

CASHDATA - Documentazione Tecnica Completa

Indice

1. [Panoramica del Sistema](#)
2. [Architettura e Struttura del Progetto](#)
3. [Configurazione e Nuova Installazione](#)
4. [Funzionalità Principali](#)
5. [Gestione Vendite](#)
6. [Chiusura Cassa](#)
7. [Inventario e Giacenze](#)
8. [Anagrafica Articoli](#)
9. [Gestione Operatori e Permessi](#)
10. [Sistema di Reportistica](#)
11. [Integrazione con ERP](#)
12. [Schema Database](#)
13. [Troubleshooting](#)

1. Panoramica del Sistema

1.1 Descrizione

CASHDATA è un **sistema POS (Point of Sale)** desktop professionale sviluppato in C# con Windows Forms, destinato alla gestione di casse registratrici per il settore retail, in particolare per il fashion (abbigliamento con gestione taglie/colori).

1.2 Tecnologie Utilizzate

Componente	Tecnologia
Framework	.NET Framework 4.7.2
UI	Windows Forms
Database	SQL Server (via SQLOLEDB.1)
ORM	Entity Framework
Reporting	ComponentOne C1Report
Excel	SpreadsheetGear 2017
JSON	Newtonsoft.Json 13.0.3
UI Avanzata	Krypton Ribbon

1.3 Componenti della Solution

```
CASHDATA.sln
├── CASHDATA (Applicazione principale - EXE)
│   ├── Program.cs (Entry point)
│   ├── Form1.cs (Form test)
│   ├── Login.cs (Form login)
│   └── Template XML per report
└── LIBAYCASH (Libreria principale - DLL)
    ├── Home.cs (Form principale)
    ├── Config.cs (Configurazione globale)
    ├── Utility.cs (Funzioni utility)
    ├── 143+ file C# (logica applicativa)
    └── Dipendenze: LIBAYCMN
```

2. Architettura e Struttura del Progetto

2.1 Struttura Directory

```
D:\dev.net\CashData\
├── CASHDATA\
│   ├── # Progetto principale
│   ├── bin\Debug\
│   │   └── # Output compilato
│   ├── Resources\
│   │   └── # Icone e immagini
│   ├── *.xml
│   │   └── # Template report
│   └── App.config
│       └── # Configurazione app
├── LIBAYCASH\
│   ├── # Libreria logica
│   ├── Properties\
│   │   └── Settings.settings
│   │       └── # Impostazioni DB
│   ├── Home.cs
│   │   └── # Form principale
│   ├── Config.cs
│   │   └── # Configurazione
│   └── [143+ file .cs]
│       └── # Classi logica
├── CreateNewDb\
│   └── # Utility creazione DB
├── Script_DB\
│   ├── # Script SQL
│   ├── per C#\
│   │   └── # Script TCS
│   └── TTCS_ALL_INSTALLATION\
└── RELEASE\
    └── # File di release
```

2.2 Entry Point dell'Applicazione

```
// Program.cs
[STAThread]
static void Main()
{
    Application.EnableVisualStyles();
    Application.SetCompatibleTextRenderingDefault(false);
    string assemblyVersion = Assembly.GetExecutingAssembly()
                                   .GetName().Version.ToString();
    Application.Run(new LIBAYCASH.Home(assemblyVersion));
}
```

2.3 Parametri da Riga di Comando

```
CASHDATA.exe [UserShowLevel] [SkipLogTableCheck]
```

- **UserShowLevel:** Livello visualizzazione interfaccia (default: dalla config)
- **SkipLogTableCheck:** Se presente, salta il controllo tabella LOG

3. Configurazione e Nuova Installazione

3.1 Checklist Nuova Installazione

FASE 1: Preparazione Database

- Installare SQL Server (Express o Standard)
- Creare database vuoto (nome consigliato: TTCS)
- Configurare account SA con password

FASE 2: Esecuzione Script SQL

Eseguire in ordine:

1. 00_DB_CREATION_CREATEDB.sql
2. 01_DB_CREATION_PROCEDURE.sql
3. 02_DB_CREATION_TABLES.sql
4. 03_DB_CREATION_WITH_PROMO.sql

FASE 3: Configurazione Iniziale

Modificare ed eseguire 02 Reimposta dati.sql:

```
DECLARE @firma INT = 10; -- Codice azienda
DECLARE @currentshop VARCHAR(6) = '000002'; -- Codice negozio
DECLARE @shopnr INT = 84; -- Numero univoco negozio
DECLARE @releasefor INT = 84; -- Numero PC (= shopnr per
singola cassa)
DECLARE @ivadefault VARCHAR(5) = '22'; -- Aliquota IVA default
DECLARE @cashclosuredate VARCHAR(8) = '20240201'; -- Data prima chiusura
```

