

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 ERP2](#)
12. [Integrazione con Bilancia](#)
13. [Schema Database](#)
14. [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

Componente	Tecnologia
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\
│   ├── bin\Debug\
│   ├── Resources\
│   ├── *.xml
│   └── App.config
│   # Progetto principale
│   # Output compilato
│   # Icone e immagini
│   # Template report
│   # Configurazione app
├── LIBAYCASH\
│   ├── Properties\
│   │   └── Settings.settings
│   ├── Home.cs
│   ├── Config.cs
│   └── [143+ file .cs]
│   # Libreria logica
│   # Impostazioni DB
│   # Form principale
│   # Configurazione
│   # Classi logica
├── CreateNewDb\
│   # Utility creazione DB
├── Script_DB\
│   ├── per C#\
│   └── TTCS_ALL_INSTALLATION\
│   # Script SQL
│   # Script TCS
└── 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 o TTCS_LOCAL per sviluppo)
- Configurare account SQL (es. sa) con password
- Verificare che l'account SQL abbia permessi sufficienti sul database

FASE 2: Esecuzione Script SQL

Eseguire in ordine dalla cartella RELEASE\01\:

1. 00_DB_CREATION_CREATEDB.sql - Creazione struttura DB
2. 01_DB_CREATION_PROCEDURE.sql - Stored procedures
3. 02_DB_CREATION_TABLES.sql - Tabelle, dati iniziali, utenti e gruppi
4. 03_DB_CREATION_WITH_PROMO.sql - Dati promozioni

Nota: Lo script 02_DB_CREATION_TABLES.sql include automaticamente:

- Parametri di configurazione (CASSA_CONFIG_PARAMETER) per tutte e 3 le sezioni (10, 20, 30)
- Parametri runtime (CASSA_PARAMETER) inclusi RECEIPT_COUNTER_PWD, HOMEMENU_ENABLED
- Parametri per-PC (CASSA_PARAMETER_PC) inclusi NEW_DAYLY_CLOSURE, NEW_START_BILL
- Gruppi utente predefiniti (GSU, ADMINISTRATORS GROUP, MANAGER, USER)
- Utenti predefiniti (su, Administator, Manager, User) con password impostate al primo avvio
- Stampante predefinita "Microsoft Print to PDF" per tutti i tipi documento
- Valori LIC (licenza) vuoti per consentire la prima installazione

FASE 3: Configurazione Iniziale

Modificare ed eseguire Cose da fare per nuova installazione\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
```

Questo script aggiorna automaticamente anche:

- I valori LIC (licenza) vengono azzerati per consentire la prima installazione
- Il parametro ACTIVATION viene impostato a '.'

FASE 4: Compilazione Applicazione

- Aprire la solution CASHDATA.sln in Visual Studio
- Impostare la configurazione su **Release** e piattaforma **x86**
- Compilare la solution
- Copiare il contenuto della cartella di output nella cartella di installazione (es. C:\APP - LOCAL\CASHDATA)

FASE 5: Configurazione CashData.xml

Prima di avviare l'applicazione, configurare il file CashData.xml nella cartella di installazione:

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <appSettings>
```

```

<add key="cnn_server" value="localhost\SQLEXPRESS" />
<add key="cnn_database" value="TTCS_LOCAL" />
<add key="cnn_user" value="sa" />
<add key="cnn_password" value="la_tua_password" />
<add key="ReleaseFor" value="84" />
<add key="PCisSqlServer" value="false" />
</appSettings>
</configuration>

```

Parametri importanti:

Parametro	Descrizione	Note
cnn_server	Indirizzo SQL Server	Es. localhost\SQLEXPRESS, 192.168.1.100
cnn_database	Nome database	Deve corrispondere al DB creato nella FASE 1
cnn_user	Utente SQL	Es. sa
cnn_password	Password SQL	Password dell'utente SQL
ReleaseFor	ID univoco del PC	Deve corrispondere a @releasefor dello script
PCisSqlServer	Il PC e' il SQL Server	false se SQL Server non e' locale o l'utente SQL non ha permessi BACKUP. Impostare a true solo se si vuole il backup automatico del DB

