

Objekt:	conf.ini - die Konfigurationsdatei des cyProviders
Ort:	<InstallationPath>\cyProvider\conf.ini Die Konfigurationsdatei "conf.ini" befindet sich im cyProvider-Installationsverzeichnis und enthält, neben den benötigten Parametern, die Verbindungsdetails zu cyProvidern mit einer Vertrauensbeziehung und die Liste aller aktiven DLL-Provider.
Sektion:	cyProvider-Konfigurationsparameter
Name:	[parameters] Sektion mit den cyProvider-Konfigurationsparametern.
Beispiel:	[parameters] rootpath=c:\cyProvider\Sources dllid=c:\cyProvider\cyLibraries tracelogfile=c:\cyProvider\Log\cyProviderTrace.log ticketlogfile=c:\cyProvider\Log\cyProviderTicket.log portnonssl=80 portssl=443 certpassword=#meinpwd providername=cyProvider displayname=cyProvider license= maxticketage=3600 generalticket=AuthProvider FileProvider trustid=#mytrustid trustedids=cyProviderX cyProviderY remoteticket=MsaProvider MailProvider
Variable:	rootpath
Werte:	<pathname> Die Variable "rootpath" benennt das Verzeichnis, in dem sich die Source-Files der Anwendungen oder die Filesystem-Tabellen befinden.
Variable:	dllid
Werte:	<pathname> Die Variable "dllid" benennt das Verzeichnis, in dem sich die Libraries des cyProviders befinden.
Variable:	tracelog
Werte:	<pathname>\<tracelogname> Die Variable "tracelogfile" bestimmt Ort und Namen der Tracelog-Datei. Nur wenn diese Variable in der "conf.ini" erscheint wird auch ein Logfile geschrieben.
Variable:	ticketlogfile
Werte:	<pathname>\<ticketlogname> Die Variable "ticketlogfile" bestimmt Ort und Namen der Ticketlog-Datei. Nur wenn diese Variable in der "conf.ini" erscheint wird auch ein Logfile geschrieben.
Variable:	portnonssl
Werte:	<portnr> Mit dieser Variablen kann eine vom Standardport 80 abweichende Portnummer für die http-Verbindung festgelegt werden.
Variable:	portssl
Werte:	<portnr> Mit dieser Variablen kann eine vom Standardport 443 abweichende Portnummer für die https-Verbindung festgelegt werden.
Variable:	certpassword
Werte:	#<password> Eingabe des Zertifikatspasswortes für die SSL-Verbindung in Klartext mit dem Präfix "#". Mit dem Neustart schreibt der cyProvider das Passwort verschlüsselt in die "conf.ini" zurück.
Variable:	providername
Werte:	<providername> Diese Variable bestimmt den Namen unter dem der cyProvider als Dienst installiert werden soll. Unter diesem Namen wird der Dienst in der Registry eingetragen.
Variable:	displayname
Werte:	<displayname> Diese Variable bestimmt den Anzeigenamen des cyProviders in der Windows Dienstverwaltung.

Variable:	license
Werte:	<licencekey> Über diese Variable wird der Lizenzschlüssel, der die Anzahl concurrent user beinhaltet, zugewiesen. Fehlt der Lizenzschlüssel startet der cyProvider im Single User Mode.
Variable:	maxticketage
Werte:	<sec> Die Variable bestimmt den Zeitraum in Sekunden in dem ein für den User ausgestelltes Ticket seine Gültigkeit nicht verliert.
Variable:	generalticket
Werte:	{AuthProvider FileProvider} Den unter "generalticket" aufgeführten Providern wird der Zugriff auf andere Provider erlaubt (interne Provider-Kommunikation).
Variable:	trustid
Werte:	#<trustid> <trustid> Eingabe einer Trustid in Klartext mit dem Präfix "#," wenn es sich um die eigene Trustid handelt. Mit dem Neustart schreibt der cyProvider das Passwort verschlüsselt in die "conf.ini" zurück. Die cyProvider, denen der aktuelle cyProvider vertraut, werden mit Hilfe der Variablen "trustedids" bestimmt. In der Sektion "trustedid" enthält die Variable die verschlüsselte "<trustid>" eines cyProviders, dem man vertraut.
Variable:	trustedids
Werte:	<cyProvider_1> ... <cyProvider_n> In "trustedids" werden Sektionen beschrieben, die Details zu den cyProvidern enthalten, denen dieser cyProvider vertraut. In Testumgebungen besteht die Möglichkeit per "trustedids=" generell allen cyProvidern zu vertrauen.
Variable:	remoteticket
Werte:	{MsaProvider MailProvider AdoProvider FileProvider} Den unter "remoteticket" aufgeführten Providern wird der Zugriff auf andere cyProvider erlaubt.
Sektion:	Details zu einer bestimmten Vertrauensbeziehung
Name:	[<trustedidname>] Sektion mit den Details der Vertrauensbeziehung zu einem anderen cyProvider.
Beispiel:	[cyProviderX] trustid=0sMF9E8EUQSNUsmo1WLexvg== Ip=88.79.143.129 192.168.178.22 Provider=MSA
Variable:	trustid
Werte:	#<trustid> <trustid> Eingabe einer Trustid in Klartext mit dem Präfix "#," wenn es sich um die eigene Trustid handelt. Mit dem Neustart schreibt der cyProvider das Passwort verschlüsselt in die "conf.ini" zurück. Die cyProvider, denen der aktuelle cyProvider vertraut, werden mit Hilfe der Variablen "trustedids" bestimmt. In der Sektion "trustedid" enthält die Variable die verschlüsselte "<trustid>" eines cyProviders, dem man vertraut.
Variable:	IP
Werte:	<IP_1> ... <IP_n> Aufzählung aller IP-Adressen mit denen ein Zugriff auf den anderen cyProvider erfolgen kann. Wichtig ist, dass alle IP-Adressen definiert werden. Beispielsweise die lokale Adresse, die VPN-Adresse, die öffentliche Adresse und falls mehrere Netzwerkkarten im Server sind, muss das für jede Karte wiederholt werden.
Variable:	Provider(trusted)
Werte:	{MSA ADO MAIL FILE} Mit dieser Variablen werden die DLL-Provider des Remote-cyProviders benannt, (nur die Kurzform d.h. MSA und nicht MsaProvider), die über TrustId zugreifen dürfen. Allerdings nutzt momentan nur der MsaProvider den Zugriff mit TrustIds.

Sektion:	Liste der benötigten DLL-Provider
Name:	[DLLProvider] Sektion mit den DLL-Providern, die beim Start des cyProviders geladen werden müssen..
Beispiel:	[DLLProvider] AdoProvider FileProvider AuthProvider MailProvider MSAProvider
Variable:	
Werte:	
Objekt:	AdoProvider.ini - die Konfigurationsdatei des AdoProviders
Ort:	<dllldir>\AdoProvider\AdoProvider.ini Die Konfigurationsdatei "AdoProvider.ini" befindet sich im AdoProvider-Verzeichnis und enthält die benötigten Parameter für die Datenbankzugriffe via ADO.
Sektion:	ADO
Name:	[ADO] In der ADO-Sektion werden die Sub(OLE)-Provider für den Zugriff auf die unterschiedlichen Datenbanken benannt. Zusätzlich wird der Gültigkeitszeitraum von ModifyLog-Einträgen festgelegt und das Ladevolumen von großen PAM-Feldern begrenzt.
Beispiel:	[ADO] Providers=DemoDB OraDB ModifyLogValid=3600 LargeMemoSize=512
Variable:	Providers
Werte:	{<providername_1> ... <providername_n>} Aufzählung aller Sub(OLE)-Provider, die der AdoProvider zur Verfügung stellen soll.
Variable:	ModifyLogValid
Werte:	<sec> "ModifyLogValid" definiert die maximale Lebensdauer eines Log-Satzes in Sekunden. Fehlt der Parameter, so wird als Defaultwert 0 gesetzt, d.h. Log-Sätze sind unbegrenzt gültig. Der AdoProvider verwendet folgendes Verfahren um Modifykonflikte aufzulösen. Alle Änderungen (Update und Delete) an Datensätzen werden in einer speicherinternen Tabelle _MLG (ModifyLogTable) mit eindeutigem Index, einer Benutzeridentifikation und dem Änderungsdatum gespeichert. Wird ein Datensatz in ein PAM oder in eine View mit "DropAllowedListen" eingelesen, nur in diesen Fällen sind Satzänderungen möglich, so wird das Selektionsdatum zwischengespeichert. Erfolgt nun eine Änderung an einem Datensatz, so prüft der AdoProvider mit Hilfe des Selektionsdatums und der MLG-Tabelle, ob zwischenzeitlich ein anderer Benutzer den Datensatz geändert hat. Ist dies der Fall erhält der Benutzer davon Kenntnis und kann die Änderung durchführen oder die Daten erneut abrufen. Die ModifyLogTable existiert nur für die Dauer der Laufzeit des cyProviders, dennoch kann sie recht groß werden. Um zu vermeiden, dass unnötig viel Speicherplatz vergeudet wird, besteht mit "ModifyLogValid" die Möglichkeit die Lebensdauer der ModifyLog-Sätze zu begrenzen. WICHTIG: Anwender, die ein PAM aufrufen, müssen die Änderung (Update/Delete) innerhalb der ModifyLogValid-Zeit ausführen. Andernfalls ergeht eine Warnung mit der Aufforderung das Formular neu zu laden und die Änderung zu wiederholen.
Variable:	LargeMemoSize
Werte:	<KB> "LargeMemoSize" ist eine Variable mit der die Ladezeit eines PAMs erheblich verringert wird. Einige PAMs enthalten Felder (Memo, Richedit, ListView oder Image) deren Datenvolumen riesig ist. Damit nicht der gesamte Datenbestand mit jedem Lesezugriff heruntergeladen wird, kann mit "LargeMemoSize" ein Ladelimit in KByte definiert werden. Ist der Bestand eines Datenfeldes größer als das Ladelimit, so wird im Feld ein Icon anstelle der Daten angezeigt. Der Anwender kann durch Doppelklick auf das Icon die Daten abrufen. Das Icon ist per Default ein "OPEN FOLDER Icon". Es steht im Verzeichnis "\GLOBALS\IMAGES" unter dem Namen "ImFolder.bmp" und kann bei Bedarf gegen ein anderes Icon ausgetauscht werden. Ab Version 5 wird diese Variable durch "LargeMemoLevel" in der "Serverdata.cdd" abgelöst.

Sektion:	Vorinitialisierungsphase des OLE-Providers
Name:	[<providename>.PreInit] In dieser Sektion können vor der Initialisierung einer Datenbankverbindung benötigte Datenbank-Parameter gesetzt werden.
Beispiel:	[OraDB.PreInit] ALTER SESSION SET NLS_NUMERIC_CHARACTERS=','
Variable:	
Werte:	
Sektion:	Datenbank-Connection des OLE-Providers
Name:	[ADO.<providename>] Diese Sektion enthält im Wesentlichen den "ConnectionString" und die Authentifizierungsdaten für die Anmeldung des OLE-Providers an der Datenbank.
Beispiel:	[ADO.DemoDB] ConnectionString=Provider=Microsoft.ACE.OLEDB.12.0; Data Source=C:\Programme\cyProvider\Data\demodb.mdb; UserID=%DatabaseAccount%; Password=%DatabasePassword%; DatabaseAccount=#myAccount DatabasePassword=#myPassword AllowExtProgs=False
Variable:	ConnectionString
Werte:	{<connectionstring>} Die Variable enthält den Verbindungsstring des OLE-Providers zur Datenbank. Enthält der String User und Passwort, sollten diese als Parameter (%DatabaseAccount%, %DatabasePassword%) im Verbindungsstring aufgenommen werden, damit die Accountdaten in den zusätzlichen Variablen auch verschlüsselt werden können.
Variable:	AllowExtProgs
Werte:	{False True} Für den Fall das externe Programme diesen OLE-Provider ebenfalls nutzen dürfen, muss die Variable auf True gesetzt werden.
Variable:	Filestore
Werte:	<pathname> Ein Filestore wird für jeden ADO Subprovider definiert. In dieses Verzeichnis werden Datenbankattribute gespeichert, die größer als der Wert des Parameters Backgroundlevel in der serverdata.ini sind. In der Datenbank selbst steht im Datenbankfeld eine Referenz zum Filestore.
Sektion:	AdoProvider Kompression
Name:	[Compression] Sektion in der die Komprimierungsparameter für den AdoProvider gesetzt werden können.
Beispiel:	[Compression] Mode=1 Level=4 Blocksize=1048576
Variable:	Mode
Werte:	{0, 1, 2, 3} Komprimierungsmodus: Mode=0 ohne Kompression Mode=1 => ZLIB (geringe Komprimierung, sehr schnell) Mode=2 => BZIP (mittlere Komprimierung, schnell) Mode=3 => PPM (hohe Komprimierung, langsam)
Variable:	Level
Werte:	{0, 1, 2, 3, 4, 5, 6, 7, 8, 9} Grad der Komprimierung (9 bedeutet maximale Komprimierung).
Variable:	Blocksize
Werte:	<Byte> Pufferspeicher in Myte, der für die Komprimierung bereitgestellt wird (default = 1 MB).

Objekt:	AuthProvider.ini - die Konfigurationsdatei des AuthProviders
Ort:	<dllDir>\AuthProvider\AuthProvider.ini Die Konfigurationsdatei "AuthProvider.ini" befindet sich im AuthProvider-Verzeichnis und enthält die benötigten Parameter für die Authentifizierung der Anwender.
Sektion:	Initialisierung des AuthProviders
Name:	[Init] In dieser Sektion wird das Anmeldeverfahren festgelegt. Zusätzlich kann bestimmt werden ob ein Logging der Benutzerzugriffe eingeschaltet werden soll.
Beispiel:	[Init] DefaultProvider=ADO TableLog=false
Variable:	DefaultProvider
Werte:	{Windows ADO} Bei einer Windows-Anmeldung, "DefaultProvider=Windows", erfolgt die Anmeldung beim lokalem Windows unter dem der cyProvider läuft oder in einer Domäne, falls in einer weiteren Sektion "Windows" eine gültige Domäne eingetragen wird. Bei der ADO-Anmeldung "DefaultLogonProvider=ADO" stehen die Parameter in der ADO-Sektion. Die Anmeldung kann an einen anderen cyProvider delegiert werden, vorausgesetzt, es gibt eine Vertrauensbeziehung mit dem lokalen cyProvider.
Variable:	TableLog
Werte:	{False True} Anmelde- und Abmelde-Vorgänge der Benutzer können mittels ADO in eine DB-Tabelle gespeichert werden, wenn die Variable "TableLog=True" gesetzt wird.
Variable:	Pin
Werte:	[-]<n> Soll bei der ADO-Anmeldung anstelle eines Passwortes ein Pincode genutzt werden, ist das mit Hilfe der Variablen "PIN" möglich. "<n>" legt dabei die Länge des Pincodes fest. Eine negative Länge bedeutet, dass führende Nullen entfernt werden.
Sektion:	Salt-Werte für Passwort-Hashing laden
Name:	[Ado.HashSelect] In dieser Sektion steht die SQL-Anweisung, mit der die Salt-Werte für das Passwort-Hashing (SHA1(salt1+<password>+salt2)) geladen werden. Fehlt die Sektion, dann erfolgt einfaches Hashing SHA1(<password>). Die Aliasnamen salt1 bzw. salt2 sind Vorschrift. Fehlen die beiden Sektionen "Ado.HashSelect" und "Ado.HashUpdate" wird das Passwort unverschlüsselt gespeichert.
Beispiel:	[Ado.HashSelect] SELECT usr.usr_salt1 as salt1, usr.usr_salt2 as salt2 from usr where usr_userid = :user
Variable:	
Werte:	
Sektion:	Speichern des Hashpasswortes
Name:	[Ado.HashUpdate] In dieser Sektion steht die SQL-Anweisung zum Speichern des Hashpasswortes. In dem Parameter :password steht der Hashwert. Zur Auflösung der Parameter kann der AuthProvider die Felder der Select Anweisung in der Sektion ADO.SQL und die Standardparameter (:user, :password, :role, :app) verwenden. Fehlen die beiden Sektionen "Ado.HashSelect" und "Ado.HashUpdate" wird das Passwort unverschlüsselt gespeichert.
Beispiel:	[Ado.HashUpdate] UPDATE usr set usr_password=:password where usr_userid = :user
Variable:	
Werte:	

Sektion: Logging Einstellungen	
Name:	[Logging] Wurde in der Init-Sektion "TableLog=True" gesetzt, müssen in dieser Sektion die entsprechenden Einstellungen für den LogProvider gemacht werden.
Beispiel:	[Logging] LogProviderUrl=https://localhost:8443 LogProvider=ADO.DemoDB LogTable=LOG
Variable: LogProviderUrl	
Werte:	<url> Die Variable enthält die URL des cyProviders, der die Log-Tabelle schreibt. Handelt es sich hierbei nicht um den lokalen cyProvider, dann muss eine Vertrauensbeziehung zwischen den beiden cyProvidern bestehen.
Variable: LogProvider	
Werte:	ADO.<Providername> Angabe des OLE-Providers mit Präfix ADO, der die LogTable schreibt.
Variable: LogTable	
Werte:	LOG "LogTable" enthält den Namen der Log-Tabelle (LOG). Die Log-Tabelle hat einen fest definierten Aufbau. In der Log Tabelle werden erfolgreiche/erfolgreiche An-/Abmeldeversuche gespeichert. CREATE TABLE LOG ([LOG_IDX] [decimal](10, 0) NOT NULL, [LOG_ID] [nvarchar](20) NULL, [LOG_DATABASE] [nvarchar](50) NULL, [LOG_APP] [nvarchar](50) NULL, [LOG_USER] [nvarchar](50) NULL, [LOG_ERROR] [decimal](1, 0) NULL, [LOG_DATE] [nvarchar](14)) Eine zweite Tabelle LGT enthält sprachabhängige Texte zur Erläuterung. CREATE TABLE LGT ([LGT_IDX] [decimal](10, 0) NOT NULL, [LGT_ID] [nvarchar](20) NULL, [LGT_IDTEXT_0] [nvarchar](50) NULL, [LGT_IDINFO_0] [nvarchar](200) NULL) LGT_ID LGT_IDTEXT_0 LGT_IDINFO_0 LOGONSUCCESS Anmeldung erfolgreich NULL LOGONERROR Anmeldefehler Name und/oder Passwort wurden falsch eingegeben LOGOFFEXPIRED Abmeldefehler Der Benutzer war zu lange inaktiv und wurde vom System abgemeldet LOGONSUCCEDED Anmeldefehler Die max. Anzahl der Benutzerlizenzen ist überschritten LOGOFFSUCCESS Abmeldung erfolgreich NULL LOGOFFERROR Anmeldefehler Benutzer konnte wegen eines technischen Problems nicht abgemeldet werden
Sektion: Windows Domänenanmeldung	
Name:	[windows] Bei einer Windows-Anmeldung ("DefaultProvider=Windows" in der Init-Sektion) erfolgt die Anmeldung beim lokalen Windows-System unter dem der cyProvider läuft oder in einer Domäne. Damit eine Domänenanmeldung erfolgt, muss in der Windows-Sektion die Variable "domain" eine gültige Domäne enthalten.
Beispiel:	[windows] domain=mydomain
Variable:	
Werte:	

Sektion:	Windows externe Domänenanmeldung
Name:	[LDAP] Bei einer Windows-Anmeldung ("DefaultProvider=Windows" in der Init-Sektion), bei der die Authentifizierung nicht am lokalen Windows sondern an einem externen Domänencontroller erfolgt, wird eine ADO-Verbindung zur Ermittlung der Berechtigungsgruppe (CYSYSTEMnn nn=Rolle) der Users benötigt.
Beispiel:	[LDAP] LogonUrl=https://localhost:8443 LDAPProvider=ADO.LDAP
Variable:	LogonUrl_LDAP
Werte:	<url> Die Variable "LogonUrl" definiert die URL des cyProviders über den der ADO-Zugriff auf das active directory erfolgen soll.
Variable:	LDAPProvider
Werte:	ADO.<Providername> In der Variablen "LDAPProvider" wird der OLE-Provider zugeordnet um per ADO auf das active directoty zuzugreifen..
Sektion:	LDAP Gruppe
Name:	[LDAPGROUP] Bei einer Anmeldung an einem externen Domänencontroller wird mit dem hier definierten SQL-Statement die Gruppenzugehörigkeit und damit die Berechtigungsgruppe (CYSYSTEMnn nn=Rolle) der Users ermittelt.
Beispiel:	[LDAPGROUP.SQL] SELECT name FROM 'LDAP://dc=myCompany,dc=com' WHERE objectClass='group' and member = :distinguishedname and name = 'CYSYSTEM*' order by name desc
Variable:	
Werte:	
Sektion:	LDAP distinguishedname
Name:	[LDAPUSER] Bei einer Anmeldung an einem externen Domänencontroller wird zur Ermittlung der Berechtigungsgruppe (CYSYSTEMnn nn=Rolle) der Users zunächst der distinguishedname des Users ermittelt. Die eigentliche Gruppenzugehörigkeit wird dann mit Hilfe des SQL-Statemants aus der [LDAPGROUP.SQL]-Sektion ermittelt.
Beispiel:	[LDAPUSER.SQL] SELECT distinguishedname FROM 'LDAP://dc=myCompany,dc=com' WHERE objectClass='user' and samAccountname=:user
Variable:	
Werte:	
Sektion:	Einstellungen für den Datenbank Logon
Name:	[ADO] Wurde in der Init-Sektion "DefaultProvider=ADO" gesetzt, müssen in dieser Sektion die entsprechenden Einstellungen für den LogonProvider gemacht werden.
Beispiel:	[ADO] LogonUrl=https://localhost:8443 LogonProvider=ADO.DemoDB
Variable:	LogonUrl
Werte:	<url> Die Variable "LogonUrl" definiert die URL des cyProviders, bei dem die Anmeldung erfolgen soll.
Variable:	LogonProvider
Werte:	ADO.<Providername> In der Variablen "LogonProvider" wird der OLE-Provider zugeordnet.