FASE 4: Primo Avvio Applicazione

Al primo avvio, inserire:

- **Server DB:** SERVER\SQLEXPRESS o IP
- **Database:** TTCS
- **ReleaseFor:** Numero univoco PC
- **PCisSqlServer:** false (se server remoto)

3.2 Parametri di Configurazione Essenziali

Parametri in CASSA_CONFIG_PARAMETER

Parametro	Valore Esempio	Descrizione
FIRMA_SHOP	10	Codice azienda
CURRENT_SHOP	000002	Codice negozio DIAMOD
VAT_DEFAULT	22	Aliquota IVA %
VAT_BUONO	22	IVA per buoni
BARCODELEN	13	Lunghezza EAN
LanguageCode	0	0=Italiano, 1=Tedesco
DecimalSeparator	,	Separatore decimali
GroupSeparator	.	Separatore migliaia
CurrencySymbol	EUR	Simbolo valuta

Parametri in Properties/Settings.settings

```
<Setting Name="cnn_server" Type="System.String">SERVER\SQLEXPRESS</Setting>
<Setting Name="cnn_database" Type="System.String">TTCS</Setting>
<Setting Name="cnn_provider" Type="System.String">SQLOLEDB.1</Setting>
```

```
<Setting Name="ReleaseFor" Type="System.Int32">84</Setting>
<Setting Name="PCisSqlServer" Type="System.Boolean">False</Setting>
```

3.3 Connection String

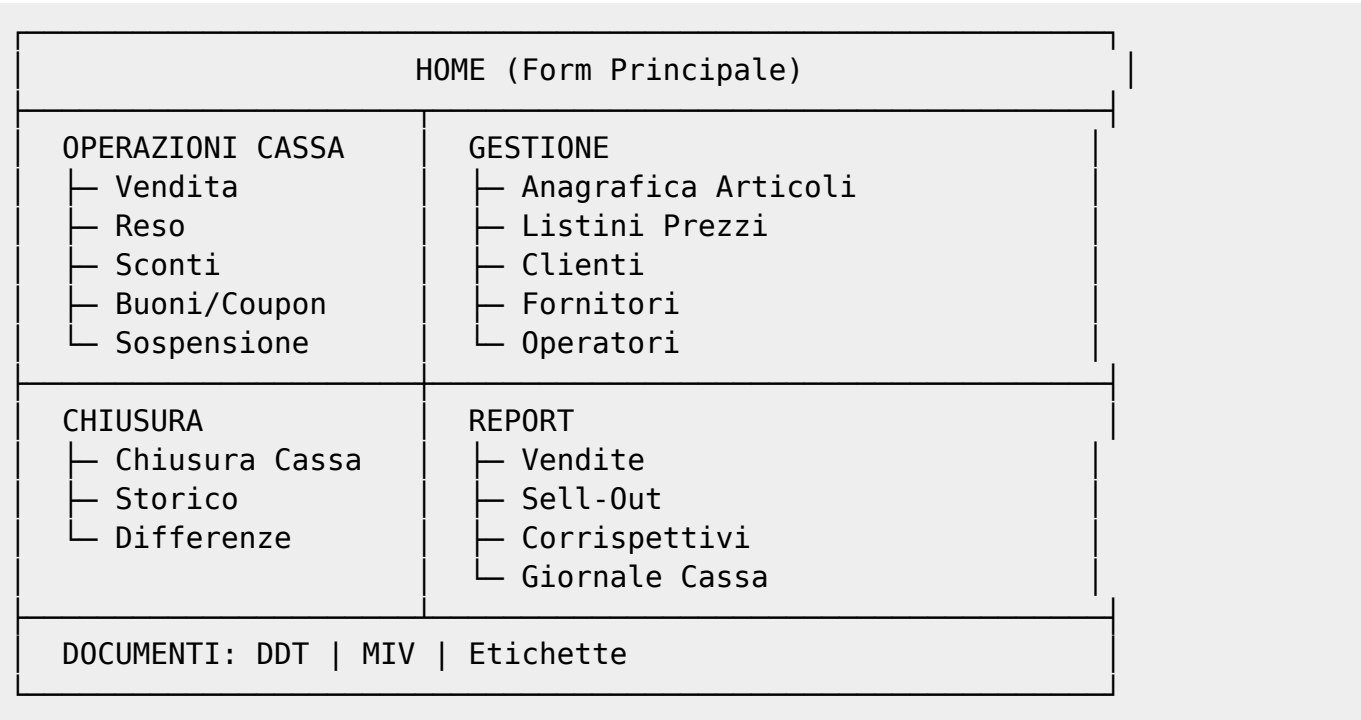
Provider=SQLOLEDB.1;Data Source=SERVER\SQLEXPRESS;Initial Catalog=TTCS;User ID=sa;Password=xxx;