FASE 6: Primo Avvio Applicazione

1. **Avviare CASHDATA.exe** dalla cartella di installazione
2. **Cartella di installazione:** Al primo avvio, l'app chiede di confermare la cartella. Selezionare il percorso dove e' stato copiato l'eseguibile (es. C:\APP - LOCAL\CASHDATA)
3. **Creazione sottocartelle:** L'applicazione crea automaticamente: DDT\, Received\, ToSend\, backup\, basic\, EXCEL\
4. **Inizializzazione utenti:** La funzione ChechForNewFunction() al primo avvio:
 - Popola automaticamente la tabella CASSA_USER_FUNCTIONGROUP con i permessi per ogni gruppo
 - Imposta le password predefinite per gli utenti
5. **Login:** Usare le credenziali predefinite:

Utente	Password	Livello
su	su28053	SuperUser (99)
Administator	admincash	Amministratore (30)
Manager	manager1	Manager (20)
User	user	Utente base (10)

3.2 Architettura dei Parametri

Il sistema CASHDATA utilizza **3 tabelle distinte** per i parametri, ognuna con uno scopo diverso:

3.2.1 CASSA_CONFIG_PARAMETER (Configurazione)

Tabella principale di configurazione a 3 livelli (sezioni):

Sezione	RELEASEDFOR	Scopo	Caricamento
10	-1	Parametri negozio (globali)	Config.Parametri via getParValueByKey()
20	N (=ReleaseFor)	Flag funzioni UI (solo TRUE)	Config.Funzioni
30	N (=ReleaseFor)	Parametri per-PC/hardware	Config.Parametri (sovrascrive sez. 10)

Lettura: ReadConfigParameter(name, releasedfor, section) oppure
getParValueByKey(key) per le sezioni 10+30.

3.2.2 CASSA_PARAMETER (Parametri Runtime Globali)

Parametri operativi condivisi da tutte le casse del negozio (contatori, percorsi, password).

Lettura: ReadParameter(name, increment) con cash_nr = 0 (default).

Parametro	Valore	Descrizione
NEW_CLOSUREID	1	Contatore ID chiusura
NEW_START_BILL	1	Contatore scontrini (se globale)
RECEIPT_COUNTER_PWD	tex01	Password reset contatore scontrini
INVENTORY_PWD	tex01	Password conferma inventario
HOMEMENU_ENABLED	0	Menu home abilitato (0/1)
LAST_TCS	1	Ultimo file TCS eseguito
DDT_FILE	c:\texdatacash\DDT\	Percorso caricamento DDT
SALES_EXCEL_FILE	...	Percorso report vendite Excel
STOCK_EXCEL_FILE	...	Percorso report giacenze Excel

3.2.3 CASSA_PARAMETER_PC (Parametri Runtime per-PC)

Parametri specifici per ogni cassa/PC, identificati dalla colonna CASH_NR.

Lettura: ReadParameter(name, increment, Config.CASHNUMBER) con cash_nr != 0.

Parametro	Valore	Descrizione
NEW_DAYLY_CLOSURE	1	Contatore chiusura giornaliera per cassa
NEW_START_BILL	1	Contatore scontrini per cassa

3.3 Parametri di Configurazione Essenziali

Parametri in CASSA_CONFIG_PARAMETER

Parametro	Sezione	Valore Esempio	Descrizione
FIRMA_SHOP	10	10	Codice azienda
CURRENT_SHOP	10	000002	Codice negozio DIAMOD
VAT_DEFAULT	10	22	Aliquota IVA %
VAT_BUONO	10	22	IVA per buoni
BARCODELEN	10	13	Lunghezza EAN
LanguageCode	10	0	0=Italiano, 1=Tedesco
DecimalSeparator	10	,	Separatore decimali
GroupSeparator	10	.	Separatore migliaia
CurrencySymbol	10	EUR	Simbolo valuta
INSTALLFOLDER	30	C:\APP - LOCAL\CASHDATA	Cartella installazione
LIC_SOSerial	30	(vuoto)	Seriale licenza (vuoto per nuova installazione)
LIC_Volume	30	(vuoto)	Volume licenza (vuoto per nuova installazione)
LIC_ACTIVATE	30	(vuoto)	Stato attivazione (vuoto per nuova installazione)