Sektion:	SQL-Abfrage für den Datenbank Logon
Name:	[ADO.SQL] In dieser Sektion wird die SQL-Abfrage für den Datenbank Logon hinterlegt. Es müssen die Werte "USR_USERID", "USR_PASSWORD" und "ROLE" in der Datenbank zu denen vom Anwender eingegebenen Daten (":USER", ":PASSWORD") ermittelt werden. Dabei kann das Passwort auch ein Pincode (siehe Variable "Pin") sein.
Beispiel:	[ADO.SQL] SELECT usr.usr_logon as USR_USERID, usr.usr_password, rol.rol_bezeichnung as usr_role FROM USR join rol on usr.rol_id = rol.rol_id WHERE USR.USR_LOGON = :USER and USR.USR_PASSWORD = :PASSWORD
Variable:	
Werte:	
Sektion:	AuthProvider Kompression
Name:	[Compression] Sektion in der die Komprimierungsparameter für den AuthProvider gesetzt werden können.
Beispiel:	[Compression] Mode=1 Level=4 Blocksize=1048576
Variable:	Mode
Werte:	{0, 1, 2, 3} Komprimierungsmodus: Mode=0 ohne Kompression Mode=1 => ZLIB (geringe Komprimierung, sehr schnell) Mode=2 => BZIP (mittlere Komprimierung, schnell) Mode=3 => PPM (hohe Komprimierung, langsam)
Variable:	Level
Werte:	{0, 1, 2, 3, 4, 5, 6, 7, 8, 9} Grad der Komprimierung (9 bedeutet maximale Komprimierung).
Variable:	Blocksize
Werte:	<Byte> Pufferspeicher in Myte, der für die Komprimierung bereitgestellt wird (default = 1 MB).
Objekt:	FileProvider.ini - die Konfigurationsdatei des FileProviders
Ort:	<dllldir>\FileProvider\FileProvider.ini Die Konfigurationsdatei "FileProvider.ini" befindet sich im FileProvider-Verzeichnis und enthält die benötigten Parameter für die Zugriffsrechte der Anwenderrollen und die Definitionen der Filesystem-Tabellen.
Sektion:	Initialisierung des FileProviders
Name:	[Init] Diese Sektion enthält die Variable "Environment" (cyProvider-Betriebsart für Testumgebungen) sowie die Aufzählung von Filesystem-Tabellen.
Beispiel:	[Init] Environment=Test NtfsTables=OBJ SEC SPZ PAR
Variable:	Environment
Werte:	{Test} Mit der Variablen "Environment=Test" wird der Fileprovider gezwungen bei jedem Aufruf einer Anwendung auf Kosten der Performance die Verzeichnisstruktur neu einzulesen. Damit wird ein Neustart des cyProviders in Entwicklungsumgebungen mit häufigen Änderungen überflüssig. Ausnahme: das GLOBALS-Verzeichnis wird nicht neu eingelesen. Änderungen an der Verzeichnisstruktur im GLOBALS-Verzeichnis werden erst nach einem Neustart des cyProviders erkannt.
Variable:	NtfsTables
Werte:	{<tablename_1> ... <tablename_n>} Aufzählung aller Filesystem-Tabellen. Die Filesystem-Tabelle wird in einer eigenen Sektion näher spezifiziert.

Sektion:	Tabelle des Filesystems
Name:	[<tablename>] In dieser Sektion wird für eine Tabelle des Filesystems die Datei mit dem Create-Statement und das Verzeichnis, in dem die Tabelle gespeichert werden soll, benannt.
Beispiel:	[OBJ] SqlFile=CreateOBJ.Sql RootPath=c:\cyProvider\Data\cySyntax\Object
Variable:	rootpath
Werte:	<pathname> Die Variable "rootpath" benennt das Verzeichnis, in dem sich die Source-Files der Anwendungen oder die Filesystem-Tabellen befinden.
Variable:	SqlFile
Werte:	{<filename>} Name der Datei, die das Create-Statement für die Filesystem-Tabelle enthält. Diese Datei muss im FileProvider-Verzeichnis liegen.
Sektion:	Filerechte für Lesezugriff
Name:	[Select] In dieser Sektion wird festgelegt, welche Rollenberechtigung für das Lesen von Verzeichnissen, Dateien oder Filesystem-Tabellen erforderlich ist. Um ein Verzeichnis, eine Datei oder eine ganze Anwendung zu laden, muss die Rolle des Benutzers gleich oder größer sein als die der hier zugewiesenen Rolle. Es können Rollen von 00 - 99 zugewiesen werden. Die Rolle 00 bedeutet, dass keine Berechtigung für das Objekt erforderlich ist. Damit ein Benutzer mit einer hohen Rolle nicht automatisch auf alle Objekte zugreifen kann, gibt es die Möglichkeit des Anwendungsrollen-Mappings. Das Anwendungsrollen-Mapping erfolgt in einer separaten Sektion pro Anwendung.
Beispiel:	[Select] browser=00 myapp\appglobals=00 myapp\roles\guest=00 myapp\roles\admin=02 Object=00 Section=00
Variable:	
Werte:	
Sektion:	Filerechte für das Einfügen von Daten
Name:	[Insert] In dieser Sektion wird festgelegt welche Benutzerrollen in welchen Filesystem-Tabellen Daten einfügen dürfen.
Beispiel:	[Insert] Object=02 Section=02
Variable:	
Werte:	
Sektion:	Filerechte für das Ändern von Daten
Name:	[Update] In dieser Sektion wird festgelegt welche Benutzerrollen in welchen Filesystem-Tabellen Daten ändern dürfen.
Beispiel:	[Update] Object=02 Section=02
Variable:	
Werte:	

Sektion:	Filerechte für das Löschen von Daten
Name:	[Delete] In dieser Sektion wird festgelegt welche Benutzerrollen in welchen Filesystem-Tabellen Daten löschen dürfen.
Beispiel:	[Delete] Object=02 Section=02
Variable:	
Werte:	
Sektion:	Mapping von Anwendungsrollen
Name:	[<applicationname>] Das Mapping von Anwendungsrollen erfolgt in der Form Anwendungsrolle=Benutzerrolle. Vor jedem Dateizugriff durchsucht der Fileprovider die Mapping-Tabelle nach der größten Benutzerrolle die kleiner oder gleich der Rolle des Anwenders ist und ordnet die hier definierte Anwendungsrolle zu. Ist im Beispiel unten die Rolle des Anwenders für die Applikation "cyDoku" größer gleich "90", so wird ihm innerhalb der Anwendung die Anwendungsrolle "02" zugeordnet. Das heißt, er kann alle Verzeichnisse/Dateien lesen, die die Rollenberechtigung "02" voraussetzen.
Beispiel:	[cyDoku] 00=89 02=90
Variable:	
Werte:	
Sektion:	FileProvider Kompression
Name:	[Compression] Sektion in der die Komprimierungsparameter für den AuthProvider gesetzt werden können.
Beispiel:	[Compression] Mode=1 Level=4 Blocksize=1048576
Variable:	Mode
Werte:	{0, 1, 2, 3} Komprimierungsmodus: Mode=0 ohne Kompression Mode=1 => ZLIB (geringe Komprimierung, sehr schnell) Mode=2 => BZIP (mittlere Komprimierung, schnell) Mode=3 => PPM (hohe Komprimierung, langsam)
Variable:	Level
Werte:	{0, 1, 2, 3, 4, 5, 6, 7, 8, 9} Grad der Komprimierung (9 bedeutet maximale Komprimierung).
Variable:	Blocksize
Werte:	<Byte> Pufferspeicher in Myte, der für die Komprimierung bereitgestellt wird (default = 1 MB).
Objekt:	MailProvider.ini - die Konfigurationsdatei des MailProviders
Ort:	<dllldir>\MailProvider\MailProvider.ini Die Konfigurationsdatei "MailProvider.ini" befindet sich im MailProvider-Verzeichnis und enthält die benötigten Parameter für den Mailverkehr.

Sektion:	Definitionen zum Mailversand
Name:	[Mail] Sektion mit Einstellungen zum Mailversand.
Beispiel:	[Mail] Mailer=mySMTPServer Background=True BackTimer=10000
Variable:	Mailer(list)
Werte:	{<smtphost_1> ... <smtphost_n>} Aufzählung aller SMTP-Server über die Mails versendet werden können. Die einzelnen Mailer werden in einer separaten Sektion definiert.
Variable:	Background
Werte:	{False True} Mit "Background=True" (Default=False) wird die Mail im Hintergrund versandt. Eine Hintergrundtask prüft in Zeitintervallen ("Backtimer"), ob Mails zum Versand anstehen und sendet diese zum SMTP-Server. ANMERKUNG: der Versand im Hintergrund verbessert das Antwortzeitverhalten um den Faktor 6.
Variable:	BackTimer
Werte:	{ms} "Backtimer" legt das Zeitintervall in Millisekunden (Default=60000) fest, in dem die Hintergrundtask überprüfen soll, ob neue Mails zum Versand vorliegen.
Sektion:	Einstellungen für eine SMTP-Server-Verbindung
Name:	[smtphostname] Sektion mit Einstellungen für eine SMTP-Server-Verbindung. Zur Zeit wird nur "Secure=None" unterstützt.
Beispiel:	[mySmtphost] MAILFROM="myCompany" info@mydomain POP3USER=xyz POP3PASSWORD=#mypassword POP3HOST=pop3.mydomain POP3PORT=110 SMTPUSER=xyz SMTPPASSWORD=#mypassword SMTPHOST=smtp.mydomain SMTPPORT=25 Secure=None
Variable:	
Werte:	
Objekt:	MsaProvider.ini - die Konfigurationsdatei des MsaProviders
Ort:	<dllDir>\MsaProvider\MsaProvider.ini Die Konfigurationsdatei "MsaProvider.ini" befindet sich im MsaProvider-Verzeichnis und enthält die benötigten Parameter für den MSA-Datenbankzugriff. Mit dem MsaProvider (Multiple Source Access) besteht die Möglichkeit, beliebige Tabellen von anderen cyProvidern in eine speicherinterne Datenbank zu laden, auf die der cyAgent per SQL-Abfragen zugreifen kann. Dadurch ist es möglich, Daten aus unterschiedlichen Datenbanken (und Datenbanksystemen bzw. Dateistrukturen) von verschiedenen cyProvidern in einem SAM oder PAM darzustellen. Der MsaProvider unterstützt nur lesenden Zugriff. Liegen die Tabellen in Datenbanken anderer cyProvider, so muss eine Vertrauensbeziehung zwischen diesen cyProvidern bestehen.

Sektion:	MsaProvider Kompression
Name:	[Compression] Sektion in der die Komprimierungsparameter für den MsaProvider gesetzt werden können.
Beispiel:	[Compression] Mode=1 Level=4 Blocksize=1048576
Variable:	Mode
Werte:	{0, 1, 2, 3} Komprimierungsmodus: Mode=0 ohne Kompression Mode=1 => ZLIB (geringe Komprimierung, sehr schnell) Mode=2 => BZIP (mittlere Komprimierung, schnell) Mode=3 => PPM (hohe Komprimierung, langsam)
Variable:	Level
Werte:	{0, 1, 2, 3, 4, 5, 6, 7, 8, 9} Grad der Komprimierung (9 bedeutet maximale Komprimierung).
Variable:	Blocksize
Werte:	<Byte> Pufferspeicher in Myte, der für die Komprimierung bereitgestellt wird (default = 1 MB).
Objekt:	Anwendungs- und rollenspezifische MSA-Definitionen
Ort:	<dllid>\MsaProvider\MsaDefs\<Application>\<RoleName>.cdd Die Datei "<RoleName>.cdd" enthält die Informationen zum Aufbau der speicherinternen Datenbank der jeweiligen Anwendungsrolle.
Sektion:	Definition einer MSA-Datenbank
Name:	[<MsaDbName>] Aufzählung der Tabellen, die für den Aufbau einer MSA-Datenbank benötigt werden. Mit Hilfe der Variablen "Provider=MSA.<MsaDbName>" wird diese MSA-Datenbank einer View oder einem PAM zur Verfügung gestellt.
Beispiel:	[MSADEMO] Tables=MSADEMO_PER MSADEMO_SAL MSADEMO_JOB
Variable:	Tables
Werte:	{<tablename_1> ... <tablename_n>} Aufzählung der Tabellen, die zur MSA-Datenbank gehören.
Sektion:	SQL-Abfrage zum Aufbau einer MSA-Tabelle
Name:	[<tablename>.SQL] Untersektion zur MSA-Tabelle, die das SQL-Statement zur Datenbeschaffung enthält.
Beispiel:	[MSADEMO_PER.SQL] select * from PER
Variable:	
Werte:	

Sektion:	Definition einer MSA-Tabelle
Name:	[<tablename>] Diese Sektion beschreibt den Namen (Alias) der MSA-Tabelle und über welchen cyProvider ("URL" und "Provider") auf die physikalische Tabelle zugegriffen werden soll.
Beispiel:	[MSADEMO_PER] TableName=PER URL=https://myHost.myDomain:443/cyService Provider=ADO.myDB
Variable:	Provider
Werte:	{ADO.<providername> MSA.<msaDbName> FILE.DIR LOCAL Mail} Angabe des Providers, der die Daten bereitstellen soll. Aktuell stehen fünf unterschiedliche Typen (ADO, MSA, FILE, LOCAL, Mail) zur Verfügung. Während ADO für den klassischen Datenbankzugriff benutzt wird, können mit Hilfe von MSA (Multiple Source Access) Daten aus unterschiedlichen Datenbanken in eine speicherinterne Datenbank geladen werden. Der FileProvider stellt Daten des Filesystems systemkonform per SQL zur Verfügung. Der LocalProvider greift wiederum auf eine speicherinterne Datenbank (MemoryTable) zu, die zuvor vom cyAgent erstellt wurde. Der LocalProvider unterstützt zur Zeit nur PAMs. Der MSAProvider und der LocalProvider unterstützen nur den lesenden Zugriff. Der MailProvider unterstützt den Mailversand, falls ein PAM mit "MailSupport=true" definiert wird.
Variable:	TableName
Werte:	<tablename> Zuordnung eines Namens für die MSA-Tabelle.
Variable:	URL
Werte:	<url>/cyService Adresse des cyProviders der die Tabelle verwaltet.
Objekt:	globale Definitionen
Ort:	rootpath\GLOBALS Das Globals-Verzeichnis enthält, neben den Unterverzeichnissen ("IMAGES", "LANGUAGE" und "MENUS") mit den entsprechenden Definitionen, die für alle Anwendungen des cyProviders gelten, die Datei "serverdata.cdd".
Sektion:	
Name:	
Beispiel:	
Variable:	
Werte:	
Variable:	
Werte:	
Objekt:	Serverdata
Ort:	rootpath\GLOBALS\serverdata.cdd Die "serverdata.cdd" enthält übergeordnete Definitionen, die die Arbeitsweise des cyProvider bestimmen.

Sektion:	globale Servereinstellungen
Name:	[init] In dieser Sektion werden globale Servereinstellungen wie der BackGroundLevel, der LargeMemoLevel und der Compatibility Mode festgelegt.
Beispiel:	[Init] Backgroundlevel=1000 Largememolevel=500 Compatible=4
Variable:	BackGroundLevel
Werte:	<KB> "BackGroundLevel" (Angabe in KB) definiert wie groß ein Feld mindestens sein muss, damit es im Hintergrund übertragen wird. Felder die gleich oder größer dem Backgroundlevel sind, werden im Hintergrund übertragen. Der Defaultwert ist 0. Bei einem Wert kleiner oder gleich 0 findet keine Überwachung der Feldlängen statt.
Variable:	LargeMemoLevel
Werte:	<KB> Ab Version 5 definiert LargeMemoLevel (Angabe in KB) wie groß ein Memo/Blob Feld mindestens sein muss, damit anstelle seines Inhaltes ein Folder Icon angezeigt wird. Der Anwender kann durch Doppelklick auf das Icon die Daten abrufen. Der Defaultwert ist 0. Bei einem Wert kleiner oder gleich 0 findet keine Überwachung der Feldlängen statt.
Variable:	Compatible
Werte:	<Version> Der Compatibility Mode definiert die Kompatibilität zur cyProvider-Version. Der Defaultwert ist 4.
Objekt:	cyAgentVer
Ort:	rootpath\GLOBALS\cyAgentVer.cdd Die "cyAgentVer.cdd" enthält Informationen zur aktuellen cyAgent-Version.
Sektion:	cyAgent Versionsinformationen
Name:	[versioninfo] Diese Sektion enthält neben der aktuellen Version noch die Information in welchem Verzeichnis sich die Datei cyAgent.exe befindet. Wendet sich ein cyAgent mit einer älteren Version an den cyProvider, wird ihm diese cyAgent.exe zum download angeboten. Diese Funktion setzt als Client Betriebssystem mindestens Microsoft Vista voraus.
Beispiel:	[versioninfo] version=5.2.0.122 agentfile=c:\cyProvider\Upgrade\cyagent.exe
Variable:	
Werte:	
Sektion:	Restart cyAgent nach download
Name:	[restart] Diese Sektion wird benötigt um einen Batch-Job zu erstellen, der nach dem download die cyAgent.exe in das ursprüngliche Verzeichnis kopiert und den cyAgent wieder startet.
Beispiel:	[restart] Timeout /T 1 > nul copy /Y /B "%snewagentfile%" /B "%sagentfile%" cd "%sagentdir%" start cyAgent "%sstarterini%" quit
Variable:	
Werte:	
Objekt:	globale Images
Ort:	rootpath\GLOBALS\IMAGES Verzeichnis mit den Images, die für alle Anwendungen des cyProviders gelten.