3.4 Configurazione Moduli

Modulo	Parametro	Descrizione
Fornitori	MODULE_SUPPLIER	Gestione fornitori
Clienti	MODULE_CUSTOMER	Anagrafica clienti e fedeltà
Venditori	MODULE_SELLER	Tracciamento venditori
Multi-negozio	MODULE_MULTISHOP	Gestione multi-punto vendita

4. Funzionalità Principali

4.1 Mappa Funzionalità



5. Gestione Vendite

5.1 Flusso di Vendita

1. SCANSIONE ARTICOLO (EAN)
 - ↳ ArticleCheck() verifica articolo e prezzo
2. AGGIUNTA AL CARRELLO
 - ↳ addArticoloCarrello() crea record SCONTRINO
3. MODIFICA (opzionale)
 - ↳ Quantità: btnTastierinoSpec1_Click()
 - ↳ Prezzo: LogicaPrezzo()
 - ↳ Sconto %: LogicaPercentuale()
4. PAGAMENTO
 - ↳ btnPaga_Click() apre dialog PAY
 - ↳ Selezione metodo pagamento
 - ↳ Stampa scontrino fiscale
 - ↳ Registrazione in CASSA_SALES
5. CHIUSURA TRANSAZIONE
 - ↳ Eliminazione record SCONTRINO temporanei

5.2 Metodi di Pagamento

Codice	Costante	Descrizione
1	PAYMANT_MONEY_CODE	Contanti
2	PAYMANT_CARD_CODE	Carta di credito
3	PAYMANT_BANCOMAT_CODE	Bancomat
4	PAYMANT_CHEQUE_CODE	Assegni
5	PAYMANT_COUPON_CODE	Buoni/Coupon
9	PAYMANT_DISCOUNT_CODE	Sconto

5.3 Gestione Sconti

Causale	Codice	Descrizione
PR	SpecialOffer	Offerta speciale
SA	Sales	Svendita
FS	EndSeries	Fine serie
MF	BrokenArticle	Articolo difettoso
VP	Staff	Sconto personale
AR	RoundingPrice	Arrotondamento
CL	Customer	Sconto cliente

Causale	Codice	Descrizione
AL	OtherReason	Altro motivo

5.4 Tipi di Sconto

```
public enum Discount_Type
{
    Inserted,          // Sconto diretto
    Progressive,       // Cumula con sconti precedenti
    Sum                // Somma percentuali
}
```

5.5 Gestione Resi

- Stato articolo: Config.STATO_RESO = "R"
- Quantità diventa negativa (-1)
- Se AskReturnCause = true: richiede causale reso
- Tabella causali: CASSA_RETURN_CAUSE

5.6 Sistema Attese (Sospensione)

- Massimo 4 transazioni contemporanee
- Ogni attesa ha colore distinto
- Struct ATTESA memorizza stato:

```
struct ATTESA {
    List<DataGridViewRow> dgSpesa;
    CASSA_CUSTOMER cliente;
    int pezziTot;
    double toto;
    int punti;
}
```

6. Chiusura Cassa

6.1 Flusso Chiusura

TAB 0: CONTEGGIO DENARO
└─ Inserimento per taglio (€50, €20, €10, €5, €2, €1, cent)

- └ Calcolo totale inserito
- └ Verifica differenza con sistema

TAB 1: VERIFICA PAGAMENTI

- └ Visualizzazione totali per metodo
- └ Inserimento importi verificati
- └ Spunta verifica completata
- └ Differenza totale deve essere = 0

TAB 2: CONFERMA FINALE

- └ Prelievo contante finale
- └ Calcolo nuovo fondo cassa
- └ Conferma chiusura

PROCESSO CHIUSURA (Utility.ChiusuraCassa)

- └ Comunicazione registratore telematico (rapporto Z)
- └ Stampa rapporto chiusura
- └ TRANSAZIONE DATABASE:
 - └ UPDATE CASSA_FUND (nuovo fondo)
 - └ UPDATE CASSA_SALES (stato 0→10)
 - └ UPDATE CASSA_SALES_PAYMODE (stato 0→10)
 - └ INSERT CASSA_DAILY_CLOSURE
 - └ EXEC PRC_SALES_CLOSURE_IVA
 - └ CloseCashSession()
- └ Stampa rapporto differenze (opzionale)
- └ Esportazione dati (se configurato)

6.2 Stati delle Vendite

```
public enum SaleStatus
{
    NewSales = 0,           // Non ancora chiuso
    SalesSessionClosed = 10, // Chiuso in sessione
    Exported = 20,          // Esportato
    SentToCentral = 35,     // Inviato a centrale
    AcceptedFromCentral = 90 // Accettato
}
```