3.4 Connection String

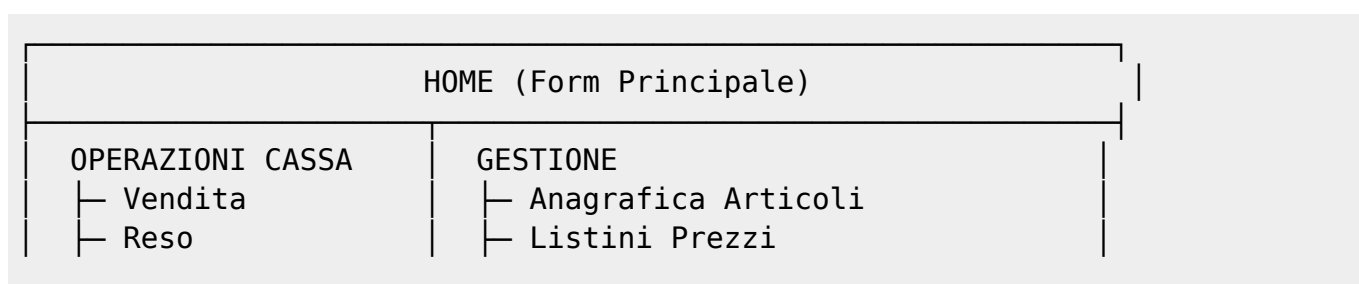
```
Provider=SQLOLEDB.1;Data Source=SERVER\SQLEXPRESS;Initial Catalog=TTCS;User ID=sa;Password=xxx;
```

3.5 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



└ Sconti └ Buoni/Coupon └ Sospensione	└ Clienti └ Fornitori └ Operatori	
CHIUSURA └ Chiusura Cassa └ Storico └ Differenze	REPORT └ Vendite └ Sell-Out └ Corrispettivi └ Giornale Cassa	
DOCUMENTI: DDT MIV Etichette		

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

Codice	Costante	Descrizione
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
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
SendedToCentral = 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
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

Giacenza Finale = Giacenza Iniziale - Quantità Venduta
 Sell Out % = (Quantità Venduta / Quantità Iniziale) × 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/T0: 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
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

Funzione	Descrizione
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 ERP2

Questa sezione descrive come configurare l'integrazione tra CASHDATA e il gestionale **ERP2** per la sincronizzazione bidirezionale di articoli, listini, chiusure cassa e DDT.

Script di riferimento: RELEASE\02 Integrazione con ERP2\

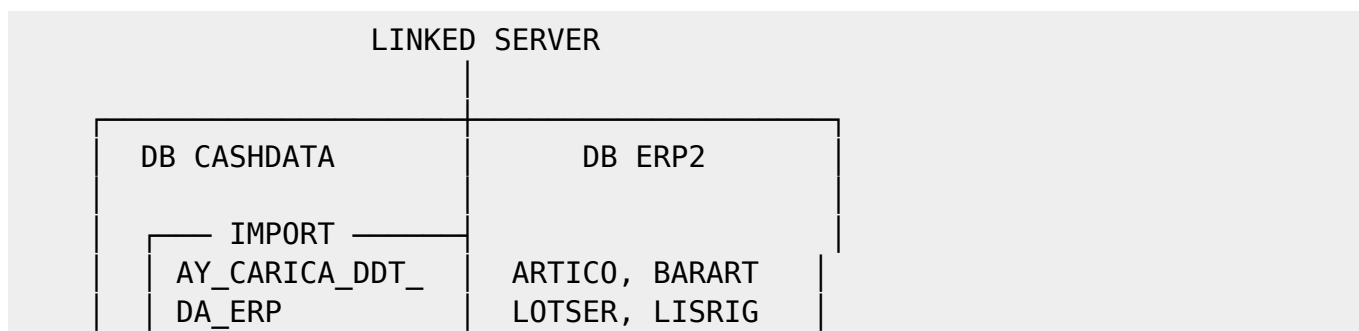
Nota: Le viste e stored procedure possono variare a seconda dell'installazione specifica del cliente. Gli script forniti sono template da adattare.