Sektion:	
Name:	
Beispiel:	
Variable:	
Werte:	
Variable:	
Werte:	
Objekt:	globale Sprachdateien
Ort:	rootpath\GLOBALS\LANGUAGE\APPLICATION Verzeichnis mit den globalen Sprachdateien (" <code><languageid>.cdd</code> ") des cyProviders, in denen Buttontexte, Dialoge, etc. in der jeweiligen Sprache abgelegt sind. Die Languageid entspricht den Sprachkürzeln nach ISO 639-2 .
Sektion:	
Name:	
Beispiel:	
Variable:	
Werte:	
Variable:	
Werte:	
Objekt:	globale Sprachdateien externer Komponenten
Ort:	rootpath\GLOBALS\LANGUAGE\EXTERNAL Verzeichnis mit den globalen Sprachdateien (" <code><languageid>.cdd</code> ") des cyProviders, in denen Buttontexte, Dialoge, etc. externer Programmkomponenten in der jeweiligen Sprache abgelegt sind. Die Languageid entspricht den Sprachkürzeln nach ISO 639-2 .
Sektion:	
Name:	
Beispiel:	
Variable:	
Werte:	
Variable:	
Werte:	
Objekt:	LanguageID-Mapping
Ort:	rootpath\GLOBALS\LANGUAGE\INFO Das Info-Verzeichnis enthält die Datei "LanguageIDS.cdd", in der die vom Anwender gewählte Sprache einer Zahl zugeordnet wird. Mit den Einstellungen DEU=0 und ENG=1 können beispielsweise in dem definierten SQL-Statement über den Parameter <code>%LanguageID%</code> sprachspezifische Inhalte selektiert werden. Der cyProvider ersetzt in der SQL-Abfrage dann das Attribut <code>remark_%LanguageID%</code> durch <code>remark_0</code> , falls der Anwender die deutsche Sprache eingestellt hat. Hat man das Datenmodell entsprechend aufgebaut (z.B. durch Anlegen der Attribute <code>remark_0</code> und <code>remark_1</code> in derselben Tabelle), kann online zwischen den Sprachen hin- und hergeschaltet und der sprachabhängige Feldinhalt angezeigt werden.

Sektion:	
Name:	
Beispiel:	
Variable:	
Werte:	
Variable:	
Werte:	
Objekt:	globale Sprachdateien zu den Fehlermeldungen
Ort:	rootpath\GLOBALS\LANGUAGE\MESSAGE Verzeichnis mit den globalen Sprachdateien (" <code><languageid>.cdd</code> ") des cyProviders, in denen Systemfehlermeldungen in der jeweiligen Sprache hinterlegt sind. Die Languageid entspricht den Sprachkürzeln nach ISO 639-2 .
Sektion:	
Name:	
Beispiel:	
Variable:	
Werte:	
Variable:	
Werte:	
Objekt:	globale Menüdefinitionen
Ort:	rootpath\GLOBALS\MENUS Verzeichnis mit den globalen Menüs (Bars) für alle SAMs und PAMs. Solange im Menus-Verzeichnis einer Anwendung keine anderen Definitionen vorgenommen werden, gelten diese Menüs für alle Applikationen des cyProviders. Die Menüs für SAMs werden in der Datei "TcySAM.cdd" und die Menüs für PAMs in "TcyPAM.cdd" definiert.
Sektion:	
Name:	
Beispiel:	
Variable:	
Werte:	
Variable:	
Werte:	
Objekt:	globale Skindefinitionen
Ort:	rootpath\GLOBALS\SKINS Das Verzeichnis enthält die Skins die der cyProvider für den cyAgent zur Verfügung stellt.

Sektion:	
Name:	
Beispiel:	
Variable:	
Werte:	
Variable:	
Werte:	
Objekt: Design-Definitionen der Anwendung	
Ort:	rootpath\<<ApplicationName>\APPGLOBALS\Design.cdd Die Datei "Design.cdd" enthält die Style- und Farbdefinitionen sowie StyleConditions der Anwendung. Mit StyleConditions ist es möglich, in den Views den einzelnen Zellen in Abhängigkeit vom Inhalt einen anderen Style (z.B. Hintergrund- und Textfarbe) zuzuordnen.
Sektion: Critical colors	
Name:	[CriticalColors] In der Sektion "CriticalColors" werden Hintergrundfarben für nicht eingabebereite Felder, für Mussfelder oder für Felder, die einen Konflikt (Datensatz inzwischen von einem anderen Anwender geändert) ausgelöst haben, definiert. Ein 3 Byte großer Hexwert, mit dem Präfix \$, bestimmt den Farbton. Das linke Byte definiert den Anteil Rottöne, das mittlere Byte den Anteil Grüntöne und das letzte den Anteil der Blautöne. Alternativ kann die Farbe auch als umgerechneter Dezimalwert angegeben werden.
Beispiel:	[CriticalColors] ColorDisabled=\$F0F8FF ColorNotNull=\$99FFE5 ColorConflict=\$FF0000
Variable: ColorConflict	
Werte:	\$<hex> Die Variable enthält einen 3 Byte großen Hexadezimalwert (mit dem Präfix \$) oder alternativ den umgerechneten Dezimalwert für die Hintergrundfarbe von PAM-Feldern bei Modify-Konflikten. Diese werden z.B. ausgelöst, wenn der Anwender einen Datensatz ändern will, der inzwischen von einem anderen Anwender geändert wurde.
Variable: ColorDisabled	
Werte:	\$<hex> Die Variable enthält einen 3 Byte großen Hexadezimalwert (mit dem Präfix \$) oder alternativ den umgerechneten Dezimalwert für die Hintergrundfarbe eines nicht eingabebereiten PAM-Feldes. Im Gegensatz zur Variablen "DisabledFieldStyle" in der "TcyPAM-Sektion" wird hier eine Farbe unabhängig vom aktiven Stylesheet zugeordnet.
Variable: ColorNotNull	
Werte:	\$<hex> Die Variable enthält einen 3 Byte großen Hexadezimalwert (mit dem Präfix \$) oder alternativ den umgerechneten Dezimalwert für die Hintergrundfarbe eines PAM-Mussfeldes. Im Gegensatz zur Variablen "NotNullFieldStyle" in der "TcyPAM-Sektion" wird hier eine Farbe unabhängig vom aktiven Stylesheet zugeordnet.
Sektion: SAM Styles	
Name:	[TcySAM] Sektion mit weiteren Style-Zuordnungen für SAM-Objekte, die nicht in den Stylesheets der "Styles.cdd" enthalten sind.
Beispiel:	[TcySAM] InfoStyle=IncSearch
Variable: InfoStyle	
Werte:	<stylename> Die Info Fields der SAMs oder PAMs erhalten den Style "<stylename>" des aktiven Stylesheets. Diese Zuordnung kann direkt im Form beim SAM oder PAM erfolgen oder global für alle SAMs und PAMs in der "Design.cdd".

Sektion:	PAM Styles
Name:	[TcyPAM] Sektion mit weiteren Style-Zuordnungen für PAM-Objekte, die nicht in den Stylesheets der "Styles.cdd" enthalten sind.
Beispiel:	[TcyPAM] InfoStyle=IncSearch FocusedStyle=Selection LabelStyle=Header NormalStyle=Content DisabledStyle=Header DisabledFieldStyle=Background NotNullFieldStyle=GroupByBox
Variable:	InfoStyle
Werte:	<stylename> Die Info Fields der SAMs oder PAMs erhalten den Style "<stylename>" des aktiven Stylesheets. Diese Zuordnung kann direkt im Form beim SAM oder PAM erfolgen oder global für alle SAMs und PAMs in der "Design.cdd".
Variable:	FocusedStyle
Werte:	<stylename> Das angeklickte PAM-Feld erhält den Style "<stylename>" des aktiven Stylesheets.
Variable:	LabelStyle
Werte:	<stylename> Die Label des PAMs erhalten den Style "<stylename>" des aktiven Stylesheets.
Variable:	NormalStyle
Werte:	<stylename> Die nicht angeklickten und eingabefähigen Felder des PAMs erhalten den Style "<stylename>" des aktiven Stylesheets.
Variable:	DisabledStyle
Werte:	<stylename> Die nicht eingabefähigen Felder des PAMs erhalten den Style "<stylename>" des aktiven Stylesheets.
Variable:	DisabledFieldStyle
Werte:	<stylename> Ist die Variable "ColorDisabled" in der CriticalColors-Sektion nicht definiert, erhalten nicht eingabefähige Felder des PAMs den Style "<stylename>" des aktiven Stylesheets.
Variable:	NotNullFieldStyle
Werte:	<stylename> Ist die Variable "ColorNotNull" in der CriticalColors-Sektion nicht definiert, erhalten Mussfelder des PAMs den Style "<stylename>" des aktiven Stylesheets.
Sektion:	Styles
Name:	[Styles] In dieser Sektion werden alle Styles aufgeführt, die zusätzlich zu den Styles der Stylesheets benötigt werden. Die Stylesdefinition erfolgt in einer separaten Sektion unter dem Stylenamen.
Beispiel:	[Styles] greenblack yellowblack redblack bluewhite pinkblack
Variable:	
Werte:	

Sektion:	Style
Name:	[<stylename>] Sektion zur Definition eines Styles.
Beispiel:	[yellowblack] color=\$00FFFF font.name=Arial font.color=\$000000
Variable:	color
Werte:	\$<hex> Die Variable enthält einen 3 Byte großen Hexadezimalwert (mit dem Präfix \$) oder alternativ den umgerechneten Dezimalwert für die Hintergrundfarbe des Styles.
Variable:	font.color
Werte:	\$<hex> Die Variable enthält einen 3 Byte großen Hexadezimalwert (mit dem Präfix \$) oder alternativ den umgerechneten Dezimalwert für die Schriftfarbe des Styles.
Variable:	font.name
Werte:	<fontname> Name der Schrift.
Variable:	font.style
Werte:	[fsBold] [fsItalic] [fsUnderline] [fsStrikeOut] Zuordnung des gewünschten Schriftstils: fsBold (fett), fsItalic (kursiv), fsUnderline (unterstrichen), fsStrikeOut (durchgestrichen).
Variable:	font.size
Werte:	n "font.size" bestimmt die Größe der Schrift in Punkt.
Sektion:	Style Condition
Name:	[<stylecondition> Felder einer View können einer "Stylecondition" zugeordnet werden, um sie in Abhängigkeit vom Feldinhalt in einem anderen Style darzustellen. In dieser Sektion werden die verschiedenen Styles der jeweiligen Bedingung zugeordnet.
Beispiel:	[order] greenblack=(ord_amount >= 100000) yellowblack=(ord_amount < 100000 and ord_amount >= 50000)
Variable:	style
Werte:	(condition) Während der "<style>" selber in separaten Styles-Sektion definiert sein muss, ist die Bedingung ein Boolescher-Ausdruck aus Konstanten und Feldern der View.
Objekt:	StyleSheets der Anwendung
Ort:	rootpath\<ApplicationName>\APPGLOBALS\Style.cdd Die Datei "Style.cdd" enthält die StyleSheets, die dem Anwender zur Auswahl zur Verfügung stehen. Jedem SAM wird über die Variable ActiveStyleSheetName eines dieser StyleSheets zugeordnet. Die PAMs übernehmen die StyleSheets ihrer MasterView. Diese Zuordnung kann der Anwender mit Hilfe des Style-Designers ändern. Er hat sogar die Möglichkeit, mit dem Style-Designer eigene StyleSheets zu entwerfen, die in der lokale Datei "<ApplicationName>_StyleFile.cdd" abgelegt werden. Administratoren haben die Möglichkeit Stylesheets aus der lokalen StyleFile in die "Style.cdd" zu übernehmen und damit die Auswahl der Styles zu erweitern.

Sektion:	
Name:	
Beispiel:	
Variable:	
Werte:	
Variable:	
Werte:	
Objekt: Definitionen zum Aufruf externer Programme	
Ort:	rootpath\<<ApplicationName>\APPGLOBALS\Programs.cdd Die Datei "Programs.cdd" enthält Definitionen zum Aufruf externer Programme.
Sektion:	
Name:	
Beispiel:	
Variable:	
Werte:	
Variable:	
Werte:	
Objekt: Images der Anwendung	
Ort:	rootpath\<<ApplicationName>\APPGLOBALS\IMAGES Verzeichnis mit den Images der Anwendung.
Sektion:	
Name:	
Beispiel:	
Variable:	
Werte:	
Variable:	
Werte:	
Objekt: Sprachdateien der Anwendung	
Ort:	rootpath\<<ApplicationName>\APPGLOBALS\LANGUAGE Verzeichnis mit den Sprachdateien ("<<languageid>.cdd") der Anwendung in denen Menüpunkte, Feldüberschriften, Hinweise, etc. in der jeweiligen Sprache abgelegt sind. Die Languageid entspricht den Sprachkürzeln nach ISO 639-2 .
Sektion: Anwendungsüberschrift	
Name:	[<applicationname> In dieser Sektion wird die Überschrift der Anwendung festgelegt, die oben links im Hauptfenster angezeigt wird.
Beispiel:	[myApplication] Caption=Hello World
Variable:	
Werte:	

Sektion:	Überschriften und Hinweise zu Menüpunkten
Name:	[Barmanager] Sektion mit den Überschriften und Hinweisen zu den Menüpunkten der Anwendung. Die Hinweise erscheinen in der Systeminfozeile oben rechts. Ein & vor einem Buchstaben einer Überschrift, markiert diesen Buchstaben im Menüpunkt durch einen Unterstrich als Tastaturbefehl.
Beispiel:	[Barmanger] mnuOptions1.Caption=&Sprache mnuOptions1.Hint=Sprachauswahl
Variable:	
Werte:	
Sektion:	Überschriften und Hinweise zu zusätzlichen Buttons
Name:	[Barmanager.Extension] Für zusätzlich definierte Buttons der Anwendung werden in dieser Sektion die Überschriften und Hinweise gepflegt. Die Hinweise erscheinen in der Systeminfozeile oben rechts.
Beispiel:	[Barmanager.Extension] btnShell.Caption=Dokument öffnen btnShell.Hint=Dokument mit externen Programm öffnen
Variable:	
Werte:	
Sektion:	Alias Map
Name:	[AliasMap] Die "AliasMap" bietet die Möglichkeit, dass die Überschriften und Hinweise zu einem SAM(View) oder PAM aus einer anderen Sektion übernommen werden. Dabei übernimmt der Erstgenannte die Texte vom Zweiten. Der cyAgent sucht die Texte immer zunächst in der Sektion mit dem Namen der View oder des PAMs in der Language-Datei der Anwendung. Findet er keine Sektion unter diesem Namen oder existiert in dieser Sektion kein Eintrag für ein bestimmtes Datenbankfeld, wird die Suche über die "AliasMap" fortgesetzt. Sollte auch hier nichts gefunden werden, wird nach einer gleichnamigen Sektion in der globalen Sprachdatei ("/GLOBALS/LANGUAGE/APPLICATION") gesucht. Damit besteht die Möglichkeit Texte, die für mehrere Views und PAMs gültig sind, über eine Alias-Sektion bereitzustellen und nur die davon abweichenden Texte in einer eigenen Sektion vorzuhalten.
Beispiel:	[AliasMap] cyPAM_Order=View_Order
Variable:	
Werte:	
Sektion:	Überschriften und Hinweise einer View
Name:	[<viewname>] Sektion mit den Überschriften und Hinweisen einer View.
Beispiel:	[View_Order] Caption=Aufträge Filter0=alle Aufträge ord_id.Caption=Auftrags-ID ord_id.Hint=Auftrags-Identifikationsnummer OnDBLClick.ActionHint=Doppelklick: Pflege der Auftragsdaten
Variable:	
Werte:	

Sektion:	Überschriften und Hinweise eines PAMs
Name:	[<pamname>] Sektion mit den Überschriften und Hinweisen eines PAMs.
Beispiel:	[cyPAM_Order] ord_id.Caption=Auftrags-ID ord_id.Hint=Auftrags-Identifikationsnummer ord_status.Caption=Status ord_status.CaptionChecked=aktiv ord_status.CaptionUnchecked=passiv ord_status.CaptionUndefined=nicht definiert
Variable:	
Werte:	
Sektion:	Überschriften der Form
Name:	[<formname>] Sektion mit der Überschrift des Forms sowie den Überschriften der Tabulatoren, falls im Form vorhanden.
Beispiel:	[cyPAM_Order] Caption=Pflege der Auftragsdaten Tab1.Caption=Stammdaten Tab2.Caption=Bemerkungen
Variable:	
Werte:	
Sektion:	Überschriften eines Reports und seiner Textvariablen
Name:	[<reportname>] Sektion mit der Überschrift des Reports im Menü und seiner Textvariablen.
Beispiel:	[Report_Order] Caption=Liste aller Aufträge Header.Caption=Auftragsübersicht
Variable:	
Werte:	
Objekt:	Sprachdateien mit den Fehlermeldungen der Anwendung
Ort:	rootpath\<ApplicationName>\APPGLOBALS\LANGUAGE\MESSAGE Verzeichnis mit den Sprachdateien (" <languageid> .cdd") der Anwendung in denen Fehlermeldungen von Executables in der jeweiligen Sprache hinterlegt sind. Die Languageid entspricht den Sprachkürzeln nach ISO 639-2 .
Sektion:	
Name:	
Beispiel:	
Variable:	
Werte:	
Variable:	
Werte:	

Objekt:	anwendungsspezifische Menüdefinitionen
Ort:	rootpath\<<ApplicationName>\APPGLOBALS\MENUS Verzeichnis mit den anwendungsspezifischen Menüs (Bars). Während im Verzeichnis "\GLOBALS\MENUS" die Menüs für die Applikationen definiert werden, können in diesem Verzeichnis Menüs, die nur für diese Anwendung gelten, abgelegt werden. Dabei werden Menüs für SAMs in der Datei "TcySAM.cdd" und Menüs für PAMs der Anwendung in "TcyPAM.cdd" definiert. Zusätzlich besteht noch die Möglichkeit, für einzelne SAMs oder PAMs eigene Menüs anzulegen.
Sektion:	
Name:	
Beispiel:	
Variable:	
Werte:	
Variable:	
Werte:	
Objekt:	Definitionen der Anwendungsrolle
Ort:	rootpath\<<ApplicationName>\ROLES\<<rolename>\<applicationname>.cdd Die Datei "<ApplicationName>.cdd" im Rollen-Verzeichnis der Anwendung enthält, neben einigen rollen- und anwendungsspezifischen Definitionen, im Wesentlichen die Definitionen zum Aufbau des Menüs der Anwendung.
Sektion:	Anwendungsdefinitionen
Name:	[<applicationname> Diese Sektion enthält die Liste der Anwendungsparameter und die Information welches Formular beim Start angezeigt werden soll. Zusätzlich besteht die Möglichkeit für die Anwendung den "PamLock" zu aktivieren.
Beispiel:	[cyDoku] PamLock=False Parameters=Mandant PI AutoStart=Frm_Syntax
Variable:	PamLock
Werte:	{False True} Änderungen im PAM werden verworfen, wenn das Fenster geschlossen wird. Der Anwender erhält keine Information über anstehende Änderungen. Mit "PamLock=True" wechseln PAMs in den Zustand modal, sobald eine Änderung ansteht. Alle Fenster, die zum Zeitpunkt der Änderungsanforderung bereits offen sind oder andere PAMs/SAMs im aktuellen Fenster sind deaktiviert. Auch Satzpositionierungen werden gesperrt. Versucht der Anwender das Fenster zu schließen, dann wird er über die ausstehende Änderung informiert und das Fenster bleibt offen. Erst nachdem er die Änderung durchgeführt oder abgebrochen hat, ist normales Arbeiten wieder möglich.
Variable:	Parameters
Werte:	{<parametername_1> <parametername_2> ... <parametername_n>} Aufzählung der anwendungsspezifischen Parameter, die jeweils in einer separaten Sektion definiert werden.
Variable:	AutoStart
Werte:	{<menuitem>} Per Autostart kann ein Menüpunkt bestimmt werden, der beim Start der Anwendung automatisch geöffnet werden soll.