6.3 Tabelle Coinvolte

Tabella	Descrizione
CASSA_DAILY_CLOSURE	Storico chiusure giornaliere
CASSA_FUND	Fondo cassa attuale
CASSA_FUND_PROTOCOL	Movimenti fondo
CASSA_SALES	Vendite registrate

Tabella	Descrizione
CASSA_SALES_PAYMODE	Dettaglio pagamenti

6.4 Codici Speciali in CASSA_DAILY_CLOSURE

CDC_PAYTYPECODE	Significato
-1	Differenza cassa
-2	Fondo cassa
1-N	Codici metodo pagamento

7. Inventario e Giacenze

7.1 Tabelle di Gestione

Tabella	Descrizione
AY_CASSA_VENDITE	Vendite registrate
AY_CASSA_VENDITE_ECCESSO	Vendite oltre stock
CASSA_STOCK	Giacenze per articolo
CASSA_TMP_STOCK_GRID_*	Tabelle temporanee

7.2 Calcolo Giacenza

$$\text{Giacenza Finale} = \text{Giacenza Iniziale} - \text{Quantità Venduta}$$
$$\text{Sell Out \%} = (\text{Quantità Venduta} / \text{Quantità Iniziale}) \times 100$$

7.3 Supporto Multi-Magazzino

- Campo CODMAG in tutte le tabelle movimenti
- Parametro MODULE_MULTISHOP abilita gestione
- CURRENT_SHOP identifica magazzino corrente

7.4 Report Inventario

Report	Stored Procedure	Descrizione
SELLOUT	PRC_CASSA_SELLOUT_V001	Analisi sell-out
SALESREPORT	PRC_CASSA_SALESREPORT_V001	Dettaglio vendite
DAILYDATA	PRC_CASSA_DAILYDATA_V001	Corrispettivi

8. Anagrafica Articoli

8.1 Struttura CASSA_ARTICOLO

Campo	Tipo	Descrizione
ART_FIRMA	INT	Codice azienda
ART_EAN	NVARCHAR(20)	Codice a barre
ART_CODART	NVARCHAR(50)	Codice articolo
ART_NAME_DESC	NVARCHAR(MAX)	Descrizione
ART_COL_DESC	NVARCHAR(50)	Descrizione colore
ART_CODCOLOR	NVARCHAR(20)	Codice colore
ART_SIZE	NVARCHAR(20)	Taglia
ART_SCALA	INT	Scala taglie
ART_VARIANTE	INT	Codice variante
ART_VATPERCENT	DECIMAL	Aliquota IVA

8.2 Gestione Attributi

Tabella	Descrizione
CASSA_ART_ATTRIB	Attributi per articolo
CASSA_ART_ATTRIB_DESC	Descrizione attributi

Campi dinamici: CAA_ATTR_001, CAA_ATTR_002, ... CAA_ATTR_999

8.3 Sistema Listini

Tabella	Descrizione
CASSA_LISTINO_NAME	Anagrafica listini
CASSA_LISTINO	Prezzi per articolo

Campi Listino:

- CPLN_NRLIS: Numero listino
- CPLN_SAISON: Stagione
- CPLN_VALIDFROM/TO: Date validità
- CPL_ENDUSERPRICE: Prezzo al pubblico
- CPL_ORIGINALPRICE: Prezzo originale
- CPL_PRICE: Prezzo costo

8.4 Scale Taglie

Tabella	Descrizione
GROESSENSKALA	Anagrafica scale
GROESSEN	Valori taglie

Esempio scala:

```
Scala_Nr=1, Scala_Bezeichnung="EU Size"  
├ Index=1: "XS"  
├ Index=2: "S"  
├ Index=3: "M"  
├ Index=4: "L"  
└ Index=5: "XL"
```

8.5 Ricerca Articoli

Query base:

```
SELECT ART_NAME_DESC, ART_COL_DESC, ART_SIZE, ART_EAN, ART_CODART  
FROM CASSA_ARTICOLO  
WHERE ART_NAME_DESC LIKE '%' + @ARTNAME + '%'  
      AND ART_EAN LIKE '%' + @ARTEAN + '%'  
      AND ART_FIRMA = @FIRMA  
ORDER BY ART_NAME_DESC, ART_COL_DESC, ART_SIZE
```

9. Gestione Operatori e Permessi

9.1 Tabelle Utenti

Tabella	Descrizione
CASSA_USER	Anagrafica utenti
CASSA_USER_GROUP	Gruppi utenti
CASSA_USER_FUNCTIONGROUP	Funzioni per gruppo
CASSA_USER_OP	Log accessi