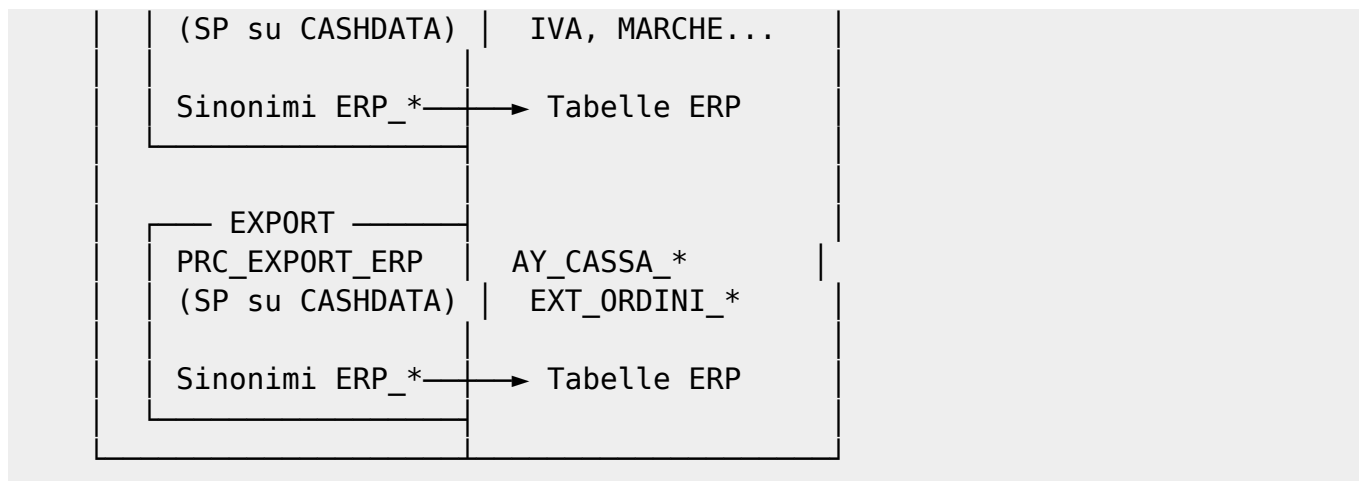
11.1 Panoramica

L'integrazione ERP2 permette di:

- **Import** da ERP: Caricare articoli e listini prezzi dal gestionale alla cassa
- **Export** verso ERP: Esportare automaticamente chiusure cassa, vendite, IVA e DDT
- Sincronizzare lo stato dei DDT tra i due sistemi

Flusso bidirezionale:





11.2 Parametri di Configurazione

Parametri da configurare in CASSA_CONFIG_PARAMETER per abilitare l'integrazione:

Parametro	Sezione	Valore	Descrizione
EXPORT_ERP	10	true	Abilita l'export verso ERP alla chiusura cassa
ExportDataForTransmission	10	0	Modalita export: 0 = alla chiusura cassa, -1 = manuale (mostra bottone), N = ogni N millisecondi
ERP_CODMAG	20	(es. <i>NEGCU</i>)	Obbligatorio. Codice magazzino nel sistema ERP. Usato anche come prefisso per chiavi DDT
Invio_DDT_to_ERP	10	0 o 1	1 = esporta anche i DDT verso ERP (opzionale)
ExportDataType	10	0	Formato dati export: 1 = CSV, 2 = Excel, 0 = nessun file
Carico_Ordini	10	false	Abilita caricamento ordini da ERP

11.3 Procedura di Installazione

Gli script si trovano in RELEASE\02 Integrazione con ERP2\ e vanno eseguiti in ordine:

STEP 0: Creare il Linked Server (su database master)

Script: 00_Creare linked server.txt

Crea un linked server denominato ERPSERVER per connettere SQL Server di CASHDATA a quello del gestionale ERP2.

Valori da personalizzare:

Parametro	Valore di esempio	Descrizione
@datasrc	SERVER\GESTERP	Nome/IP e istanza del SQL Server dove risiede il DB ERP2

Parametro	Valore di esempio	Descrizione
@rmtuser	sa	Utente SQL per accedere al DB ERP2
@rmtpassword	#####	Password dell'utente SQL remoto

```
-- Esempio (da 00_Creare linked server.txt)
EXEC master.dbo.sp_addlinkedserver
    @server = N'ERPSERVER',
    @srvproduct = N'',
    @provider = N'SQLNCLI',
    @datasrc = N'SERVER\GESTERP' -- PERSONALIZZARE

EXEC master.dbo.sp_addlinkedsrvlogin
    @rmtsrvname = N'ERPSERVER',
    @useself = N'False',
    @locallogin = NULL,
    @rmtuser = N'sa', -- PERSONALIZZARE
    @rmtpassword = '#####' -- PERSONALIZZARE
```

