	SOAP_ERR	Error (for internal use)



Documentazione CndexLinkUser DLL Interfaccia al server Cndex

Codici di errore OPENcontrol (classi da 17 a 62)

Questi errori sono generati dal software OPENcontrol. Per una descrizione dettagliata si prega di consultare la documentazione utente di OPENcontrol.

Classe	ID	Descrizione
IPCEclass	17 (0x11)	Classe di errore IPC
CNSEclass	18 (0x12)	Classe di errore Consoles
EMGEclass	19 (0x13)	Classe di errore Emergenze
IOEclass	20 (0x14)	Classe di errore IO
PLCEclass	21 (0x15)	Classe di errore PLC
GMCEclass	22 (0x16)	Classe di errore GMC
CNCEclass	23 (0x17)	Classe di errore CNC
OSWEclass	24 (0x18)	Classe di errore OSWire
TBLEclass	25 (0x19)	Classe di errore Tabelle
GEOEclass	26 (0x1A)	Classe di errore Geometria
SPLEclass	27 (0x1B)	Classe di errore Spline
LODEclass	31 (0x1F)	Classe di errore LOADER
SEREclass	32 (0x20)	Classe di errore SERVO
SEROSWcls	33 (0x21)	Classe SERVO OSWIRE
SERDSIcls	34 (0x22)	Classe SERVO SERCOS
SERCANcls	35 (0x23)	Classe SERVO CANOPEN
SERSENScls	36 (0x24)	Classe SERVO SENSORE
SERCATcls	37 (0x25)	Classe SERVO EtherCAT
SERMECcls	38 (0x26)	Classe SERVO Mechatrolink
MONEClass	48 (0x30)	Classe di errore MONITORING
SLIEClass	49 (0x31)	Classe di errore Serial Line
SECEClass	50 (0x32)	Classe di errore Security
SERCCClass	51 (0x33)	Classe DLL SERCOS
FLDBClass	52 (0x34)	Classe DLL FIELDBUS
CANEClass	53 (0x35)	Classe di errore dll canbus
ECATClass	54 (0x36)	Classe di errore dll ethercat
PROFIEClass	55 (0x37)	Classe di errore dll profibus
XMLEClass	56 (0x38)	Classe di errore dll XML
SOCKEClass	57 (0x39)	Classe di errore dll socket
CTRLEclass	58 (0x3A)	Classe di errore Controls
_3DEclass	59 (0x3B)	Classe di errore 3D
RETAEclass	60 (0x3C)	Classe di errore retain DB
MECHClass	61 (0x3D)	Classe di errore Mechatrolink
SOAPClass	62 (0x3E)	Classe di errore SOAP Server