9.2 Struttura CASSA_USER

Campo	Descrizione
USR_FIRMA	Codice azienda
USR_CODE	Codice utente (PK)
USR_DESCR	Nome utente

Campo	Descrizione
USR_PWD	Password crittografata (TripleDES)
USR_GROUP_CODE	Gruppo appartenenza
USR_CHECK	Hash integrità (MD5)
USR_DISABLE	Flag disabilitazione

9.3 Crittografia Password

- **Algoritmo:** TripleDES
- **Chiave:** MD5 di "castelletto28053"
- **Modalità:** ECB
- **Padding:** PKCS7
- **Output:** Base64

9.4 Verifica Permessi

```
bool hasPermission = Utility.FunctionForUser(userCode, functionCode);
```

Se utente non ha permesso, viene richiesto cambio operatore.

9.5 Funzioni Controllate

Funzione	Descrizione
0	Accesso sistema
39	Gestione posizione/reso
45	Pagamento
49	Aggiunta articolo
50	Reso

9.6 Tracciamento su Scontrino

- Campo Venditore in tabella SCONTRINO
- Trasferito a CSL_Seller in CASSA_SALES
- Report operatori: PRC_CASSA_USERREPORT_V001

10. Sistema di Reportistica

10.1 Template Report (XML)

File	Tipo	Descrizione
BILL.xml	Scontrino	Scontrino fiscale completo
BILL-NO-PRICES.xml	Scontrino	Scontrino senza prezzi
BILL_CLOSURE.xml	Chiusura	Riepilogo chiusura giornaliera
BILL_CLOSURE_DIFF.xml	Chiusura	Rapporto differenze
SALESREPORT.xml	Report	Dettaglio vendite
SELLOUT.xml	Report	Analisi sell-out
USERREPORT.xml	Report	Vendite per operatore
DAILYDATA.xml	Report	Registro corrispettivi
CASHPROTOCOL.xml	Report	Giornale di cassa
DDT_0.xml	Documento	Documento di trasporto
MIV_0.xml	Documento	Merce in visione
LABEL_STD.xml	Etichette	Etichette articoli
OPEN_DRAWER.xml	Cassa	Apertura cassetto

10.2 Stored Procedure Report

SP	Report
dbo.BILL_PRINT	Scontrino
dbo.BILL_CASH_CLOSURE_PRINT	Chiusura cassa
dbo.PRC_CASSA_SALESREPORT_V001	Vendite
dbo.PRC_CASSA_SELLOUT_V001	Sell-out
dbo.PRC_CASSA_USERREPORT_V001	Operatori
dbo.PRC_CASSA_DAILYDATA_V001	Corrispettivi
dbo.PRC_CASSA_PROTOCOL_V001	Giornale

10.3 Configurazione Stampanti

Tipo Documento	Stampante Consigliata
Scontrini	Stampante termica POS
DDT/Documenti	Stampante laser/inkjet
Etichette	Stampante etichette (Zebra, ecc.)

10.4 File Logo

```
D:\dev.net\CashData\CASHDATA\
├─ BILLPic1.png          # Logo scontrino
├─ BILL_CLOSUREPic1.png  # Logo chiusura
├─ DDTPic1.jpg, DDTPic2.jpg # Logo DDT
├─ MIVPic1.jpg, MIVPic2.jpg # Logo MIV
```

11. Integrazione con ERP

Questa sezione descrive come configurare l'integrazione tra CASHDATA e un sistema ERP esterno per l'esportazione automatica dei dati di chiusura cassa e DDT.

11.1 Panoramica

L'integrazione ERP permette di:

- Esportare automaticamente i dati delle chiusure cassa (vendite, pagamenti, IVA)
- Esportare i DDT confermati verso il gestionale
- Sincronizzare lo stato dei DDT tra i due sistemi

Flusso di esportazione:

CHIUSURA CASSA COMPLETATA

↓
ExchangeData()

↓
PRC_EXPORT_ERP (Stored Procedure)

→ ERP_CASSA_CHIUSURE (totali per metodo pagamento)
→ ERP_CASSA_VENDITE (dettaglio articoli venduti)
→ ERP_CASSA_CHIUSURE_IVA (ripartizione IVA)
→ ERP_CASSA_ELENCO_CHIUSURE (registro chiusure)
→ [Se Invio_DDT_to_ERP = 1]
 → ORDINI_TOCOMPANY (testate DDT)
 → ORDINI_DET_TOCOMPANY (righe DDT)

11.2 Configurazione nel Database CASHDATA

STEP 1: Abilitare l'export ERP

Configurare i seguenti parametri in CASSA_CONFIG_PARAMETER:

Parametro	Valore	Descrizione
EXPORT_ERP	true	Abilita l'export verso ERP

Parametro	Valore	Descrizione
ExportDataForTrasmission	0	0 = export alla chiusura cassa
Invio_DDT_to_ERP	1	1 = esporta anche i DDT (opzionale)

STEP 2: Configurare il codice magazzino ERP

Inserire il parametro ERP_CODMAG che identifica il magazzino/negozio nel sistema ERP:

```
-- Modificare i valori secondo l'installazione specifica
DECLARE @firma INT = 10;
DECLARE @shopnr INT = 84;
DECLARE @releasefor INT = 84;
DECLARE @codmag VARCHAR(50) = 'NEGCU'; -- Codice magazzino nel sistema ERP

IF NOT EXISTS (SELECT 1 FROM CASSA_CONFIG_PARAMETER WHERE CP_NAME =
'ERP_CODMAG' AND CP_FIRMA = @firma)
BEGIN
    INSERT INTO CASSA_CONFIG_PARAMETER
    (SHOP_NR, RELEASEFOR, CP_SECTION, CP_SECTION_PROGR, CP_FIRMA,
CP_DATATYPE, CP_NAME, CP_VALUE, CP_TEXT)
    VALUES
    (@shopnr, @releasefor, 20, 100, @firma, 'string', 'ERP_CODMAG', @codmag,
'Codice magazzino per export ERP')

    PRINT 'Parametro ERP_CODMAG inserito con valore: ' + @codmag
END
```

STEP 3: Creare/Aggiornare la Stored Procedure PRC_EXPORT_ERP

Nel database CASHDATA, creare o aggiornare la stored procedure:

```
ALTER PROCEDURE [dbo].[PRC_EXPORT_ERP]
AS
BEGIN
    DECLARE @CASSULTCHIU AS INT
    DECLARE @ERPULTCHIU AS INT
    DECLARE @CODMAG AS VARCHAR(50)
    DECLARE @0rdprexix AS VARCHAR(10)

    -- Lettura CODMAG da parametro di configurazione
    SELECT @CODMAG = CP_VALUE
    FROM CASSA_CONFIG_PARAMETER
    WHERE CP_NAME = 'ERP_CODMAG'

    IF @CODMAG IS NULL OR LTRIM(RTRIM(@CODMAG)) = ''
```

```

BEGIN
    RAISERROR('Parametro ERP_CODMAG non configurato. Contattare la
software house.', 16, 1)
    RETURN
END

-- Ordprexix deriva da CODMAG (usato come prefisso per i DDT)
SET @Ordprexix = @CODMAG

-- Determina range chiusure da esportare
SET @ERPULTCHIU = ISNULL((SELECT MAX(IDCHIUUSURA) FROM
[ERP_CASSA_ELENCO_CHIUSURE] WHERE CODMAG = @CODMAG), 0)
SET @CASSULTCHIU = ISNULL((SELECT MAX(IDCHIUUSURA) FROM
V_CASSA_CHIUSURE), 0)

-- Pulizia e inserimento dati chiusure
DELETE FROM ERP_CASSA_CHIUSURE WHERE IDCHIUUSURA > @ERPULTCHIU AND
IDCHIUUSURA <= @CASSULTCHIU AND CODMAG = @CODMAG
INSERT INTO ERP_CASSA_CHIUSURE (IDCHIUUSURA, DATA, CODPAG, VALCHIUUSURA,
CODMAG)
SELECT IDCHIUUSURA, DATASCONTRINO, CODPAG, VALCHIUUSURA, @CODMAG
FROM V_CASSA_CHIUSURE WHERE IDCHIUUSURA > @ERPULTCHIU AND IDCHIUUSURA <=
@CASSULTCHIU

-- ... continua con altre operazioni ...
END

```

11.3 Configurazione nel Database ERP

Nel database del gestionale ERP è necessario creare le tabelle di destinazione per ricevere i dati esportati da CASHDATA.

File script: D:\dev.net\CashData\CASHDATA\ERP2Tabelle.sql

Tabelle per Chiusure Cassa

```

-- Chiusure per metodo di pagamento
CREATE TABLE [dbo].[AY_CASSA_CHIUSURE](
    [ID] [INT] IDENTITY(1,1) NOT NULL,
    [IDCHIUUSURA] [INT] NOT NULL,
    [DATA] [smalldatetime] NULL,
    [CODPAG] [VARCHAR](50) NULL,
    [VALCHIUUSURA] [DECIMAL](12, 2) NULL,
    [CODMAG] [VARCHAR](50) NULL
)