Le opzioni importanti impostate dallo script:

- data access = true (necessario per le query cross-server)
- connect timeout = 0 e query timeout = 0 (nessun timeout)
- use remote collation = true

STEP 1: Creare tabelle e viste nel DB ERP2

Script: 01_Tabelle&Viste da creare su db ERP.txt

Eseguire sul **database del gestionale ERP2**. Lo script crea:

1. **Login e utente** texdatacash con permessi di lettura e scrittura
2. **Tabelle per ricevere i dati delle chiusure cassa:**
 - AY_CASSA_ELENCO_CHIUSURE - Registro chiusure esportate
 - AY_CASSA_CHIUSURE - Totali per metodo pagamento
 - AY_CASSA_CHIUSURE_IVA - Ripartizione IVA per chiusura
 - AY_CASSA_PAGAMENTI - Anagrafica metodi di pagamento
 - AY_CASSA_VENDITE - Dettaglio vendite per articolo
 - AY_CASSA_VENDITE_ECESSO - Vendite oltre giacenza teorica
3. **Tabelle per DDT** (se Invio_DDT_to_ERP = 1):
 - EXT_ORDINI_TOCOMPANY - Testate ordini/DDT
 - EXT_ORDINI_DET_TOCOMPANY - Righe ordini/DDT
 - EXT_ORDINI_TOCOMPANY_STOR - Storico DDT
4. **Sinonimi** per accesso semplificato:
 - ORDINI_DET_TOCOMPANY → EXT_ORDINI_DET_TOCOMPANY
 - ORDINI_TOCOMPANY → EXT_ORDINI_TOCOMPANY
 - ORDINI_TOCOMPANY_STOR → EXT_ORDINI_TOCOMPANY_STOR
5. **Vista** V_AY_CASSA_ARTICOLI_LOTTI - Articoli con giacenze per lotto/magazzino

Valori da personalizzare:

Parametro	Valore di esempio	Descrizione
Password texdatacash	qgt... (placeholder)	Impostare una password sicura
ALMG.ID IN (41, 42) nella vista	ID magazzini	Sostituire con gli ID magazzino corretti del gestionale

STEP 2: Creare sinonimi, viste e SP nel DB CASHDATA

Script: 02_Da fare su DB CashData.txt

Eseguire sul **database CASHDATA**. Lo script crea:

a) Sinonimi verso le tabelle ERP (via linked server)

Creare un sinonimo per ogni tabella ERP referenziata. Il pattern e':

```
CREATE SYNONYM [dbo].[ERP_ARTICO] FOR
[ERPSEVER].[NOMEDB_ERP].[dbo].[ARTICO]
```

Sinonimi necessari (sostituire NOMEDB_ERP con il nome effettivo del database ERP2):

Sinonimo nel DB CASHDATA	Tabella nel DB ERP
ERP_ARTICO	ARTICO
ERP_BARART	BARART
ERP_LOTSER	LOTSER
ERP_CLASSI	CLASSI
ERP_GRUPPI	GRUPPI
ERP_IVA	IVA
ERP_LISRIG	LISRIG
ERP_LISTES	LISTES
ERP_MARCHE	MARCHE
ERP_CATOMO	CATOMO
ERP_SCOCLI	SCOCLI
ERP_VALUTA	VALUTA
ERP_BAMTES	BAMTES
ERP_FATRIG	FATRIG
ERP_CASSA_CHIUSURE	AY_CASSA_CHIUSURE
ERP_CASSA_CHIUSURE_IVA	AY_CASSA_CHIUSURE_IVA
ERP_CASSA_ELENCO_CHIUSURE	AY_CASSA_ELENCO_CHIUSURE
ERP_CASSA_VENDITE	AY_CASSA_VENDITE
ORDINI_TOCOMPANY	EXT_ORDINI_TOCOMPANY (o sinonimo)
ORDINI_DET_TOCOMPANY	EXT_ORDINI_DET_TOCOMPANY (o sinonimo)
ORDINI_TOCOMPANY_STOR	EXT_ORDINI_TOCOMPANY_STOR (o sinonimo)

b) Vista V_ERP_ARTICOLO - Unisce articoli ERP con barcode, IVA, lotti e classificazione

c) Vista V_ERP_ARTICOLO_PREZZI - Estende V_ERP_ARTICOLO aggiungendo prezzi da listino, marca e categoria (puo' variare per installazione)

d) Stored Procedure AY_CARICA_DDT_DA_ERP - Importa articoli e aggiorna listini:

- Con parametro @LEGGILISTINI = 0: importa nuovi articoli da ERP in CASSA_READ_DELNOTE e CASSA_READ_DELNOTE_ART_ATTRIB
- Con parametro @LEGGILISTINI = 1: aggiorna prezzi, marca e categoria per articoli esistenti, e registra nuove aliquote IVA
- Puo' essere invocata dal bottone "Carica da ERP" nella sezione Controllo dell'UI, oppure schedulata come SQL Server Job

e) Stored Procedure PRC_EXPORT_ERP - Esporta chiusure cassa verso ERP:

- Esportazione incrementale basata su IDCHIUSURA
- Gestisce chiusure, vendite, IVA
- Se Invio_DDT_to_ERP = 1: esporta anche DDT con stato 45 (confermato) → 50 (trasmesso)

11.4 Checklist Integrazione ERP2

Sul server SQL Server (master):

- Creare linked server ERPSERVER con 00_Creare linked server.txt
- Verificare connettivita con `SELECT * FROM OPENQUERY(ERPSERVER, 'SELECT 1')`

Sul database ERP2:

- Creare login/utente texdatacash con permessi read/write
- Creare le 6 tabelle AY_CASSA_*
- (Se DDT) Creare le 3 tabelle EXT_ORDINI_* e relativi sinonimi
- Creare/modificare la vista V_AY_CASSA_ARTICOLI_LOTTI con ID magazzini corretti
- Popolare AY_CASSA_PAGAMENTI con i codici pagamento corretti

Sul database CASHDATA:

- Creare tutti i sinonimi ERP_* verso le tabelle ERP (personalizzare nome DB)
- Creare la vista V_ERP_ARTICOLO
- Creare la vista V_ERP_ARTICOLO_PREZZI (personalizzare per installazione)
- Creare/aggiornare la SP AY_CARICA_DDT_DA_ERP
- Creare/aggiornare la SP PRC_EXPORT_ERP
- Configurare parametro EXPORT_ERP = true in CASSA_CONFIG_PARAMETER
- Configurare parametro ExportDataForTransmission = 0
- Configurare parametro ERP_CODMAG con il codice magazzino corretto
- (Opzionale) Configurare Invio_DDT_to_ERP = 1 per export DDT
- Verificare esistenza viste V_CASSA_CHIUSURE e V_CASSA_VENDITE_ARTICOLI (create dallo script 01_DB_CREATION_PROCEDURE.sql)

- Testare import articoli: EXEC AY_CARICA_DDT_DA_ERP @LEGGILISTINI = 0
- Testare export chiusura: EXEC PRC_EXPORT_ERP

11.5 Stati DDT

Stato	Descrizione	Origine
45	Confermato	CASHDATA
50	Trasmesso	CASHDATA (dopo export PRC_EXPORT_ERP)
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 (sezione 20)
"Errore export dati vendite"	Errore generico SP	Verificare log SQL Server, controllare esistenza tabelle e sinonimi ERP
Linked server non raggiungibile	Rete o credenziali	Testare con SELECT * FROM OPENQUERY(ERPSEVER, 'SELECT 1')
Errore collation nei JOIN	Collation diversa tra DB	Le viste usano COLLATE Latin1_General_CI_AS per gestire le differenze
DDT non esportati	Invio_DDT_to_ERP = 0	Impostare parametro a 1 in CASSA_CONFIG_PARAMETER
Dati duplicati in ERP	Chiusure riesportate	Verificare IDCHIUSURA in ERP_CASSA_ELENCO_CHIUSURE
Articoli non importati	Barcode già presente in CASSA_READ_DELNOTE_PROTOCOL	L'import è incrementale: salta articoli già importati
Bottone "Esporta Dati" non visibile	ExportDataForTransmission != -1	Per il bottone manuale, impostare ExportDataForTransmission = -1

11b. Integrazione con Bilancia

Questa sezione descrive come configurare l'export del listino prezzi verso bilance elettroniche collegate al sistema.