Sektion:	Anwendungsparameter
Name:	[<parametername>] Typ und Wert von Anwendungsparametern werden in einer eigenen Sektion definiert.
Beispiel:	[Mandant] Type=String Value=CY [PI] Type=Float Value=3.14
Variable:	Type
Werte:	{string integer float} Definition des Parametertyps.
Variable:	Value
Werte:	<value> Definition des Parameterwertes.
Sektion:	Bar Manager
Name:	[BarManager] In der BarManager-Sektion wird definiert welche "ToolBar" das Menü der Anwendung präsentiert und wie das Design dieses Menüs aussehen soll.
Beispiel:	[BarManager] ToolBars=AppMenu Gradient=200 255 200 255 200 255 BarImage=barglyph.bmp MenuAnimations=maSlide
Variable:	ToolBars
Werte:	<menuname> Die Variable "ToolBars" enthält den Namen der Sektion, in der das Menü der Anwendung definiert wird.
Variable:	Gradient
Werte:	<red_up> <red_to> <green_up> <green_to> <blue_up> <blue_to> Definition des Farbverlaufes für den Bildhintergrund der ersten Spalte der Untermenüs.
Variable:	BarImage
Werte:	<filename>.bmp Die erste Spalte der Untermenüs wird mit einem Bildzeichen hinterlegt. Die Datei "<filename>.bmp" liegt im Verzeichnis "APPGLOBALS\IMAGES".
Variable:	MenuAnimations
Werte:	{maNone maRandom maUnfold maSlide maFade} Zuordnung einer Menü-Animation: - maNone keine - maRandom zufällige Auswahl (maUnfold, maSlide or maFade) - maUnfold auffaltendes Menü - maSlide gleitendes Dropdown- oder Popup-Menü - maFade einblendendes Popup-Menü

Sektion:	Toolbar
Name:	[toolbarname] In der Toolbar-Sektion wird die Bar für das Hauptmenü der Anwendung definiert.
Beispiel:	[AppMenu] OneOnRow=True WholeRow=True MultiLine=True DockControlName=Top Menus=mnuFinish mnuModules mnuOptions MnuHelp mnuWindows
Variable:	OneOnRow
Werte:	{True False} "OneOnRow=True" lässt nur eine Bar pro Zeile zu.
Variable:	WholeRow
Werte:	{True False} "WholeRow=True" lässt nur eine Bar pro Zeile zu und nutzt die gesamte Zeile für die Bar.
Variable:	MultiLine
Werte:	{True False} "MultiLine=True" stellt bei Platzproblemen der Bar weitere Zeilen zur Verfügung.
Variable:	DockControlName
Werte:	{Top} "DockControlName=Top" plaziert das Menü unterhalb der System-Statuszeile.
Variable:	Menus
Werte:	{<menuitem_1> <menuitem_2> ... <menuitem_n> } Die Menüpunkte des Haupt- oder seiner Untermenüs in der gewünschten Anordnung von links nach rechts. Jeder Menüpunkt wird in einer weiteren Sektion mit dem hier vergebenen Namen definiert.

Sektion:	Menüpunkt
Name:	[<menuitem>] Sektion zur Definition eines Menüpunktes. Dabei kann ein Menüpunkt wiederum ein Untermenü aufrufen
Beispiel:	[mnuModules] MnuType=TMnuSub Menus=mnuModules1 mnuModules2 - mnuModules3 mnuModules4 Image=mnulcon.bmp [mnuModules1] MnuType=TMnuBtn PaintStyle=psCaptionGlyph ButtonStyle=bsChecked [mnuModules1.Action] OnClick=Form.Frm_Object
Variable:	MnuType
Werte:	{TMnuSub TMnuBtn TMnuLng TMnuChilds TMnuSkin} Die Variable legt fest, ob es sich bei dem Menüpunkt um ein - Untermenü (TMnuSub), - einen Button (TMnuBtn), - ein Submenü zur Sprachauswahl (TMnuLng) - ein Submenü mit den aktiven Fenstern (TMnuChilds) - ein Submenü zur Auswahl eines Skins (TMnuSkin) handelt.
Variable:	Menus
Werte:	{<menuitem_1> <menuitem_2> ... <menuitem_n> } Die Menüpunkte des Haupt- oder seiner Untermenüs in der gewünschten Anordnung von links nach rechts. Jeder Menüpunkt wird in einer weiteren Sektion mit dem hier vergebenen Namen definiert.
Variable:	PaintStyle
Werte:	{psStandard psCaption psCaptionGlyph psCaptionInMenu} Die Variable legt den Style/Beschriftung des Menüpunktes fest: - psStandard (nur Images in der Toolbar, im Submenü Images und Beschriftung) - psCaption (nur Beschriftung) - psCaptionGlyph (Images und Beschriftung) - psCaptionInMenu (nur Images in der Toolbar, im Submenü nur Beschriftung)
Variable:	ButtonStyle
Werte:	{bsDefault bsChecked bsDropDown} Die Variable legt den Button Style fest: - bsDefault (Standard Button) - bsChecked (Toggle Button) - bsDropDown (Button mit DropDownMenü)
Variable:	Image
Werte:	<filename>.bmp Zuordnung der Datei, die das Picture enthält, welches je nach gewählten "PaintStyle" im Button angezeigt werden soll. Die Datei "<filename>.bmp" liegt im Verzeichnis "APPGLOBALS\IMAGES" oder im Verzeichnis "GLOBALS\IMAGES".
Variable:	Accelerator
Werte:	<hotkey> Zuordnung eines Tastencodes zu einem Button.

Sektion:	Aktionen eines Menüpunktes
Name:	[<menuitem>.Action] Untersektion zur Definition der Aktion, die bei einem Menüpunkte ausgelöst werden soll.
Beispiel:	[mnuFinish.Action] OnClick=System.terminate
Variable:	OnClick
Werte:	{Form.<formname> System.<functionname> Prog.<programe>} Definition der Aktion, die bei einem Anklicken des Buttons / Menüpunktes ausgeführt werden soll. Dabei kann es sich um ein Form, eine Systemfunktion oder ein Programm handeln. Mögliche Systemfunktionen sind: - terminate (Anwendung schließen) - BgTransparent (den Hintergrund auf transparent setzen) - BgColor (die Hintergrundfarbe auswählen) - Font (die Schriftart wählen) - Info (??) - ShellOpen (z.B. ShellOpen.www.cyflex.de um eine URL mit dem Standardbrowser aufzurufen)
Objekt:	Formulardefinitionen der Anwendungsrolle
Ort:	rootpath\<ApplicationName>\ROLES\<rolename> Formulardefinitionen werden in Dateien vom Typ cdd im Rollen-Verzeichnis der Anwendung gespeichert. Mehrfach verwendbare Definitionen können in Dateien ausgelagert und mit Hilfe von "cddinclude <dateiname>" eingebunden werden.
Sektion:	Definitionen zum Formular
Name:	[<formname>] Diese Sektion enthält die wesentlichen Definitionen eines Formulars. Der Name der Sektion entspricht dem Namen der cdd-Datei.
Beispiel:	[Frm_myForm] Width=1020 Height=500 container=C0 C1
Variable:	Top
Werte:	n Initiale Positionierung des Formulars: Abstand zum Hauptmenü in Pixel. Bei Verwendung von Storefiles allerdings nicht sinnvoll.
Variable:	Left
Werte:	n Initiale Positionierung des Formulars: Abstand zum linken Rand des Hauptfensters in Pixel. Bei Verwendung von Storefiles allerdings nicht sinnvoll.
Variable:	Width
Werte:	n Breite des Formulars in Pixel.
Variable:	Height
Werte:	n Höhe des Formulars in Pixel.
Variable:	SizeConstraints
Werte:	{False True} Bei "SizeConstraints=True" kann der Anwender das Formular, den SAM oder den PAM nur vergrößern, nicht verkleinern.
Variable:	Container(list of container)
Werte:	{<containername_1> <containername_2> ...<containername_n>} Liste der Container, die das Formular unterteilen.

Sektion:	Definition eines Containers
Name:	[<formname>.<containername>] Diese Sektion definiert für jeden Container, der in der Sektion "form" enthaltenen Containerliste (z.B. "Container=C0 C1 C2 C3"), den Bezugspunkt. Der Bezugspunkt eines Containers kann entweder das Formular selbst ("Home=Form" - für den ersten Container ist das die Regel) oder ein anderer Container oder ein Tab innerhalb eines anderen Containers sein. Ferner kann über diese Sektion jeder Container zusätzlich vertikal ("Split=Vertical") oder horizontal ("Split=Horizontal") in einen ersten ????????????
Beispiel:	[Frm_myForm.C1] Home=C0.First.Tab1 Split=Vertical Dimension=65
Variable:	Home
Werte:	{FORM <cname>.FIRST <cname>.LAST <cname>.FIRST.<tabname> <cname>.LAST.<tabname>} Die Variable "Home" bestimmt wo der Container platziert werden soll. Er kann im Hauptfenster ("FORM") der Anwendung liegen, in einem bereits definierten Container oder unter einem Tab eines Containers. Da Container gesplittet sein können, verweist "FIRST" bzw. "LAST" darauf, in welchen Container Teil platziert werden soll. Nicht gesplittete Container werden standardmäßig über "FIRST" adressiert.
Variable:	Split
Werte:	{Horizontal Vertical} Ein Container kann horizontal oder vertikal unterteilt (gesplittet) werden.
Variable:	Dimension
Werte:	n Die Variable definiert die Größe des 1. Teilcontainers ("First") in Prozent zum gesamten Container.
Variable:	SplitterColor
Werte:	\$_<hex> Die Farbe des Splitters kann mit Hilfe eines 3 Byte großen Hexadezimalwert geändert werden. Die drei Bytes geben die RGB-Farbintensität für Blau, Grün und Rot an. Alternativ kann die Farbe auch als umgerechneter Dezimalwert angegeben werden.
Sektion:	die Links der Form
Name:	[<formname>.Link] In der Link-Sektion werden die Sektionennamen aller im Form vorhandenen SAM und/oder PAMs festgelegt.
Beispiel:	[Frm_myForm.Link] cySAM_Order=TcySAM cyPAM_Order=TcyPAM
Variable:	
Werte:	

Sektion:	SAM
Name:	[<SAMname>] Sektion mit den Definitionen eines SAMs. Mit einem SAM werden in der Regel Daten selektiert und angezeigt. Die Ausnahme bildet die Drag&Drop-Funktionalität, mit der auch SAMs Daten manipulieren können.
Beispiel:	[cySAM_Order] Container=C1.First ActiveStyleSheetName=reppredef_Standard ViewName=View_Order ViewMaster=View_Customer InfoFields=cus_name HiddenBars=barReport barUtilities HiddenMenus=Date SyncSelDateChange=cySAM_Customer ExecBefore=myMethod.myParam ExecAfter=myMethod.myParam ReportDesigner=True
Variable:	Container
Werte:	{<cname>.FIRST <cname>.LAST <cname>.FIRST.<tablename> <cname>.LAST.<tablename>} Zuordnung eines SAMs oder PAMs zu einem Container. Container können in zwei Teile gesplittet ("FIRST", "LAST") sein und damit maximal zwei SAMs/PAMs gleichzeitig beinhalten. Zusätzlich können einem Container noch beliebig viele Tabulatoren zugeordnet werden. Hinter jedem dieser Tabulatoren können wiederum bis zu zwei SAMs/PAMs liegen.
Variable:	ActiveStyleSheetName
Werte:	REPPREDEF_<stylename> Zuordnung eines initialen Styles. Die Styles-Definitionen befinden sich in der "styles.cdd". Wird die Variable "ActiveStyleSheetName" nicht gesetzt, versucht der cyAgent den Stylesheet "DevExpress" zuzuordnen. Sollte die Sektion "[DevExpress]" nicht in der "styles.cdd" vorhanden sein, wird der erste Stylesheet aus der "styles.cdd" zugeordnet.
Variable:	ViewName
Werte:	<viewname> Zuordnung einer View, die das SAM mit Daten füllen soll.
Variable:	ReportDesigner
Werte:	{False True} "ReportDesigner=True" bietet die Möglichkeit den FastReport-Designer (kostenpflichtige Komponente) zu starten.
Variable:	ViewType
Werte:	{ViewPivot ViewChart ViewOrgChart ViewSched ViewTree} Definiert einen vom Standard abweichenden view Typen (z.B. Pivot view, Chart view, OrgChart view, Scheduler view, Tree view).
Variable:	UseFrPDFReport
Werte:	{False True} "UseFrPDFReport=True" nutzt FastReport um ein PDF mit Adobe Reader zu öffnen. Default (False) wird versucht Adobe Reader zunächst per read zu öffnen. Sollte Adobe Standard oder Adobe Professional installiert sein, wird im 2. Schritt versucht Adobe mit open aufzurufen.
Variable:	ViewMaster
Werte:	<viewname> Zuordnung einer MasterView. Alle selektierten Werte der MasterView (und deren MasterView, ..., also der gesamten MasterView-Kaskade) stehen damit im aktuellen SAM/PAM als Parameter (" :<fieldname>") zur Verfügung.
Variable:	InfoFields
Werte:	(SQL.)<fieldname_1> (SQL.)<fieldname_2> ... (SQL.)<fieldname_n> Die unter "InfoFields" definierten Felder der View werden in der obersten Zeile (über der Toolbar-Zeile, unterhalb evtl. Tabs) des SAMs, PAMs angezeigt. Angezeigt werden können Felder der MasterView oder Felder die per SQL Statement ermittelt werden. Letztere werden durch ein vorangestelltes "SQL." gekennzeichnet. Das SQL Statement selber befindet sich in einer zusätzlichen Sektion [<fieldname.SQL>].

Variable:	SizeConstraints
Werte:	{False True} Bei "SizeConstraints=True" kann der Anwender das Formular, den SAM oder den PAM nur vergrößern, nicht verkleinern.
Variable:	InfoStyle
Werte:	<stylename> Die Info Fields der SAMs oder PAMs erhalten den Style "<stylename>" des aktiven Stylesheets. Diese Zuordnung kann direkt im Form beim SAM oder PAM erfolgen oder global für alle SAMs und PAMs in der "Design.cdd".
Variable:	HiddenBars
Werte:	{barPosition barReport barUtilities barDate barFilter barRecord barMail} Mit "HiddenBars" können einzelne Bars in SAMs oder PAMs ausgeblendet werden.
Variable:	HiddenMenus
Werte:	{FR Word Excel Pdf Html Xml Next Prior First Last Style Filter Date Del Ins Upd Mit "HiddenMenus" können einzelne Menü-Buttons in SAMs oder PAMs ausgeblendet werden. Dabei steht "FR" für den FastReport-Designer und "Send" für den "Mail-versenden-Button".
Variable:	SyncSelDateChange
Werte:	<cySAM_Name> Wird im Datumseditor eines weiteren SAMs "<cySAM_Name>" das Datum geändert, wird das Datum des SAMs automatisch synchronisiert und die Daten aktualisiert.
Variable:	ExecBefore
Werte:	<MethodName.ParameterName> Aufruf einer Methode der "<application>.dll" zum Zeitpunkt vor der Ausführung des SQL-Befehls.
Variable:	ExecAfter
Werte:	<MethodName.ParameterName> Aufruf einer Methode der "<application>.dll" zum Zeitpunkt nach der Ausführung des SQL-Befehls.
Variable:	AllowAdo
Werte:	{false true} Anstelle von SQLMemTable verwendet cySystem bei "AllowAdo=true" AdoQuery für die Bereitstellung von Daten. Bei Tabellen mit sehr vielen Einträgen ist AdoQuery wesentlich performanter. Allerdings verlangt AdoQuery eine andere Syntax beim Filter. Beispiel Filter SQLMemTable: (((per_von = NULL) or (per_von <= :SSD)) and ((per_bis = NULL) or (per_bis >= :SSD))) Dieser Filter muss für AdoQuery wie folgt definiert werden: (per_von = NULL and per_bis = NULL) or (per_von <> NULL and per_bis = NULL and per_von <= :SSD) or (per_von = NULL and per_bis <> NULL and per_bis >= :SSD) or (per_von <> NULL and per_bis <> NULL and per_von <= :SSD and per_bis >= :SSD)
Variable:	AllowGroup
Werte:	{false true} Die Group Area im Grid kann mit dieser Variablen unterdrückt werden.

Sektion:	View
Name:	[<viewname>] Diese Sektion enthält die Definitionen zur View, die das SAM mit Daten füllen soll.
Beispiel:	[View_Order] Provider=ADO.myDB IndexName=ord_id DateFormat=YYYYMMDD TimeFormat=HHNNSS VisibleFields=ord_no ord_name ord_valid_from HiddenFields=ord_id Filters=Filter1 Filter0 Reports=Orderlist Bands=myBand GroupColumns=cus_name SortOrder=ord_valid_from ord_name AddParams=ord_id NoDragFields=ord_valid_from ord_valid_up_to AllowGroupDrop=True AllowCellDrop=False AllowEmptyDrop=False
Variable:	Provider
Werte:	{ADO.<providername> MSA.<msaDbName> FILE.DIR LOCAL Mail} Angabe des Providers, der die Daten bereitstellen soll. Aktuell stehen fünf unterschiedliche Typen (ADO, MSA, FILE, LOCAL, Mail) zur Verfügung. Während ADO für den klassischen Datenbankzugriff benutzt wird, können mit Hilfe von MSA (Multiple Source Access) Daten aus unterschiedlichen Datenbanken in eine speicherinterne Datenbank geladen werden. Der FileProvider stellt Daten des Filesystems systemkonform per SQL zur Verfügung. Der LocalProvider greift wiederum auf eine speicherinterne Datenbank (MemoryTable) zu, die zuvor vom cyAgent erstellt wurde. Der LocalProvider unterstützt zur Zeit nur PAMs. Der MSAProvider und der LocalProvider unterstützen nur den lesenden Zugriff. Der MailProvider unterstützt den Mailversand, falls ein PAM mit "MailSupport=true" definiert wird.
Variable:	IndexName
Werte:	<fieldname> (<fieldname_2> ...) Die Variable "Indexname" legt fest, welches Datenbankfeld den Index beinhaltet. Der Index kann sich auch aus mehreren Feldern zusammensetzen, falls es kein numerischer Index ("CounterIndex=False") ist.
Variable:	CounterIndex
Werte:	{True False} Falls der Index der ApplyTable (Tabelle, die geändert werden soll) aus einem numerischen Feld besteht, der beim Insert automatisch gesetzt werden soll, ist "CounterIndex=True" zu setzen (default). Setzt sich der Index aus mehreren Feldern zusammen bzw. ist er nicht numerisch, muss "CounterIndex=False" gesetzt werden.
Variable:	DateFormat
Werte:	{YYYYMMDD ...} YYYY = Year, MM = Month, DD = Day Die Variable "DateFormat" dient dazu das Format des Datumseditors an das Format der Datumsfelder der View oder des PAMs anzupassen. Der im SQL-Statement oder im Filter verwendbare Standard-Parameter ":SD" erhält ebenfalls dieses Datumsformat.
Variable:	TimeFormat
Werte:	{HHNNSS ...} HH = Hour, NN = Minutes, SS = Seconds "TimeFormat" bestimmt das Zeitformat des Datumseditors.
Variable:	VisibleFields
Werte:	<fieldname_1> <fieldname_2> ... <fieldname_n> Liste der Felder, die standardmäßig in der View angezeigt werden sollen. Nicht aufgeführte Felder können vom Anwender bei Bedarf über den Spaltenwahlschalter eingeblendet werden. Dies gilt allerdings nicht für die unter "HiddenFields" aufgeführten Felder.
Variable:	HiddenFields
Werte:	<fieldname_1> <fieldname_2> ... <fieldname_n> Liste der Felder die nicht in der View erscheinen sollen.