```

```
-- Ripartizione IVA per chiusura
CREATE TABLE [dbo].[AY_CASSA_CHIUSURE_IVA](
    [ID] [INT] IDENTITY(1,1) NOT NULL,
    [IDCHIUSURA] [INT] NOT NULL,
    [Aliquota] [DECIMAL](12, 2) NULL,
    [Tot_senza_Iva] [DECIMAL](12, 2) NULL,
    [Tot_Iva] [DECIMAL](12, 2) NULL,
    [ID_IVA] [INT] NULL,
    [CODMAG] [VARCHAR](50) NULL
)

-- Elenco chiusure esportate
CREATE TABLE [dbo].[AY_CASSA_ELENCO_CHIUSURE](
    [ID] [INT] IDENTITY(1,1) NOT NULL,
    [IDCHIUSURA] [INT] NOT NULL,
    [STATO] [SMALLINT] NOT NULL DEFAULT(0),
    [DATCHIUI] [smalldatetime] NULL,
    [DATCRE] [smalldatetime] NULL,
    [DATMOD] [smalldatetime] NULL,
    [NOTE] [VARCHAR](500) NULL,
    [CODMAG] [VARCHAR](50) NULL
)

-- Anagrafica metodi di pagamento
CREATE TABLE [dbo].[AY_CASSA_PAGAMENTI](
    [CODPAG] [INT] NOT NULL,
    [DESPAG] [VARCHAR](200) NULL,
    [CODCON] [VARCHAR](50) NULL,
    [CODCAUINC] [VARCHAR](50) NULL
)
```

Tabelle per Vendite

```
-- Dettaglio vendite per articolo
CREATE TABLE [dbo].[AY_CASSA_VENDITE](
    [ID] [INT] IDENTITY(1,1) NOT NULL,
    [DATVEN] [smalldatetime] NULL,
    [CODART] [VARCHAR](100) NOT NULL,
    [QUANTITA] [INT] NOT NULL,
    [PREZZOVENDITA] [DECIMAL](12, 2) NOT NULL,
    [IDCHIUSURA] [INT] NOT NULL,
    [CODMAG] [VARCHAR](50) NULL
)

-- Vendite in eccesso rispetto a giacenza teorica
CREATE TABLE [dbo].[AY_CASSA_VENDITE_ECESSO](
    [ID_RIF] [INT] NOT NULL,
```

```
[DATVEN] [smalldatetime] NULL,  
[CODART] [VARCHAR](100) NOT NULL,  
[QUANTITA] [INT] NOT NULL,  
[PREZZOVENDITA] [DECIMAL](12, 2) NOT NULL,  
[IDCHIUSURA] [INT] NOT NULL,  
[CODMAG] [VARCHAR](50) NULL  
)
```

11.4 Checklist Integrazione ERP

Nel Database CASHDATA:

- Inserire parametro EXPORT_ERP = true
- Inserire parametro ExportDataForTrasmission = 0
- Inserire parametro ERP_CODMAG con il codice magazzino corretto
- (Opzionale) Inserire parametro Invio_DDT_to_ERP = 1 se serve export DDT
- Creare/aggiornare la stored procedure PRC_EXPORT_ERP
- Verificare esistenza viste V_CASSA_CHIUSURE e V_CASSA_VENDITE_ARTICOLI

Nel Database ERP:

- Creare tabella AY_CASSA_CHIUSURE
- Creare tabella AY_CASSA_CHIUSURE_IVA
- Creare tabella AY_CASSA_ELENCO_CHIUSURE
- Creare tabella AY_CASSA_PAGAMENTI
- Creare tabella AY_CASSA_VENDITE
- Creare tabella AY_CASSA_VENDITE_ECCESSO
- (Se DDT) Creare tabelle EXT_ORDINI_TOCOMPANY, EXT_ORDINI_DET_TOCOMPANY
- Configurare linked server o replica per comunicazione tra i database

11.5 Stati DDT

Stato	Descrizione	Origine
45	Confermato	CASHDATA
50	Trasmesso	CASHDATA (dopo export)
70	Importato	ERP (elaborazione OK)
80	Non importato	ERP (errore elaborazione)

11.6 Troubleshooting Integrazione ERP

Errore	Causa	Soluzione
"Parametro ERP_CODMAG non configurato..."	Parametro mancante	Inserire ERP_CODMAG in CASSA_CONFIG_PARAMETER
"Errore export dati vendite"	Errore generico SP	Verificare log SQL Server, controllare esistenza tabelle ERP
DDT non esportati	Invio_DDT_to_ERP = 0	Impostare parametro a 1
Dati duplicati in ERP	Chiusure riesportate	Verificare IDCHIUSURA in ERP_CASSA_ELENCO_CHIUSURE

12. Schema Database

12.1 Tabelle Principali

Vendite

CASSA_SALES	# Vendite registrate
CASSA_SALES_PAYMODE	# Dettaglio pagamenti
CASSA_DAILY_CLOSURE	# Chiusure giornaliere
SCONTRINO	# Carrello temporaneo