Variable:	FieldsRow
Werte:	<fieldname_1> <fieldname_2> ... <fieldname_n> Liste der Felder einer Pivot view, die in der row header area initial angezeigt werden sollen.
Variable:	FieldsData
Werte:	<fieldname_1> <fieldname_2> ... <fieldname_n> Liste der Felder einer Pivot view, die in der data header area initial angezeigt werden sollen.
Variable:	FieldsColumn
Werte:	<fieldname_1> <fieldname_2> ... <fieldname_n> Liste der Felder einer Pivot view, die in der column header area initial angezeigt werden sollen.
Variable:	RowGrandTotals
Werte:	{True False} "RowGrandTotals=False" verhindert die Ausgabe von Zeilensummen bei pivot views. (Default True).
Variable:	ColumnGrandTotals
Werte:	{True False} "ColumnGrandTotals=False" verhindert die Ausgabe von Spaltensummen bei pivot views. (Default True).
Variable:	Filters
Werte:	<filtername_1> <filtername_2> ... Liste der vordefinierten Filter einer View.
Variable:	Reports
Werte:	<reportname_1> <reportname_2> ... Liste der FastReports, die im SAM oder PAM zusätzlich zur Verfügung stehen.
Variable:	Bands
Werte:	<bandname_1> <bandname_2> ... Liste der Bänder (bands oder specialband), die im SAM bestimmte Spalten mit einem gemeinsamen Bandheader zusammenfassen und/oder Attribute eines Datensatzes untereinander platzieren oder im Falle von specialband einen zusätzlichen Header im SAM einfügen. Die Variable wird ebenso bei Reports benutzt, um ein Dataset einem Band im Report zuzuordnen. Im Report muss allerdings zuvor ein Band mit gleichem Namen angelegt worden sein.
Variable:	GroupColumns
Werte:	<fieldname_1> <fieldname_2> ... Nach den hier angegebenen Feldern wird in der View automatisch gruppiert. Bei der Verwendung von "GroupColumns" kann der Anwender nicht mehr eigenständig gruppieren.
Variable:	SortOrder
Werte:	<fieldname_1> <fieldname_2> ... Vorgabe einer Sortierung nach mehreren Feldern in Report-Datasets oder Views. Ein Minus vor dem Feldnamen veranlasst eine absteigende Sortierung. In Views ist für den Anwender eine Sortierung nach mehreren Feldern per "Shift-Klick" möglich. Klickt der Anwender ohne "Shift" auf eine Spalte, wird nur nach dieser Spalte sortiert. Per "Reset-Button" wird die vorgegebene Sortierung wiederhergestellt.
Variable:	AddParams
Werte:	<ParameterName_1> <ParameterName_2> ... <ParameterName_n> Mit "Addparams" werden einem Selektionsdataset (die selektierte Datenmenge steht abhängigen PAMs/SAMs als Parameter zur Verfügung) zusätzliche Parameter übergeben. Die Parameter in der AddParams-Liste müssen ohne ":" geschrieben werden.
Variable:	NoDragFields
Werte:	<FieldName_1> <FieldName_2> ... <FieldName_n> Die der Variablen "NoDragFields" zugeordneten Felder können ebenso wie "HiddenFields" nicht für Drag&Drop verwendet werden.
Variable:	AllowGroupDrop
Werte:	{False True} "AllowGroupDrop=True" erlaubt ein Drop auf ein gruppiertes Feld (Default=False).
Variable:	AllowCellDrop
Werte:	{True False} "AllowCellDrop=False" verhindert einen Drop auf eine Zelle (Default True).

Variable:	AllowEmptyDrop
Werte:	{True False} "AllowEmptyDrop=False" verhindert einen Drop in den Leerbereich eines SAMs (Default True).
Variable:	AllowFieldSelect
Werte:	*<präfix> oder <fieldname_1> ... "AllowFieldSelect" benennt die Felder, die für Drag&Drop-Operationen selektiert werden dürfen. Sind hier keine Felder definiert, dann sind alle erlaubt.
Variable:	ForceSelectionDateRefresh
Werte:	{True False} Mit "ForceSelectionDateRefresh=True" wird immer ein Refresh der View bei Änderung des Selektionsdatums durchgeführt.
Variable:	ForceClickValueRefresh
Werte:	{True False} Mit "ForceClickValueRefresh=True" wird immer ein Refresh der View bei Änderung des ClickValues durchgeführt.
Variable:	TreeViewParentField
Werte:	<TreeViewParentFieldName> Die Variablen "TreeViewParentField" und "TreeViewKeyField" werden zur Abbildung der Baumstruktur benötigt. Während TreeViewParentField den Namen des Datenbankfeldes beinhaltet, das auf den Master des aktuellen Datensatzes verweist, enthält TreeViewKeyField den Namen des Datenbankfeldes, das den aktuellen Datensatz bestimmt. Die Werte beider Variablen bestimmen zusammen die eindeutige Position innerhalb der Baumstruktur.
Variable:	TreeViewKeyField
Werte:	<TreeViewKeyFieldName> Die Variablen "TreeViewParentField" und "TreeViewKeyField" werden zur Abbildung der Baumstruktur benötigt. Während TreeViewParentField den Namen des Datenbankfeldes beinhaltet, das auf den Master des aktuellen Datensatzes verweist, enthält TreeViewKeyField den Namen des Datenbankfeldes, das den aktuellen Datensatz bestimmt. Die Werte beider Variablen bestimmen zusammen die eindeutige Position innerhalb der Baumstruktur.
Variable:	TreeViewImageIndexFieldName
Werte:	<TreeViewImageIndexFieldName> Die Variable "TreeViewImageIndexFieldName" bestimmt den Namen des Feldes der View SQL, welches den TreeView ImageIndex enthält. Dieser Index bestimmt wiederum welches in der Variablen "TreeViewImageListFiles" angegebene File zur Darstellung eines Knoten im Normalzustand bzw. im selektierten Zustand benutzt werden soll.
Variable:	TreeViewImageListFiles
Werte:	<TreeViewImageListFiles_1 ... TreeViewImageListFiles_n> Die Variable "TreeViewImageListFiles" enthält die Namen der Files, die die Images zur Darstellung der Knoten enthalten. Die Files selbst befinden sich entweder im GLOBALS\IMAGES - oder im <Application>\APPGLOBALS\IMAGES- Verzeichnis.
Variable:	NeedFilter
Werte:	{False True} "NeedFilter=True" erzeugt eine Warnung, falls der Anwender versucht die View ohne einen SQLFilter zu füllen. Sinnvoll bei großen Datenmengen!
Variable:	LoadStore
Werte:	{True False} "LoadStore=False" verhindert das Laden alter Storefile Inhalte nach Designänderung einer View. Dieser Parameter sollte nur temporär genutzt werden.
Variable:	OrgKeyField
Werte:	<fieldname> Die Variable "OrgKeyField" legt fest, welches Datenbankfeld bei einer OrgChart View den Index beinhaltet.

Variable: OrgParentField
Werte: <fieldname> Die Variable "OrgParentField" legt fest, welches Datenbankfeld bei einer OrgChart View den Index des Elternknoten enthält. Hat der Knoten keine Eltern ist der Index 0.
Variable: OrgTextField
Werte: <fieldname> Die Variable "OrgTextField" legt fest, welches Datenbankfeld bei einer OrgChart View den Text enthält, der im Knoten angezeigt werden soll.
Variable: OrgWidthField
Werte: <fieldname> Die Variable "OrgWidthField" legt fest, welches Datenbankfeld bei einer OrgChart View den Wert für die Knotenbreite enthält.
Variable: OrgHeightField
Werte: <fieldname> Die Variable "OrgHeightField" legt fest, welches Datenbankfeld bei einer OrgChart View den Wert für die Knotenhöhe enthält.
Variable: OrgShapeField
Werte: <fieldname> Die Variable "OrgShapeField" legt fest, welches Datenbankfeld bei einer OrgChart View den Wert für die Darstellung des Knoten enthält. 0 = Rechteck 1 = rundes Rechteck 2 = Ellipse 3 = Diamant
Variable: OrgColorField
Werte: <fieldname> Die Variable "OrgColorField" legt fest, welches Datenbankfeld bei einer OrgChart View den Farbwert (Integer) des Knoten enthält.
Variable: OrgImageAlignField
Werte: <fieldname> Die Variable "OrgImageAlignField" legt fest, welches Datenbankfeld bei einer OrgChart View den Wert für die Bildausrichtung innerhalb des Knoten enthält. 0 = keine 1 = links oben 2 = links mittig 3 = links unten 4 = rechts oben 5 = rechts mittig 6 = rechts unten 7 = oben links 8 = oben mittig 9 = oben rechts 7 = oben links 8 = oben mittig 9 = oben rechts 10 = unten links 11 = unten mittig 12 = unten rechts
Variable: OrgCHAlignField
Werte: <fieldname> Die Variable "OrgCHAlignField" legt fest, welches Datenbankfeld bei einer OrgChart View die Information über die Ausrichtung des Kindknoten enthält. 0 = links 1 = zentriert 2 = rechts

Variable:	OrgBitmapField
Werte:	<fieldname> Die Variable "OrgBitmapField" legt fest, welches Datenbankfeld bei einer OrgChart View das Image des Knoten enthält. Erlaubt sind Bmp-, Jpeg-, Gif-, Png-, Wmf- und Ico-Images. Die Images werden per Stretch an die Größe der Variablen "Imageheight" bzw. "ImageWidth" angepasst (Default ist 40 x 40).
Variable:	ImageHeight
Werte:	n Die Variable "ImageHeight" legt die Höhe eines Images (Default 40 px) im Knoten eines OrgCharts fest.
Variable:	ImageWidth
Werte:	n Die Variable "ImageWidth" legt die Breite eines Images (Default 40 px) im Knoten eines OrgCharts fest.
Variable:	OrgEditActionField
Werte:	<fieldname> Die Variable "OrgEditActionField" legt fest, welches Datenbankfeld bei einer OrgChart View den Namen der Form enthält, die aufgerufen werden soll, wenn im Popupmenu eines Knotens der Menüpunkt Bearbeiten (~chartedit) geklickt wird.
Variable:	DefaultEditAction
Werte:	Form.<formname> Wenn mit Hilfe von "OrgEditActionField" keine aufzurufende Form zugewiesen werden konnte, wird die "DefaultEditAction"-Form gestartet. Wurde auch keine "DefaultEditAction"-Form definiert, dann kann der Knoten nicht geändert werden.
Variable:	OrgClickActionField
Werte:	<fieldname> Die Variable "OrgClickActionField" legt fest, welches Datenbankfeld bei einer OrgChart View den Namen der Form enthält, die aufgerufen werden soll, wenn ein Knoten doppelt geklickt wird.
Variable:	DragDropOperation
Werte:	<methodname> :dragidx :dropidx Über die Variable "DragDropOperation" kann man eine Executable Methode in der DLL des NopProvider aufrufen. In dieser Methode sind dann die Datenbankänderungen zu programmieren, die beim Umhängen eines Knoten erforderlich sind. Ein direktes Umhängen eines Elternknoten an einen Kindknoten ist grundsätzlich nicht möglich. Als Parameter werden der Index des Drag-Knoten und der Index des Drop-Knoten übergeben.
Sektion:	Filterdefinition zur SQL-Abfrage der View
Name:	[<viewname>.SQLFILTER] Für ein SAM kann in dieser Sektion ein SQLFILTER werden. Alle in diesem SQL Statement definierten Felder stehen dem Anwender beim SAM Aufruf zur Verfügung, um die Menge der Datenbeschaffung einzuschränken. Das SQL Statement sollte so aufgebaut sein, dass keine Datensätze selektiert werden. Im SQL Statement der View wird der SQLFILTER durch Aufnahme eines Filtertags {#sqlfilter#} in der where clause aktiviert.
Beispiel:	[View_Order.SQLFILTER] select rel.rel_last_name, cit.cit_city_%languageID% from rel left join adr on rel.rel_id = adr.rel_id left join cit on adr.cit_id = cit.cit_id where rel.rel_id is null
Variable:	
Werte:	

Sektion: SQL-Abfrage der View
Name: [<code><viewname>.SQL</code>] Untersektion zur View, die das SQL-Statement zur Datenbeschaffung enthält. Die Reihenfolge der Datenbankfelder in der Select-Clause bestimmt auch deren Reihenfolge im Daten-Selektor des SAMs sowie die Reihenfolge im SAM (von links nach rechts). Für das SQL-Statement stehen alle Parameter der Masterview-Kaskade, alle System- und Anwendungsparameter sowie der Parameter "Clickvalue" zur Verfügung. Die where clause kann durch ein Filtertag <code>{#sqlfilter#}</code> zur Aufnahme eines SQLFILTERs erweitert werden.
Beispiel: [<code>View_Order.SQL</code>] <code>select * from ord</code> <code>where ord.ord_valid_up_to > :SD { and #sqlfilter#}</code>
Variable: Werte:
Sektion: SQL-Abfrage um View ohne Daten zu initialisieren
Name: [<code><viewname>.EmptyInitSQL</code>] Untersektion zur View, die das SQL-Statement enthält um eine View zunächst ohne Daten zu initialisieren. Damit wird dem User die Möglichkeit gegeben die Daten gemäß des SQL-Statements der SQL-Sektion erst mit Klick auf den Refresh-Button einzulesen. Dieses Vorgehen kann sinnvoll sein, wenn in Abhängigkeit von der Masterview mit unterschiedlichen und zum teil sehr großen Datenmengen zu rechnen ist.
Beispiel: [<code>View_Order.EmptyInitSQL</code>] <code>select * from ord</code> <code>where ord.ord_id <> ord.ord_id</code>
Variable: Werte:

Sektion:	Definitionen zum Feld einer View
Name:	[<viewname>.<fieldname>] Pro Feld des Select-Statements können zusätzliche Formatierungs- (Default ist das String-Format) und/oder vom Feldinhalt abhängige Style-Definitionen getroffen werden.
Beispiel:	[View_Order.ord_valid_from] format=fsd [View_Order.ord_volume] format=ffixed decimals=10.2 StyleCondition=Order_Highlight [View_Order.cus_name] CellIcon=Cus_logo
Variable:	Format
Werte:	{fsd, fst, fsdst, fd, ft, fdt, fGeneral, fExponent, fFixed, fNumber, fCurrency, fBlob, fMemo, fRichMemo, - fsd (stringDate) Datum z.B. yyyyymmdd, siehe "DateFormat" - fst (stringTime) Uhrzeit z.B. hhnss, siehe "TimeFormat" - fsdst (stringDateTime) Datum und Uhrzeit z.B. yyyyymmddhhnss, siehe "DateFormat"/"TimeFormat" - fd, ft, fdt sind die entsprechende Formate für Anzeigefelder unabhängig von "DateFormat"/"TimeFormat" Einstellungen - fgeneral (general number format) kürzeste Darstellung von fexponent oder ffixed - fexponent -d.ddd...E+dddd - ffixed (fixed-point format) -ddd.ddd... - fnumber wie ffixed inkl. Tausenderpunkt - fcurrency Währungsformat entsprechend der lokalen Windows-Einstellungen - fBlob (Binary Large Objekt) - fMemo Memo-Format (nur für PAM-Felder) - fRichMemo Memo-Format inkl. Texteditor (nur für PAM-Felder) - fListView spezielles Format zur Speicherung von Dokumenten (nur für PAM-Felder) - fSimpleListView spezielles Format zur Speicherung genau eines Dokumentes (nur für PAM-Felder) - fSelect Format zur Darstellung von Auswahllisten im PAM
Variable:	Decimals
Werte:	x.y Angabe der maximalen Anzahl von Ziffern (x) einer Zahl, sowie die Anzahl von Nachkommastellen (y).
Variable:	StyleCondition
Werte:	<styleconditionname> Für ein View-Feld kann in Abhängigkeit von einer Bedingung ein separater Style (Hintergrund-, Schriftfarbe, Font, ...) gesetzt werden. Die Bedingung und der jeweilige Style werden in einer Sektion mit dem Namen der "StyleCondition" in der "APPGLOBALS/DESIGN.cdd" definiert.
Variable:	CellIcon
Werte:	<fieldname> "CellIcon" bietet die Möglichkeit einem Feld der View ein Icon voranzustellen. Das Icon selbst muss in einem separaten Feld der View ("<fieldname>") zur Verfügung stehen.
Variable:	Summary
Werte:	stcount stsum stmin stmax staverage ststddev ststddevp stvariance stvariancep Definiert für die Datenzelle einer Pivot view wie dessen Werte aggregiert werden sollen.
Sektion:	View Field
Name:	Die unter dem View-Parameter "Filters" zugeordneten Filter werden in dieser Sektion definiert. Ein Filter kann leer sein. In diesem Fall enthält der Filter keine Bedingung und zeigt alle Daten der View an. Eine Bedingung zur Einschränkung der Datenmenge kann mit Hilfe aller View-Felder und der Parameter als Boolescher Ausdruck formuliert werden.
Beispiel:	
Variable:	
Werte:	

Sektion:	View Aktionen
Name:	[<viewname>.Action] In der Action-Sektion werden die Forms bestimmt, die bei einem Doppelklick auf eine View-Zeile oder auf ein bestimmtes Feld einer Zeile aufgerufen werden sollen. Das eine weitere Form aus dem SAM heraus aufgerufen werden kann, wird für den Anwender dadurch sichtbar, dass der Mauszeiger vom Pfeil zum Handsymbol wechselt.
Beispiel:	[View_Order.Action] OnDBLClick.cus_name=Form.Frm_PAM_Customer OnDBLClick=Form.Frm_PAM_Order
Variable:	OnDbIClick
Werte:	{Form.<formname> ModalForm.<formname> RefForm.<formname> System.<funktionname> Bei einem Doppelklick auf ein beliebiges Feld einer Viewzeile wird in der Regel das zugeordnete Form (alternativ ein modales oder Referenz-Form) geöffnet. Die Felder der Viewzeile stehen im aufgerufenen Form als Parameter zur Verfügung. Es besteht auch die Möglichkeit eine Systemfunktion aufzurufen oder ein Programm zu starten.
Variable:	OnDbIClick.Field
Werte:	{Form.<formname> ModalForm.<formname> RefForm.<formname> System.<funktionname> Bei einem Doppelklick auf ein bestimmtes Feld einer Viewzeile wird in der Regel das zugeordnete Form (alternativ ein modales oder Referenz-Form) geöffnet. Die Felder der Viewzeile stehen im aufgerufenen Form als Parameter zur Verfügung. Es besteht auch die Möglichkeit eine Systemfunktion aufzurufen oder ein Programm zu starten.
Sektion:	DropAllowed Liste der View
Name:	[<viewname>.DropAllowedList] Diese Sektion wird benötigt um Drag&Drop-Funktionalität zwischen 2 Views einzurichten. Die "DropAllowedList" definiert aus welchen Views Werte in die aktuelle View gezogen werden dürfen und wie die (Drop-) Aktion heißen soll, die in einer weiteren Sektion definiert werden muss.
Beispiel:	[View_Customer.DropAllowedList] FRM_Order.View_Order=Delete_Customer
Variable:	Form.View
Werte:	<DropActionName> In der "DropAllowList"-Sektion einer View werden den Views (vollständiger Name, inkl. Formname), aus denen Daten auf diese View gezogen werden, die Sektionsnamen zugeordnet, in denen die Drop-Aktionen definiert werden.

Sektion:	Drop Aktion
Name:	[<viewname>.<DropActionName>] In dieser Sektion wird die Datenbank-Operation definiert, die durch die Drag&Drop-Aktion ausgelöst wurde.
Beispiel:	[View_Customer.Delete_Customer_ID] Provider=ADO.myDB ApplyTable=ORD IndexName=ORD_ID Operation=Update WhereCondition=ORD_ID = :DRAG_ORD_ID AbortOnError=False
Variable:	Provider
Werte:	{ADO.<providername> MSA.<msaDbName> FILE.DIR LOCAL Mail} Angabe des Providers, der die Daten bereitstellen soll. Aktuell stehen fünf unterschiedliche Typen (ADO, MSA, FILE, LOCAL, Mail) zur Verfügung. Während ADO für den klassischen Datenbankzugriff benutzt wird, können mit Hilfe von MSA (Multiple Source Access) Daten aus unterschiedlichen Datenbanken in eine speicherinterne Datenbank geladen werden. Der FileProvider stellt Daten des Filesystems systemkonform per SQL zur Verfügung. Der LocalProvider greift wiederum auf eine speicherinterne Datenbank (MemoryTable) zu, die zuvor vom cyAgent erstellt wurde. Der LocalProvider unterstützt zur Zeit nur PAMs. Der MSAProvider und der LocalProvider unterstützen nur den lesenden Zugriff. Der MailProvider unterstützt den Mailversand, falls ein PAM mit "MailSupport=true" definiert wird.
Variable:	IndexName
Werte:	<fieldname> (<fieldname_2> ...) Die Variable "Indexname" legt fest, welches Datenbankfeld den Index beinhaltet. Der Index kann sich auch aus mehreren Feldern zusammensetzen, falls es kein numerischer Index ("CounterIndex=False") ist.
Variable:	ApplyTable
Werte:	<tablename> Name der Tabelle, die geändert werden soll.
Variable:	CounterIndex
Werte:	{True False} Falls der Index der ApplyTable (Tabelle, die geändert werden soll) aus einem numerischen Feld besteht, der beim Insert automatisch gesetzt werden soll, ist "CounterIndex=True" zu setzen (default). Setzt sich der Index aus mehreren Feldern zusammen bzw. ist er nicht numerisch, muss "CounterIndex=False" gesetzt werden.
Variable:	ExecBefore
Werte:	<MethodName.ParameterName> Aufruf einer Methode der "<application>.dll" zum Zeitpunkt vor der Ausführung des SQL-Befehls.
Variable:	ExecAfter
Werte:	<MethodName.ParameterName> Aufruf einer Methode der "<application>.dll" zum Zeitpunkt nach der Ausführung des SQL-Befehls.
Variable:	Operation
Werte:	{Insert Delete Update} Angabe der Datenbank-Operation, die bei einem Drop durchgeführt werden soll.
Variable:	WhereCondition
Werte:	Boolscher Ausdruck Die "WhereCondition" bestimmt den zu ändernden Datensatz. Dabei kann auf die Daten der Drag- bzw. Drop-Datenmenge über den Präfix ":drag" bzw. ":drop" zurückgegriffen werden. Wird der Wert einer einzelnen angeklickten Zelle eines Datensatzes benötigt, steht dafür der Parameter ":CLICKVALUE" zur Verfügung.
Variable:	AbortOnError
Werte:	{False True} Mit Hilfe dieser Variablen kann eine Drag&Drop-Aktion, die mehrere Datenbankänderungen beinhaltet (Multiselect), beim Auftreten eines Fehlers abgebrochen werden. Alle Datenbankänderungen bis zum Auftreten des Fehlers werden durchgeführt. Bei "AbortOnError=False" (default) wird nicht abgebrochen, d.h. alle fehlerfreien Datenbankänderungen werden durchgeführt.

Sektion:	Wertzuweisungen für Felder bei einer Drop Aktion
Name:	[<viewname>.<DropActionName>.Fields] In der Fields-Sektion erfolgt abschließend die Wertzuweisung für die Felder, die mit der Drag&Drop-Aktion geändert werden sollen. Dabei stehen alle Parameter der Masterview-Kaskade, des Drag- und des Drop-Datensatzes sowie alle System- und Anwendungsparameter und der Parameter ":Clickvalue" zur Verfügung.
Beispiel:	[View_Customer.Delete_Customer_ID.Fields] cus_id=NULL ord_modified=:AD ord_modifier=:USER
Variable:	
Werte:	
Sektion:	Clickmap
Name:	[<viewname>.CLICKMAP] In einer "Clickmap" werden Feldern, die im SAM angezeigt werden, alternative Felder (in der Regel mit nichtsprechenden Schlüsseln) zugeordnet. Klickt der Anwender eines dieser Felder an, wird der Wert des gemappten Feldes in den Parameter ":Clickvalue" übertragen. "Clickmaps" werden in der Regel bei denormalisierten Views benötigt, um eine Drag&Drop-Aktion oder ein neues PAM in Abhängigkeit eines bestimmten Schlüsselwertes einer angeklickten Zelle zu öffnen.
Beispiel:	[View_Order_Top10.Clickmap] ord_name1=ord_id1 ord_name2=ord_id2 ord_name3=ord_id3 ...
Variable:	
Werte:	
Sektion:	Definition eines View-Reports
Name:	[<viewname>.<reportname>] In dieser Sektion werden die vom Report benötigten Datasets benannt. Jedes Dataset wird in einer weiteren Untersektion weiter definiert.
Beispiel:	[View_Order.myReport] Datasets=<dataset_name_1> <dataset_name_2> ... <dataset_name_n>
Variable:	Datasets
Werte:	<dataset_name_1> <dataset_name_2> ... <dataset_name_n> Aufzählung der Datasets, die einem Report zur Verfügung gestellt werden .
Variable:	SelectionSort
Werte:	<fieldname_1> <fieldname_2> ... <fieldname_n> Die Variable "SelectionSort" bietet die Möglichkeit einen Report zu sortieren. Es wird ein Index mit den SelectionSort-Feldern für das Report-Dataset erstellt.

Sektion:	Dataset-Definitionen eines View-Reports
Name:	[<viewname>.<reportname>.<datasetname>] Sektion mit den Definitionen eines Datasets, das im zugehörigen Report benutzt wird.
Beispiel:	[View_Order.myReport.ds2] Provider=ADO.myDB Bands=detail_data Source=SQL Master=ds1.ord_ID
Variable:	Provider
Werte:	{ADO.<providername> MSA.<msaDbName> FILE.DIR LOCAL Mail} Angabe des Providers, der die Daten bereitstellen soll. Aktuell stehen fünf unterschiedliche Typen (ADO, MSA, FILE, LOCAL, Mail) zur Verfügung. Während ADO für den klassischen Datenbankzugriff benutzt wird, können mit Hilfe von MSA (Multiple Source Access) Daten aus unterschiedlichen Datenbanken in eine speicherinterne Datenbank geladen werden. Der FileProvider stellt Daten des Filesystems systemkonform per SQL zur Verfügung. Der LocalProvider greift wiederum auf eine speicherinterne Datenbank (MemoryTable) zu, die zuvor vom cyAgent erstellt wurde. Der LocalProvider unterstützt zur Zeit nur PAMs. Der MSAProvider und der LocalProvider unterstützen nur den lesenden Zugriff. Der MailProvider unterstützt den Mailversand, falls ein PAM mit "MailSupport=true" definiert wird.
Variable:	Bands
Werte:	<bandname_1> <bandname_2> ... Liste der Bänder (bands oder specialband), die im SAM bestimmte Spalten mit einem gemeinsamen Bandheader zusammenfassen und/oder Attribute eines Datensatzes untereinander platzieren oder im Falle von specialband einen zusätzlichen Header im SAM einfügen. Die Variable wird ebenso bei Reports benutzt, um ein Dataset einem Band im Report zuzuordnen. Im Report muss allerdings zuvor ein Band mit gleichem Namen angelegt worden sein.
Variable:	SortOrder
Werte:	<fieldname_1> <fieldname_2> ... Vorgabe einer Sortierung nach mehreren Feldern in Report-Datasets oder Views. Ein Minus vor dem Feldnamen veranlasst eine absteigende Sortierung. In Views ist für den Anwender eine Sortierung nach mehreren Feldern per "Shift-Klick" möglich. Klickt der Anwender ohne "Shift" auf eine Spalte, wird nur nach dieser Spalte sortiert. Per "Reset-Button" wird die vorgebene Sortierung wiederhergestellt.
Variable:	Source
Werte:	{SQL Selection} "Source=SQL" weist darauf hin, dass noch eine Untersektion mit der SQL-Abfrage für die Daten des Datasets existiert. Bei "Source=Selection" wird das Dataset mit den selektierten Daten der View gefüllt.
Variable:	Master
Werte:	<master_dataset_name>.<fieldname> Für die Darstellung einer Master-Detail-Beziehung wird die Variable "Master" benötigt. Zu dem referenzierten Feld des Masters werden im aktuellen Dataset weitere Detail-Daten zur Verfügung gestellt.
Sektion:	SQL-Abfrage zum View-Report-Dataset
Name:	[<viewname>.<reportname>.<datasetname>.SQL] Untersektion zum Report-Dataset, die das SQL-Statement zur Datenbeschaffung enthält.
Beispiel:	[View_Order.myReport.ds2.SQL] Select * from ord
Variable:	
Werte:	

Sektion:	PAM
Name:	[<PAMname>] Sektion mit den Definitionen eines PAMs. Mit einem PAM werden in der Regel Daten manipuliert (Insert, Delete, Update).
Beispiel:	[cyPAM_Order_1] Deny=Insert Delete Container=C1.Last Provider=MSA.myDB ApplyProvider=ADO.myDB ActiveStyleSheetName=reppredef_Standard IndexName=ord_id ApplyTable=ord CouterIndex=true DateFormat=YYYYMMDD TimeFormat=HHNNSS ViewMaster=View_Order InfoFields=cus_name HiddenBars=barReport barUtilities HiddenMenus=Send ExecBefore=myMethod.myParam ExecAfter=myMethod.myParam AddParams=ord_id CloneFrom=cyPAM_Order MailSupport=False MemoryTables=Sales Orders
Variable:	Deny
Werte:	{INSERT UPDATE DELETE} In einem PAM können nicht gewünschte Datenbankänderungen ausgeschlossen werden.
Variable:	Container
Werte:	{<cname>.FIRST <cname>.LAST <cname>.FIRST.<tablename> <cname>.LAST.<tablename>} Zuordnung eines SAMs oder PAMs zu einem Container. Container können in zwei Teile gesplittet ("FIRST", "LAST") sein und damit maximal zwei SAMs/PAMs gleichzeitig beinhalten. Zusätzlich können einem Container noch beliebig viele Tabulatoren zugeordnet werden. Hinter jedem dieser Tabulatoren können wiederum bis zu zwei SAMs/PAMs liegen.
Variable:	Provider
Werte:	{ADO.<providername> MSA.<msaDbName> FILE.DIR LOCAL Mail} Angabe des Providers, der die Daten bereitstellen soll. Aktuell stehen fünf unterschiedliche Typen (ADO, MSA, FILE, LOCAL, Mail) zur Verfügung. Während ADO für den klassischen Datenbankzugriff benutzt wird, können mit Hilfe von MSA (Multiple Source Access) Daten aus unterschiedlichen Datenbanken in eine speicherinterne Datenbank geladen werden. Der FileProvider stellt Daten des Filesystems systemkonform per SQL zur Verfügung. Der LocalProvider greift wiederum auf eine speicherinterne Datenbank (MemoryTable) zu, die zuvor vom cyAgent erstellt wurde. Der LocalProvider unterstützt zur Zeit nur PAMs. Der MSAProvider und der LocalProvider unterstützen nur den lesenden Zugriff. Der MailProvider unterstützt den Mailversand, falls ein PAM mit "MailSupport=true" definiert wird.
Variable:	ApplyProvider
Werte:	{ADO.<providername> MSA.<providername> FILE.DIR} Angabe eines weiteren Providers, der nur für die Änderung der Daten verantwortlich ist. Werden z. B. über einen MSA-Provider Daten selektiert, der keine Datenbankänderungen unterstützt, können Änderungen über den ApplyProvider an der ApplyTable vorgenommen werden. Wird kein "ApplyProvider" definiert, wird der mittels der Variablen "Provider" bestimmte Provider für Selektion und Änderung benutzt.
Variable:	ActiveStyleSheetName
Werte:	REPPREDEF_<stylename> Zuordnung eines initialen Styles. Die Styles-Definitionen befinden sich in der "styles.cdd". Wird die Variable "ActiveStyleSheetName" nicht gesetzt, versucht der cyAgent den Stylesheet "DevExpress" zuzuordnen. Sollte die Sektion "[DevExpress]" nicht in der "styles.cdd" vorhanden sein, wird der erste Stylesheet aus der "styles.cdd" zugeordnet.

Variable:	IndexName
Werte:	<fieldname> (<fieldname_2> ...) Die Variable "Indexname" legt fest, welches Datenbankfeld den Index beinhaltet. Der Index kann sich auch aus mehreren Feldern zusammensetzen, falls es kein numerischer Index ("CounterIndex=False") ist.
Variable:	ApplyTable
Werte:	<tablename> Name der Tabelle, die geändert werden soll.
Variable:	AutoIndex
Werte:	{True False} Bei "AutoIndex=True" wird der Index beim Insert-Statement von der Datenbank automatisch ermittelt. Der Default-Wert ist "True".
Variable:	CounterIndex
Werte:	{True False} Falls der Index der ApplyTable (Tabelle, die geändert werden soll) aus einem numerischen Feld besteht, der beim Insert automatisch gesetzt werden soll, ist "CounterIndex=True" zu setzen (default). Setzt sich der Index aus mehreren Feldern zusammen bzw. ist er nicht numerisch, muss "CounterIndex=False" gesetzt werden.
Variable:	DateFormat
Werte:	{YYYYMMDD ...} YYYY = Year, MM = Month, DD = Day Die Variable "DateFormat" dient dazu das Format des Datumseditors an das Format der Datumsfelder der View oder des PAMs anzupassen. Der im SQL-Statement oder im Filter verwendbare Standard-Parameter ":SD" erhält ebenfalls dieses Datumsformat.
Variable:	ReportDesigner
Werte:	{False True} "ReportDesigner=True" bietet die Möglichkeit den FastReport-Designer (kostenpflichtige Komponente) zu starten.
Variable:	TimeFormat
Werte:	{HHNNSS ...} HH = Hour, NN = Minutes, SS = Seconds "TimeFormat" bestimmt das Zeitformat des Datumseditors.
Variable:	ViewMaster
Werte:	<viewname> Zuordnung einer MasterView. Alle selektierten Werte der MasterView (und deren MasterView, ..., also der gesamten MasterView-Kaskade) stehen damit im aktuellen SAM/PAM als Parameter (" :<fieldname>") zur Verfügung.
Variable:	InfoFields
Werte:	(SQL.)<fieldname_1> (SQL.)<fieldname_2> ... (SQL.)<fieldname_n> Die unter "InfoFields" definierten Felder der View werden in der obersten Zeile (über der Toolbar-Zeile, unterhalb evtl. Tabs) des SAMs, PAMs angezeigt. Angezeigt werden können Felder der Masterview oder Felder die per SQL Statement ermittelt werden. Letztere werden durch ein vorangestelltes "SQL." gekennzeichnet. Das SQL Statement selber befindet sich in einer zusätzlichen Sektion [<fieldname.SQL].
Variable:	InfoStyle
Werte:	<stylename> Die Info Fields der SAMs oder PAMs erhalten den Style "<stylename>" des aktiven Stylesheets. Diese Zuordnung kann direkt im Form beim SAM oder PAM erfolgen oder global für alle SAMs und PAMs in der "Design.cdd".
Variable:	HiddenBars
Werte:	{barPosition barReport barUtilities barDate barFilter barRecord barMail} Mit "HiddenBars" können einzelne Bars in SAMs oder PAMs ausgeblendet werden.
Variable:	HiddenMenus
Werte:	{FR Word Excel Pdf Html Xml Next Prior First Last Style Filter Date Del Ins Upd Mit "HiddenMenus" können einzelne Menü-Buttons in SAMs oder PAMs ausgeblendet werden. Dabei steht "FR" für den FastReport-Designer und "Send" für den "Mail-versenden-Button".

Variable:	ExecBefore
Werte:	<MethodName.ParameterName> Aufruf einer Methode der "<application>.dll" zum Zeitpunkt vor der Ausführung des SQL-Befehls.
Variable:	ExecAfter
Werte:	<MethodName.ParameterName> Aufruf einer Methode der "<application>.dll" zum Zeitpunkt nach der Ausführung des SQL-Befehls.
Variable:	AddParams
Werte:	<ParameterName_1> <ParameterName_2> ... <ParameterName_n> Mit "Addparams" werden einem Selektionsdataset (die selektierte Datenmenge steht abhängigen PAMs/SAMs als Parameter zur Verfügung) zusätzliche Parameter übergeben. Die Parameter in der AddParams-Liste müssen ohne ":" geschrieben werden.
Variable:	AddApplyParams
Werte:	<ParameterName_1> <ParameterName_2> ... <ParameterName_n> Mit "AddApplyParams" werden zusätzliche Parameter einem Applydataset übergeben. Die Parameter in der "AddApplyParams"-Liste müssen ohne ":" geschrieben werden.
Variable:	CloneFrom
Werte:	<PAMname> Das PAM hat kein eigenes SQL-Statement sondern bezieht sich als "Clone" auf die bereits selektierten Daten eines anderen PAMs. Für den Fall, dass sehr viele Daten einer Tabelle zu ändern sind und der Platz in einem PAM nicht ausreicht, kann "CloneFrom" verwendet werden.
Variable:	MailSupport
Werte:	{False True} "MailSupport=True" aktiviert die Möglichkeit aus dem PAM heraus emails zu versenden.
Variable:	MemoryTables
Werte:	<MTname_1> <MTname_2> <MTname_n> MemoryTables sind speicherinterne Datenbanken, die lokal vom cyAgent erzeugt werden. Der Aufbau einer MemoryTable wird in den Sektionen "[<MTname]" und "[<MTname.SQL]" definiert. Um auf eine lokale Datenbank zuzugreifen muss zusätzlich "Provider=Local" definiert sein. Verwendung finden MemoryTables vorallem im MSA-Umfeld, um für Auswahlfelder ("Format=fSelect") die Daten bereitzustellen.
Variable:	RightBorder
Werte:	n Abstand der mit akRight ausgerichteten Felder zum rechten Rand in Pixel (default = 10).
Sektion:	SQL-Abfrage des PAMs
Name:	[<PAMname>.SQL] Untersektion zum PAM, die das SQL-Statement zur Datenbeschaffung enthält. Die Reihenfolge der Datenbankfelder in der Select-Clause sollte mit der Reihenfolge der Anordnung der Felder im PAM (von links oben nach rechts unten) wegen der Cursorsteuerung übereinstimmen. Für das SQL-Statement stehen alle Parameter der Masterview-Kaskade, alle System- und Anwendungsparameter und der Parameter ":Clickvalue" zur Verfügung.
Beispiel:	[cyPAM_Order.SQL] select * from ord where ord.ord_id = :ord_id
Variable:	
Werte:	

Sektion:	PAM Feld Definition
Name:	[<PAMname>.<PAMfieldname>] Für jedes Feld eines PAMs, das auf dem Formular erscheinen soll bzw. für die Datenbankänderung benötigt wird, müssen in einer eigenen Sektion weitere Definitionen hinterlegt werden.
Beispiel:	[cyPAM_Order.ord_id] format=ffixed decimals=10.0 Enabled=False Visible=True LabelAlign=Left HAlign=Left VAlign=Top FieldTop=10 FieldLeft=10 FieldWidth=100 FieldHeight=50 Fieldtypes=isIndexField, isNotNullField PanelAnchors=akLeft, akTop InitValue=:ord_id
Variable:	Format
Werte:	{fsd, fst, fsdst, fd, ft, fdt, fGeneral, fExponent, fFixed, fNumber, fCurrency, fBlob, fMemo, fRichMemo, - fsd (stringDate) Datum z.B. yyyyymmdd, siehe "DateFormat" - fst (stringTime) Uhrzeit z.B. hhnnss, siehe "TimeFormat" - fsdst (stringDateTime) Datum und Uhrzeit z.B. yyyyymmddhhnnss, siehe "DateFormat"/"TimeFormat" - fd, ft, fdt sind die entsprechende Formate für Anzeigefelder unabhängig von "DateFormat"/"TimeFormat" Einstellungen - fgeneral (general number format) kürzeste Darstellung von fexponent oder ffixed - fexponent -d.ddd...E+dddd - ffixed (fixed-point format) -ddd.ddd... - fnumber wie ffixed inkl. Tausenderpunkt - fcurrency Währungsformat entsprechend der lokalen Windows-Einstellungen - fBlob (Binary Large Objekt) - fMemo Memo-Format (nur für PAM-Felder) - fRichMemo Memo-Format inkl. Texteditor (nur für PAM-Felder) - fListView spezielles Format zur Speicherung von Dokumenten (nur für PAM-Felder) - fSimpleListView spezielles Format zur Speicherung genau eines Dokumentes (nur für PAM-Felder) - fSelect Format zur Darstellung von Auswahllisten im PAM
Variable:	FileNameField
Werte:	<pam field> Ein PAM Feld vom Format "fSimpleListView" benötigt eine Referenz auf ein Datenbankfeld, das den Namen der Datei enthält, die im "fSimpleListView" Feld gespeichert wurde bzw. gespeichert werden soll.
Variable:	checked
Werte:	{<string> null} Die Variable "checked" kennzeichnet das PAM Feld als ein spezielles Feld, bei dem der Anwender einen von 3 Zuständen (check, unchecked, undefined) auswählen kann. Der Wert der Variablen (null oder ein String/Character) wird in die Datenbank übernommen.
Variable:	unchecked
Werte:	{<string> null} Die Variable "unchecked" kennzeichnet das PAM Feld als ein spezielles Feld, bei dem der Anwender einen von 3 Zuständen (check, unchecked, undefined) auswählen kann. Der Wert der Variablen (null oder ein String/Character) wird in die Datenbank übernommen.
Variable:	undefined
Werte:	{<string> null} Die Variable "undefined" kennzeichnet das PAM Feld als ein spezielles Feld, bei dem der Anwender einen von 3 Zuständen (check, unchecked, undefined) auswählen kann. Der Wert der Variablen (null oder ein String/Character) wird in die Datenbank übernommen. Bei einem Muss-Feld (Fieldtype=isNotNullField) wird der Zustand undefined nicht unterstützt.