Articoli

CASSA_ARTICOLO	# Anagrafica articoli
CASSA_LISTINO	# Prezzi
CASSA_LISTINO_NAME	# Anagrafica listini
CASSA_ART_ATTRIB	# Attributi articoli
CASSA_STOCK	# Giacenze

Utenti

CASSA_USER	# Utenti
CASSA_USER_GROUP	# Gruppi
CASSA_USER_FUNCTIONGROUP	# Funzioni
CASSA_USER_OP	# Log accessi

Configurazione

```
CASSA_CONFIG_PARAMETER    # Parametri configurazione
CASSA_CONFIG_SECTION      # Sezioni configurazione
CASSA_PARAMETER_PC        # Parametri per PC
```

Documenti

```
CASSA_DDT                  # Documenti di trasporto
CASSA_MIV                  # Merce in visione
```

12.2 Relazioni Chiave

```
CASSA_USER.USR_GROUP_CODE → CASSA_USER_GROUP.UGR_CODE
CASSA_SALES.CSL_FIRMA → Config.FIRMA
CASSA_ARTICOLO.ART_EAN → CASSA_LISTINO.CPL_BARCODE
CASSA_DAILY_CLOSURE.CDC_CLOSURE_ID → CASSA_SALES.CSL_CLOSURE_ID
```

13. Troubleshooting

13.1 Errori Comuni di Connessione

Errore	Causa	Soluzione
"Errore connessione DB"	Server non raggiungibile	Verificare IP/nome server
"Login failed"	Credenziali errate	Verificare username/password
"Database not found"	DB non esiste	Creare database ed eseguire script

13.2 Errori di Configurazione

Errore	Causa	Soluzione
"Parametro mancante: X"	Parametro non configurato	Inserire in CASSA_CONFIG_PARAMETER
"ReleaseFor non valido"	PC non registrato	Configurare ReleaseFor in Settings

13.3 Errori Operativi

Errore	Causa	Soluzione
"Differenza ≠ 0"	Conteggio errato	Verificare contanti inseriti

Errore	Causa	Soluzione
"Pagamenti non verificati"	Manca spunta	Completare verifica pagamenti
"Nuovo fondo negativo"	Prelievo eccessivo	Ridurre importo prelievo

13.4 Log e Diagnostica

- **Log accessi:** Tabella CASSA_USER_OP
- **Log operazioni:** Consultare con report CASHPROTOCOL
- **Errori applicazione:** Verificare file di log in cartella installazione

Appendice A: Enumerazioni Principali

A.1 Customer (Tipi Cliente)

```
public enum Customer {  
    First_Installation_ITALIAN = 0,  
    First_Installation_GERMAN = 1,  
    Develop = 2,  
    Demo_Server = 3,  
    // ... altri 80+ clienti  
}
```

A.2 ConfigSection

```
public enum ConfigSection {  
    SHOP_10 = 10,        // Configurazione negozio  
    FUNCTION_20 = 20,    // Configurazione funzioni  
    CASH_30 = 30         // Configurazione cassa  
}
```

A.3 ECR_Type_Code (Tipi Registratore)

```
public enum ECR_Type_Code {  
    Olivetti_Nettuna_300,  
    // altri tipi supportati  
}
```

A.4 DocPrinter (Tipi Documento)

```
public enum DocPrinter {  
    SCONTRINO = 1,  
    ETICHETTE = 2,  
    DDT = 3,  
    MIV = 4,  
    DIFF_CASSA = 5,  
    GIORNALE = 6  
}
```

Appendice B: File Importanti

B.1 Sorgenti Principali

File	Righe	Descrizione
Config.cs	~1500	Configurazione globale
Utility.cs	~7500	Funzioni utility
Home.cs	~1000	Form principale
HomeLogic.cs	~3000	Logica form principale
Form1.cs	~2700	Form vendite
PAY.cs	~2500	Dialog pagamento
CashChiusura.cs	~800	Form chiusura cassa
UpdateUtility.cs	~1600	Importazione dati

B.2 Script Database

File	Descrizione
00_DB_CREATION_CREATEDB.sql	Creazione database
01_DB_CREATION_PROCEDURE.sql	Stored procedures
02_DB_CREATION_TABLES.sql	Tabelle
02 Reimposta dati.sql	Configurazione iniziale

Documento generato automaticamente dall'analisi del codice sorgente CASHDATA.

Data generazione: Gennaio 2026

Versione: 1.0

From:

<https://kb.analytic.it/> - **kb.analytic.it**

Permanent link:

<https://kb.analytic.it/cashdata?rev=1768906101>

Last update: **2026/01/20 11:48**