Variable:	UseThousandSeparator
Werte:	{False True} Die Variable regelt die Verwendung des Tausenderpunktes bei numerischen Feldern.
Variable:	Decimals
Werte:	x.y Angabe der maximalen Anzahl von Ziffern (x) einer Zahl, sowie die Anzahl von Nachkommastellen (y).
Variable:	Enabled
Werte:	{True False} Die Variable "Enabled" legt fest, ob ein Feld eingabebereit (True) ist oder nur angezeigt (False) wird.
Variable:	EnableOnInsert
Werte:	{True False} Für den Fall, dass ein Mussfeld beim Insert-Vorgang nicht automatisch von der Masterview bereitgestellt werden kann und es trotzdem bei Anzeige- /Update-Vorgängen verborgen bleiben soll, gibt es die Variable "EnableOnInsert=True". Damit wird dieses Feld nur beim Insert sichtbar und eingabebereit.
Variable:	AllowWrite
Werte:	{False True} Die Variable "AllowWrite" ermöglicht die Eingabe in einem Referenzfeld. Der eingegebene Wert wird im Parameter :LookupValue gespeichert und kann im Referenzformular zur Datenselektion benutzt werden.
Variable:	Visible
Werte:	{True False} Die Variable "Visible" legt fest, ob ein Feld sichtbar (True) ist oder verborgen (False) bleibt.
Variable:	PasswordValue
Werte:	{False True} Die Variable "PasswordValue" legt fest, ob ein Passwort (vom Typ String) unlesbar (True) ist oder lesbar (False) bleibt.
Variable:	HashValue
Werte:	{False True} Ist "HashValue=True", dann wird aus dem Passwort der Hashwert gebildet. Hierbei werden die Salt-Werte berücksichtigt, die in der Masterview Kaskade, als APP-Parameter oder auch im PAM-Selektionsdataset definiert sein können. Der eingegebene String ist ähnlich wie bei "PasswordValue=True" nicht lesbar.
Variable:	LabelAlign
Werte:	{Left Center Right} Die Variable "LabelAlign" definiert die Ausrichtung der Überschrift eines Feldes.
Variable:	HAlign
Werte:	{Left Center Right} Die Variable "HAlign" dient zur horizontalen Ausrichtung des Feldesinhaltes. Ohne diese Variable werden Character-Felder standardmäßig links und numerische Felder standardmäßig rechts ausgerichtet.
Variable:	VAlign
Werte:	{Top Center Bottom} Die Variable "VAlign" dient zur vertikalen Ausrichtung des Feldes.
Variable:	FieldTop
Werte:	n Positionierung des Feldes: Abstand n Pixel zum oberen Rand des PAMs.
Variable:	FieldLeft
Werte:	n Positionierung des Feldes: Abstand n Pixel zum linken Rand des PAMs.
Variable:	FieldWidth
Werte:	n Breite des Feldes in Pixel.

Variable:	FieldHeight
Werte:	n Höhe des Feldes in Pixel.
Variable:	FieldTypes
Werte:	{isIndexField isReadOnlyField isNotNullField isModifierField} Der Variablen "Fieldtypes" können ein oder mehrere durch Kommata getrennte Werte zugewiesen werden. Die Bedeutung der Fieldtypes im einzelnen: - isIndexField = PAM-Feld enthält den Index der ApplyTable - isReadOnlyField = PAM-Feld ist ein Anzeigefeld (wie "enabled=false"), das aber auf Events (z.B. rechter Mausklick) reagiert - isNotNullField = PAM-Feld ist ein Mussfeld - isModifierFiled = PAM-Feld dient zur Dokumentation von Datensatzänderungen
Variable:	PanelAnchors
Werte:	{akLeft, akTop, akRight, akBottom} "PanelAnchors" legt fest, wie sich das Feld verhalten soll, wenn der Anwender die PAM-Ränder verändert. Ist zum Beispiel ein Feld mit dem rechten Rand verankert ("akRight") und zieht der Anwender den rechten Rand des PAMs größer wird das Feld automatisch nach rechts vergrößert.
Variable:	InitValue
Werte:	{:<paramtername> SQL.<Provider>} Mit der Variablen "InitValue" werden Pamfelder initialisiert. Die Initialisierung erfolgt im Insert-Status, wenn das Feld leer ist. Der Initialisierungswert ist entweder der Wert eines Parameters oder er wird mittels Select ("SQL.<Provider>") aus einer Tabelle gelesen. Als Parameter sind alle Felder des PAM-Selektionsdatasets, die Parameter der Masterview-Kaskade und die vordefinierten Parameter zulässig. Sie müssen mit dem Präfix ":" definiert werden. Soll der Wert mittels SQL ermittelt werden, ist eine zusätzliche Sektion ["<pamname>.<fieldname>.SQL"] erforderlich.
Variable:	SaveValue
Werte:	{:<paramtername> SQL.<Provider>} Die Variable "SaveValue" bietet die Möglichkeit einen anderen Wert, anstelle des Anzeigewertes abzuspeichern. Wie bei "InitValue" erfolgt die Wertezuordnung entweder mittels Parameter oder SQL-Anweisung. Drückt der Anwender die Speichertaste, dann wird anstelle des Anzeigewertes der "SaveValue" gespeichert.
Variable:	Data
Werte:	{SQL SQL.local} Mit der Variablen "InitValue" werden Pam-Felder initialisiert. Die Initialisierung erfolgt im Insert-Status, wenn das Feld leer ist. Der Initialisierungswert ist entweder der Wert eines Parameters oder er wird mittels Select ("SQL.<Provider>") aus einer Tabelle gelesen. Für PAM-Felder vom "Format=fSelect" definiert die Variable "Data" über welchen Provider die Daten besorgt werden sollen. Während bei "Data=SQL" der Provider des PAMs benutzt wird, stellt bei "Data=SQL.local" der LocalProvider die Daten aus einer MemoryTable zur Verfügung.
Variable:	InsertFromMaster
Werte:	{True False} Ist sichergestellt, dass beim Insert ein PAM-Feld mit dem Wert eines Feldes einer MasterView automatisch gefüllt werden kann, ist das mit "InsertFromMaster=True" möglich.
Variable:	ModifierType
Werte:	{mtCreator mtCreatedOn mtModifier mtModifiedOn} Ein PAM-Feld vom Typ "isModifierField" hat genau einen der folgenden ModifierTypes: - mtCreator (gespeichert wird der Name des User, der den Datensatz angelegt hat) - mtCreatedOn (gespeichert wird das Insert-Datum des Datensatzes) - mtModifier (gespeichert wird der Name des User, der den Datensatz zuletzt geändert hat) - mtModifiedOn (gespeichert wird das letzte Update-Datum des Datensatzes)

Variable:	ApplyTarget
Werte:	{<fieldname>} Soll ein PAM-Feld seinen Wert nicht durch Eingabe sondern durch Auswahl aus einem zusätzlichen (Referenz-) SAM erhalten (siehe Untersektion Action) werden 3 weitere Variablen ("ApplyTarget", "ReferenzSelection", "ReferenzTarget") zur Zurodnung der Daten benötigt. "ApplyTarget" bestimmt den Namen des Datenbankfeldes aus der Selektionsdatenmenge des PAMs, in das der referenzierte Wert gespeichert werden soll.
Variable:	ReferenceSelection
Werte:	{<fieldname>} "ReferenceSelection" gibt den Namen des Feldes der Referenzdatenmenge an, dessen Inhalt im PAM, im aktuellen Feld, angezeigt werden soll.
Variable:	ReferenceTarget
Werte:	{<fieldname>} "ReferenceTarget" bestimmt den Namen des Feldes der Referenzdatenmenge, das in das zugehörige Feld des PAMs ("ApplyTarget") übertragen werden muss. Möglicherweise hat das gewünschte Feld im Referenz-SAM einen anderen Namen.
Variable:	setfilename
Werte:	<pamfieldname> Variable "SetFileName" für Memo-, Richmemo- und Image-Felder: Mit "SetFileName=<pamfieldname>" kann ein PAM-Feld mit dem Namen der Datei initialisiert werden. Die Initialisierung erfolgt, wenn der Anwender die Datei lädt.
Variable:	AllowImageInsert
Werte:	{True False} Die Variable regelt bei Richmemo-Feldern, ob diese Images aufnehmen dürfen.
Variable:	Wrap
Werte:	{True False} Die Variable schaltet bei Richmemo-Feldern den automatischen Zeilenumbruch ein.
Variable:	InitBindings
Werte:	<bindingname_1> <bindingname_2> ... <bindingname_n> Bindings bieten die Möglichkeit auf Feldinhalte zu reagieren. Die Definition eines Bindings erfolgt in einer eigenen Sektion. Es können mehrere bindings einem PAM-Feld zugeordnet werden. InitBindings triggern nachdem alle Felder des PAMs initialisiert sind, bzw. die InitValue's geladen wurden.
Variable:	LiveBindings
Werte:	<bindingname_1> <bindingname_2> ... <bindingname_n> Bindings bieten die Möglichkeit auf Feldinhalte zu reagieren. Die Definition eines Bindings erfolgt in einer eigenen Sektion. Es können mehrere bindings einem PAM-Feld zugeordnet werden. LiveBindings triggern wenn der Benutzer ein PAM Feld ändert.
Variable:	PostBindings
Werte:	<bindingname_1> <bindingname_2> ... <bindingname_n> Bindings bieten die Möglichkeit auf Feldinhalte zu reagieren. Die Definition eines Bindings erfolgt in einer eigenen Sektion. Es können mehrere bindings einem PAM-Feld zugeordnet werden. PostBindings triggern unmittelbar bevor der Datensatz in die Datenbank geschrieben oder aus der Datenbank gelöscht wird.

Sektion:	SQL-Abfrage für ein PAM Feld
Name:	[<PAMname>.<fieldname>.SQL] Untersektion zum PAM-Feld, die das SQL-Statement zur Datenbeschaffung enthält. Handelt es sich bei dem Feld um ein "fSelect-Feld" (Referenzfelder, die per Auswahlliste zugeordnet werden), muss die SQL-Abfrage eine Tabelle mit den beiden Spalten "Target" und "Selection" liefern. "Target" entspricht dem Fremdschlüssel, bzw. dem PAM-Feld, und "Selection" enthält den Text, der in der Auswahlliste angezeigt werden soll. Soll das PAM-Feld nur mit einem initialen Wert ("InitValue") gefüllt werden, muss die SQL-Abfrage ein Feld mit dem Alias "INIT" zurückliefern.
Beispiel:	Initvalue: [cyPAM_Order.cus_name.SQL] select cus_name as INIT from cus where cus.cus_id = :cus_id fSelect: [cyPAM_Order.cus_id.SQL] select cus_id as Target, cus_name as Selection from cus order by Selection
Variable:	TextColor
Werte:	{<RGB color>} Die Variable "TextColor" bietet die Möglichkeit den Text eines Feldes in einer bestimmten RGB Farbe darzustellen. Der Farbwert kann als Dezimalwert, Hexwert (z.B. \$006464FF) oder Konstante (z.B. clBlue) angegeben werden.
Sektion:	PAM Aktionen
Name:	[<pamname>.Action] Für PAM-Felder, die ihre Werte aus (Referenz-) SAMs übernehmen sollen, wird in dieser Sektion der Name des SAMs ("RefForm", Referenz-Formular) zugeordnet. Das "RefForm" wird als modales Fenster beim Anklicken des Feldes aufgerufen.
Beispiel:	[cyPAM_Order.Action] OnClick.cus_name=RefForm.Ref_SAM_Customer
Variable:	OnClick.Field
Werte:	{RefForm.<formname>} Wird ein Referenzfeld angeklickt, wird das zugeordnete Form (Referenz-SAM) geöffnet.
Sektion:	Definition eines PAM-Reports
Name:	[<pamname>.<reportname>] In dieser Sektion werden die vom Report benötigten Datasets benannt. Jedes Dataset wird in einer weiteren Untersektion weiter definiert.
Beispiel:	[cyPAM_Order.myReport] Datasets=<dataset_name_1> <dataset_name_2> ... <dataset_name_n>
Variable:	Datasets
Werte:	<dataset_name_1> <dataset_name_2> ... <dataset_name_n> Aufzählung der Datasets, die einem Report zur Verfügung gestellt werden .
Variable:	SelectionSort
Werte:	<fieldname_1> <fieldname_2> ... <fieldname_n> Die Variable "SelectionSort" bietet die Möglichkeit einen Report zu sortieren. Es wird ein Index mit den SelectionSort-Feldern für das Report-Dataset erstellt.

Sektion:	Dataset-Definitionen eines PAM-Reports
Name:	[<pamname>.<reportname>.<datasetname>] Sektion mit den Definitionen eines Datasets, das im zugehörigen Report benutzt wird.
Beispiel:	[cyPAM_Order.myReport.ds2] Provider=ADO.myDB Bands=detail_data Source=SQL Master=ds1.ord_ID
Variable:	Provider
Werte:	{ADO.<providername> MSA.<msaDbName> FILE.DIR LOCAL Mail} Angabe des Providers, der die Daten bereitstellen soll. Aktuell stehen fünf unterschiedliche Typen (ADO, MSA, FILE, LOCAL, Mail) zur Verfügung. Während ADO für den klassischen Datenbankzugriff benutzt wird, können mit Hilfe von MSA (Multiple Source Access) Daten aus unterschiedlichen Datenbanken in eine speicherinterne Datenbank geladen werden. Der FileProvider stellt Daten des Filesystems systemkonform per SQL zur Verfügung. Der LocalProvider greift wiederum auf eine speicherinterne Datenbank (MemoryTable) zu, die zuvor vom cyAgent erstellt wurde. Der LocalProvider unterstützt zur Zeit nur PAMs. Der MSAProvider und der LocalProvider unterstützen nur den lesenden Zugriff. Der MailProvider unterstützt den Mailversand, falls ein PAM mit "MailSupport=true" definiert wird.
Variable:	Bands
Werte:	<bandname_1> <bandname_2> ... Liste der Bänder (bands oder specialband), die im SAM bestimmte Spalten mit einem gemeinsamen Bandheader zusammenfassen und/oder Attribute eines Datensatzes untereinander platzieren oder im Falle von specialband einen zusätzlichen Header im SAM einfügen. Die Variable wird ebenso bei Reports benutzt, um ein Dataset einem Band im Report zuzuordnen. Im Report muss allerdings zuvor ein Band mit gleichem Namen angelegt worden sein.
Variable:	SortOrder
Werte:	<fieldname_1> <fieldname_2> ... Vorgabe einer Sortierung nach mehreren Feldern in Report-Datasets oder Views. Ein Minus vor dem Feldnamen veranlasst eine absteigende Sortierung. In Views ist für den Anwender eine Sortierung nach mehreren Feldern per "Shift-Klick" möglich. Klickt der Anwender ohne "Shift" auf eine Spalte, wird nur nach dieser Spalte sortiert. Per "Reset-Button" wird die vorgebene Sortierung wiederhergestellt.
Variable:	Source
Werte:	{SQL Selection} "Source=SQL" weist darauf hin, dass noch eine Untersektion mit der SQL-Abfrage für die Daten des Datasets existiert. Bei "Source=Selection" wird das Dataset mit den selektierten Daten der View gefüllt.
Variable:	Master
Werte:	<master_dataset_name>.<fieldname> Für die Darstellung einer Master-Detail-Beziehung wird die Variable "Master" benötigt. Zu dem referenzierten Feld des Masters werden im aktuellen Dataset weitere Detail-Daten zur Verfügung gestellt.
Sektion:	SQL-Abfrage zum PAM-Report-Dataset
Name:	[<pamname>.<reportname>.<datasetname>.SQL] Untersektion zum Report-Dataset, die das SQL-Statement zur Datenbeschaffung enthält.
Beispiel:	[cyPAM_Order.myReport.ds2.SQL] Select * from ord
Variable:	
Werte:	

Sektion:	Definitionen für den Mailversand
Name:	[<pamname>.Mail] Wurde für ein PAM "MailSupport=True" gesetzt, werden in dieser Sektion alle für den Mailversand erforderlichen Definitionen hinterlegt.
Beispiel:	[cyPAM_Mail.Mail] Mailer=mySMTP_Server Provider=Mail Subject=:MDT_SUBJECT Header=:MDT_HEADER Text=:MDT_TEXT Footer=:MDT_FOOTER Attach=:MDT_ATTACH Address=:MDT_ADDRESS CC=:MDT_CC BCC=:MDT_CC MailFrom="CYFLEX GmbH" info@cyflex.de Priority=2
Variable:	Mailer
Werte:	{<mailername>} Die Variable "Mailer" ordnet die Mail einem in der "MailProvider.ini" definierten SMTP-Server zu.
Variable:	Provider
Werte:	{ADO.<providername> MSA.<msaDbName> FILE.DIR LOCAL Mail} Angabe des Providers, der die Daten bereitstellen soll. Aktuell stehen fünf unterschiedliche Typen (ADO, MSA, FILE, LOCAL, Mail) zur Verfügung. Während ADO für den klassischen Datenbankzugriff benutzt wird, können mit Hilfe von MSA (Multiple Source Access) Daten aus unterschiedlichen Datenbanken in eine speicherinterne Datenbank geladen werden. Der FileProvider stellt Daten des Filesystems systemkonform per SQL zur Verfügung. Der LocalProvider greift wiederum auf eine speicherinterne Datenbank (MemoryTable) zu, die zuvor vom cyAgent erstellt wurde. Der LocalProvider unterstützt zur Zeit nur PAMs. Der MSAProvider und der LocalProvider unterstützen nur den lesenden Zugriff. Der MailProvider unterstützt den Mailversand, falls ein PAM mit "MailSupport=true" definiert wird.
Variable:	Subject
Werte:	:<Paramtername> Dem Betreff der Mail wird ein PAM-Feld (Parameter) zugeordnet.
Variable:	Header
Werte:	:<Paramtername> <SQL Abfrage> Die Zuordnung von Kopfzeilen (Header) zu einer Mail kann über ein PAM-Feld (Parameter) oder über eine SQL-Abfrage erfolgen. Die SQL-Abfrage kann auch in einer weiteren Sektion hinterlegt werden. Siehe Sektion "pam.Mail.Header.SQL". Wird die Variable "Header" benutzt, hat diese immer Vorrang vor einer SQL-Sektion.
Variable:	Text
Werte:	:<Paramtername> Dem Text der Mail wird ein PAM-Feld (Parameter) zugeordnet.
Variable:	Footer
Werte:	:<Paramtername> <SQL Abfrage> Die Zuordnung von Fußzeilen (Footer) zu einer Mail kann über ein PAM-Feld (Parameter) oder über eine SQL-Abfrage erfolgen. Die SQL-Abfrage kann auch in einer weiteren Sektion hinterlegt werden. Siehe Sektion "pam.Mail.Footer.SQL". Wird die Variable "Footer" benutzt, hat diese immer Vorrang vor einer SQL-Sektion.
Variable:	Attach
Werte:	:<Paramtername> <SQL Abfrage> Dem Mailanhang der Mail wird ein PAM-Feld (Parameter) zugeordnet. Alternativ kann der Mailanhang auch über eine separate SQL-Abfrage, die in einer weiteren Sektion hinterlegt ist, ermittelt werden. Siehe Sektion "pam.Mail.Attach.SQL".

Variable: Address
<p>Werte: :<Paramtername> <SQL Abfrage></p> <p>Die Zuordnung der Adressaten zu einer Mail kann über ein PAM-Feld (Parameter) oder über eine SQL-Abfrage erfolgen. Die SQL-Abfrage kann auch in einer weiteren Sektion hinterlegt werden. Siehe Sektion "pam.Mail.Address.SQL". Wird die variable "Address" benutzt, hat diese immer Vorrang vor einer SQL-Sektion.</p> <p>Mehrere Adressen in einem Adressfeld sind durch Semikolon zu trennen.</p>
Variable: CC
<p>Werte: :<Paramtername></p> <p>Der Adressliste der Kopieempfänger der Mail wird ein PAM-Feld (Parameter) zugeordnet. Alternativ kann die Adressliste auch über eine separate SQL-Abfrage, die in einer weiteren Sektion hinterlegt ist, ermittelt werden. Siehe Sektion "pam.Mail.CC.SQL".</p> <p>Mehrere Adressen in der Liste sind durch Semikolon zu trennen.</p>
Variable: BCC
<p>Werte: :<Paramtername></p> <p>Der Adressliste der Blindkopieempfänger der Mail wird ein PAM-Feld (Parameter) zugeordnet. Alternativ kann die Adressliste auch über eine separate SQL-Abfrage, die in einer weiteren Sektion hinterlegt ist, ermittelt werden. Siehe Sektion "pam.Mail.BCC.SQL".</p> <p>Mehrere Adressen in der Liste sind durch Semikolon zu trennen.</p>
Variable: MailFrom
<p>Werte: :<Paramtername> <Konstante></p> <p>Dem Absender der Mail wird ein PAM-Feld (Parameter) oder eine Konstante zugeordnet. Alternativ kann der Absender auch über eine separate SQL-Abfrage, die in einer weiteren Sektion hinterlegt ist, ermittelt werden. Siehe Sektion "pam.Mail.MailFrom.SQL".</p>
Variable: Priority
<p>Werte: {0 1 2 3 4 }</p> <p>Festlegung der Mail-Priorität: 0-Höchste,1-Hoch,2-Normal,3-Niedrig,4-Niedrigste.</p>
Sektion: SQL Abfrage für den Mail-Footer
<p>Name: [<pamname>.Mail.Footer.SQL]</p> <p>Der Mailfooter wird mit Hilfe einer separaten SQL-Abfrage bestimmt. Diese Sektion ist nur aktiv, wenn der Parameter "Footer" nicht benutzt wird.</p> <p>Beispiel: [cyPAM_Mail.Mail.Footer.SQL] Select txt_footer as footer from txt where ...</p>
Variable:
Werte:
Sektion: SQL Abfrage für den Mail-Anhang
<p>Name: [<pamname>.Mail.Attach.SQL]</p> <p>Der Mailanhang wird mit Hilfe einer separaten SQL-Abfrage bestimmt.</p> <p>Beispiel: [cyPAM_Mail.Mail.Attach.SQL] Select txt_text as attach from txt where ...</p>
Variable:
Werte:
Sektion: SQL Abfrage für die Mail-Adressen
<p>Name: [<pamname>.Mail.Address.SQL]</p> <p>Sektion zur Ermittlung der Mail-Adressen mit Hilfe einer separaten SQL-Abfrage. Diese Sektion ist nur aktiv, wenn der Parameter Address nicht benutzt wird.</p> <p>Beispiel: [cyPAM_Mail.Mail.Address.SQL] Select mail_address from ...</p>
Variable:
Werte:

Sektion:	PAM Memory Table
Name:	[<pamname>.<MemoryTableName>] Zu jeder in der PAM-Sektion definierten Memorytable wird in einer eigenen Sektion der Provider für die Datenbeschaffung definiert. Der Aufbau dieser Speichertabellen wird vom cyAgent initiiert. Er ist erforderlich, wenn Daten über einen zusätzlichen Provider benötigt werden.
Beispiel:	[cyPAM_order.customer] Provider=ADO.myDB
Variable:	Provider
Werte:	{ADO.<providername> MSA.<msaDbName> FILE.DIR LOCAL Mail} Angabe des Providers, der die Daten bereitstellen soll. Aktuell stehen fünf unterschiedliche Typen (ADO, MSA, FILE, LOCAL, Mail) zur Verfügung. Während ADO für den klassischen Datenbankzugriff benutzt wird, können mit Hilfe von MSA (Multiple Source Access) Daten aus unterschiedlichen Datenbanken in eine speicherinterne Datenbank geladen werden. Der FileProvider stellt Daten des Filesystems systemkonform per SQL zur Verfügung. Der LocalProvider greift wiederum auf eine speicherinterne Datenbank (MemoryTable) zu, die zuvor vom cyAgent erstellt wurde. Der LocalProvider unterstützt zur Zeit nur PAMs. Der MSAProvider und der LocalProvider unterstützen nur den lesenden Zugriff. Der MailProvider unterstützt den Mailversand, falls ein PAM mit "MailSupport=true" definiert wird.
Sektion:	SQL-Abfrage einer Memory Table
Name:	[<pamname>.<MemoryTableName>.SQL] Untersektion zur MemoryTable-Sektion, die das SQL-Statement zur Datenbeschaffung enthält.
Beispiel:	[cyPAM_order.customer.SQL] select * from sal where sal.ord_id = :ord_id
Variable:	
Werte:	
Sektion:	Band
Name:	[<bandname>] Mit Hilfe von Bändern kann man Felder eines SAMs in einem Bereich (Band) horizontal und vertikal durch die Zuordnung eines Levels anordnen. Alle Felder mit gleichem Wert (Level) werden nebeneinander in einer Zeile (gleiche Hierarchiestufe) angezeigt. Andernfalls erscheinen Felder mit höherem Wert unterhalb des Feldes mit niedrigerem Wert. Ordnet man einer View Bands zu, müssen alle Felder, die sichtbar sein sollen, auch einem Band zugeordnet werden.
Beispiel:	[AddressBand] HeaderAlignmentHorz=taLeftJustify cus_street=0 cus_postbox=1 cus_city=1
Variable:	HeaderAlignmentHorz
Werte:	{taLeftJustify taCenter taRightJustify} Ausrichtung des Headers eines Bands oder Specialbands.

Sektion:	Specialband
Name:	[<specialbandname>] Specialbands werden zur Zeit für die Anzeige von Kalendertagen in einer zusätzlichen Spaltenüberschrift genutzt.
Beispiel:	[myBand] SpecialBand=fbSDOM Enum=SCH_** HeaderAlignmentHorz=taCenter
Variable:	HeaderAlignmentHorz
Werte:	{taLeftJustify taCenter taRightJustify} Ausrichtung des Headers eines Bands oder Specialbands.
Variable:	SpecialBand
Werte:	{fbSDOM fbLDOM} Ein Specialband kann aktuell vom Typ fbSDOM (Short Days of Month) oder vom Typ fbLDOM (Long Days of Month) sein. Bei diesen Typen werden die Band-Header (Wochentage) automatisch generiert. Den Tagen können noch unterschiedliche Styles zugeordnet werden. Dies geschieht in der Design-Datei "Design.cdd" (im Verzeichnis "\<applicationname>\lappglobals") in der Sektion "[DaysOfWeek]".
Variable:	Enum
Werte:	<präfix>** Die Variable "Enum" ermöglicht eine automatische Zuordnung von Feldern zum SpecialBand. Sie ist aber nur bei Feldern anwendbar, die in ihrem Namen eine fortlaufende Nummer führen (im Beispiel SCH_01, SCH_02...SCH_31). Andernfalls sind die Felder wie im Band einzeln zuzuordnen.
Sektion:	SQL Abfrage für ein InfoField
Name:	[<fieldname>.SQL] Sektion zur Ermittlung eines InfoFields mit Hilfe einer separaten SQL-Abfrage.
Beispiel:	[lastname.SQL] Select lastname from ...
Variable:	
Werte:	

Sektion:	Binding Definition
Name:	[<bindingname>] Sektion zur Definition eines bindings. Sie besteht aus einem Ausdruck der Expression (vom Typ Boolean) und den Action Variablen ActionsTrue und ActionsFalse.
Beispiel:	[LiveOrder] expression=IfThen(GetV(order)<>Null(), True, False) ActionsTrue=LiveOrderTrue ActionsFalse=LiveOrderFalse
Variable:	expression
Werte:	{IfThen(<term>, <>truevalue>, <>falsevalue>) <sectionname>} Die Variable "expression" enthält einen auszuwertenden Ausdruck oder verweist auf eigene Sektion, die diesen Ausdruck enthält. Das Ergebnis der Auswertung bestimmt welche Action ausgeführt werden soll. Die Funktion "IfThen" prüft die Bedingung im Ausdruck. Ist die Bedingung wahr, wird <>truevalue> bzw. bei falsch <>falsevalue> zurückgegeben. <>truevalue> und <>falsevalue> unterliegen keiner Datentypenbeschränkung. Hier sind Boolean, Numerische- oder String-Daten erlaubt. Allerdings müssen <>truevalue> und <>falsevalue> vom gleichen Datentyp sein. Verschachtelte IfThen Ausdrücke sind zulässig. Welche Operationen, Funktionen und Konstanten verwendet werden können ist unter Infos im Dokument "bindings" beschrieben. Zu beachten ist deren case-sensitive Schreibweise!
Variable:	ActionsTrue
Werte:	{ <sectionname> action_1 action_2 ... action_n} Die Variable "ActionsTrue" verweist auf eine Sektion, in der die Aktionen definiert sind, die ausgeführt werden sollen, falls das Ergebnis der "expression" wahr ist. Es ist ebenfalls möglich diese Aktionen als Liste direkt der Variablen zuzuordnen. Welche Operationen, Funktionen und Konstanten bei der Definition von Aktionen verwendet werden können ist unter Infos im Dokument "bindings" beschrieben. Zu beachten ist deren case-sensitive Schreibweise!
Variable:	ActionsFalse
Werte:	{ <sectionname> action_1 action_2 ... action_n} Die Variable "ActionsFalse" verweist auf eine Sektion, in der die Aktionen definiert sind, die ausgeführt werden sollen, falls das Ergebnis der "expression" falsch ist. Es ist ebenfalls möglich diese Aktionen als Liste direkt der Variablen zuzuordnen. Welche Operationen, Funktionen und Konstanten bei der Definition von Aktionen verwendet werden können ist unter Infos im Dokument "bindings" beschrieben. Zu beachten ist deren case-sensitive Schreibweise!
Sektion:	Binding Ausdruck
Name:	[<bindingexpressionname>] Sektion zur Definition eines komplexen binding Ausdrucks.
Beispiel:	[exprsection] IfThen(..., IfThen(..., ...) ...)
Variable:	
Werte:	

Sektion:	Binding Aktion
Name:	[<bindingactionname> Sektion zur Definition einer Binding Aktion. Es können beliebig viele Aktionen definiert werden. Aktionen ermöglichen die Ausgabe von Nachrichten inkl. der Anforderung einer Antwort vom User. Es können Parameterwerte, Werte oder Eigenschaften von PAM Feldern abgefragt und verändert werden. Es existieren allerdings Einschränkungen bei "fselect Feldern" oder PAM Felder die per Referenzselektion ermittelt werden.
Beispiel:	[OrderTrue] SetP(order,'Enabled',False)
Variable:	
Werte:	
Objekt:	FastReports der Anwendungsrolle
Ort:	rootpath\<ApplicationName>\ROLES\<rolename>\REPORTS Im Reports-Verzeichnis sind alle FastReports der Rolle abgelegt. Sie werden in der Regel mit dem ReportDesigner erstellt und in Dateien vom Typ ".fr3" im XML-Format gespeichert.
Sektion:	
Name:	
Beispiel:	
Variable:	
Werte:	
Variable:	
Werte:	
Objekt:	cyAgent-Aufruf per ini Datei
Ort:	<InstallationPath>\cyAgent\<filename>.ini Der einfachste und schnellste Weg eine bestimmte Anwendung über den cyAgent zu starten bildet eine Verknüpfung mit der Datei "cyAgent.exe". Dabei wird der Name der ini-Datei (ohne Typ) als Parameter im Zieleintrag der Verknüpfung, durch eine Leerstelle von "cyAgent.exe" getrennt, übergeben. Befindet sich die ini-Datei in einem anderen Verzeichnis muss der relative Pfad dem Dateinamen vorangestellt werden (z.B. cyAgent.exe ./IniFiles/<filename>). Die ini-Datei enthält die für den Aufruf der gewünschten Applikation erforderlichen Informationen. Ohne ini-Datei startet der cyAgent-Browser und stellt nach Eingabe der cyProvider-URL die gewünschte Verbindung her.

Sektion:	Initialisierung cyAgent
Name:	[Init] Diese Sektion enthält alle für den Aufruf einer cyProvider-Anwendung benötigten Parameter.
Beispiel:	[init] LogonServer=https://myHost.myDomain/cyService Application=cyDoku User=Guest DefaultLang=deu ProfileDir=c:\temp ;ShowCDDInfo=True ;TraceFile= ;ReportFile= ;StackTrace=C:\temp\StackTrace.log ;ProxyPort= ;ProxyServer= ;ProxyUserName= ;ProxyPassword=
Variable:	LogonServer
Werte:	<url cyprovider>/cyService Die Variable "LogonServer" enthält die URL des cyProviders, an dem die Anmeldung erfolgen soll.
Variable:	Application
Werte:	<applicationname> Die Variable bezeichnet die Anwendung, die der cyAgent starten soll.
Variable:	User
Werte:	{<username> Guest} Initialisierung des Anmeldeforms mit dem Benutzernamen. Dabei bewirkt der Eintrag "User=Guest" einen Autologon. Der Gastbenutzer benötigt kein Passwort und bekommt automatisch die Benutzerrolle "Guest" zugewiesen.
Variable:	DefaultLang
Werte:	<LanguageID> Die Variable bestimmt durch Angabe der LanguageID die Spracheinstellung beim Start der Anwendung. Die LanguageID entspricht den Sprachkürzeln nach ISO 639-2 .
Variable:	AllowStore
Werte:	[True False] Die Variable bestimmt ob ein lokales Storefile mit den persönlichen Einstellungen des Anwenders vom cyAgent erstellt werden darf.
Variable:	ProfileDir
Werte:	<pathname> Die Variable bestimmt das Verzeichnis, in das die beiden Storefiles (" <applicationname>_StoreFile.cdd ", " <applicationname>_StyleFile.cdd ") gespeichert werden sollen. Wird der Variablen kein Verzeichnis zugewiesen, ist "Profiledir" per Default das Verzeichnis der Environment-Variablen "USERPROFILE" mit dem Unterverzeichnis "\cysystem".
Variable:	ShowCDDInfo
Werte:	{True False} Die Variable ermöglicht in der Entwicklungsphase einer Anwendung die Anzeige von CDD-Datei Infos in der Statuszeile. Es werden Zusatzinformationen wie z.B. Feldnamen, SAM-/PAM-Namen und Name der CDD-Datei angezeigt, wenn mit der Maus auf ein Feld im PAM oder auf die Gruppen-Fußzeile im SAM positioniert wird.
Variable:	TraceFile
Werte:	<pathname>\<filename> Die Variable bietet dem Entwickler die Möglichkeit ein Tracefile vom cyAgent erstellen zu lassen. Bei fehlender oder leerer TraceFile-Variablen wird keine TraceFile-Datei erstellt.
Variable:	ReportFile
Werte:	<pathname>\<filename> Die Variable bietet dem Entwickler die Möglichkeit ein Tracefile zum Testen von FastReports erstellen zu lassen. Bei fehlender oder leerer ReportFile-Variablen wird keine ReportFile-Datei erstellt.

Variable: StackTrace
Werte: <pathname>\<filename> Der StackTrace erstellt einen Report bei Speicherschutzverletzungen. Dieser Report ist hilfreich bei der Fehlersuche. Im cyAgent-Verzeichnis muss die MAP-Datei des cyAgents vorhanden sein. Ist die Stacktrace Datei nicht definiert, wird kein Report erstellt.
Variable: ProxyPort
Werte: <portnumber> Bei Verwendung eines Proxy-Servers wird mit dieser Variablen der zu verwendende Port festgelegt.
Variable: ProxyServer
Werte: <proxy_url> Bei Verwendung eines Proxy-Servers wird mit dieser Variablen die IP-Adresse oder der DNS-Name des Proxy-Servers festgelegt.
Variable: ProxyUserName
Werte: <username> Bei Verwendung eines Proxy-Servers kann mit dieser Variablen zusätzlich ein spezieller Proxy-User übergeben werden.
Variable: ProxyPassword
Werte: <password> Bei Verwendung eines Proxy-Servers kann mit dieser Variablen zusätzlich ein Passwort des Proxy-User übergeben werden.
Variable: UseSkins
Werte: {false true} Die Variable ermöglicht dem cyAgent die Benutzung von Skins, falls Skins im cyProvider Verzeichnis \Globals\Skins zur Verfügung stehen.
Variable: InitialSkin
Werte: <skinname> Die Variable bietet die Möglichkeit im Falle von "UseSkins=true" einen initialen Skin vorzugeben. Das gilt nur für den Fall, dass der Benutzer keinen eigenen Skin eingestellt hat.
Variable: NoMonitorSwitch
Werte: {false true} Beim Einsatz von zwei Monitoren erstreckt sich das Formular beim Maximieren über beide Monitore, wenn die Variable "NoMonitorSwitch" auf "true" gesetzt ist.