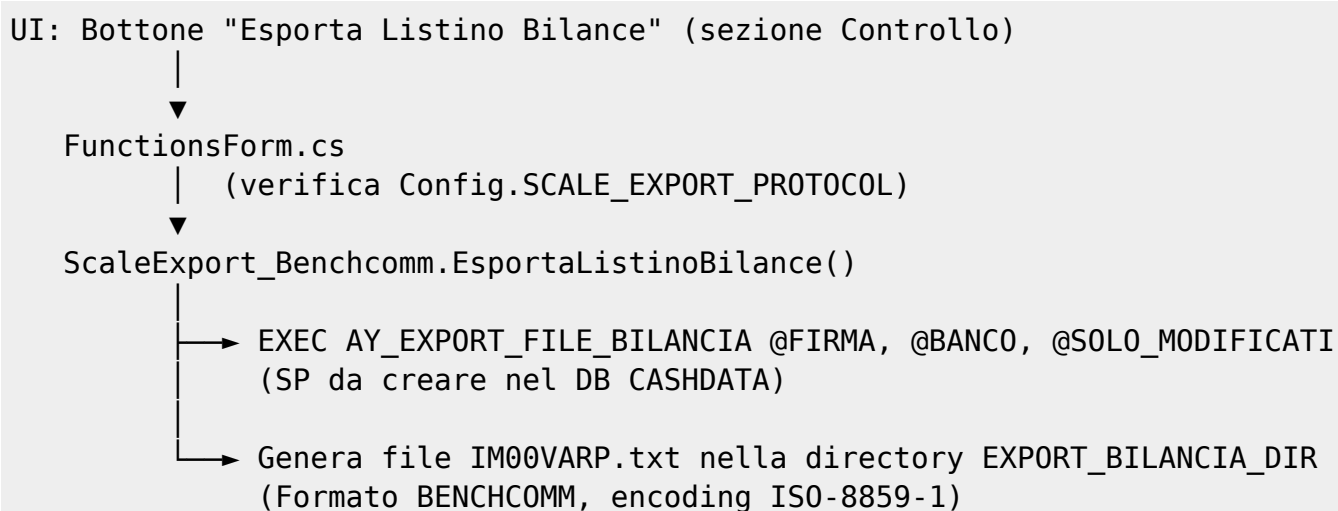
Script di riferimento: RELEASE\03 Integrazione con Bilancia\

11b.1 Panoramica

L'integrazione bilancia permette di:

- Esportare il listino articoli verso bilance elettroniche
- Supportare il formato **BENCHCOMM** (standard per bilance da banco)
- Generare file in formato testo a posizione fissa (100 caratteri per riga)

Flusso di esportazione:



11b.2 Parametri di Configurazione

Parametri da configurare in CASSA_CONFIG_PARAMETER:

Parametro	Sezione	Valore	Descrizione
SCALE_EXPORT_ENABLED	10	true	Abilita il bottone "Esporta Listino Bilance" nella sezione Controllo dell'UI
SCALE_EXPORT_PROTOCOL	10	BENCHCOMM	Formato/protocollo bilancia. Attualmente supportato: BENCHCOMM
EXPORT_BILANCIA_DIR	10	(es. C:\Export\Bilance)	Directory dove salvare i file di export. Default: C:\Export\Bilance

11b.3 Procedura di Installazione

STEP 1: Configurare i parametri

Inserire/aggiornare i 3 parametri nella tabella CASSA_CONFIG_PARAMETER:

```
-- Abilitare export bilancia
UPDATE CASSA_CONFIG_PARAMETER SET CP_VALUE = 'true'
```

```
WHERE CP_NAME = 'SCALE_EXPORT_ENABLED' AND CP_SECTION = 10

-- Impostare protocollo BENCHCOMM
UPDATE CASSA_CONFIG_PARAMETER SET CP_VALUE = 'BENCHCOMM'
WHERE CP_NAME = 'SCALE_EXPORT_PROTOCOL' AND CP_SECTION = 10

-- Impostare directory di export
UPDATE CASSA_CONFIG_PARAMETER SET CP_VALUE = 'C:\Export\Balance'
WHERE CP_NAME = 'EXPORT_BILANCIA_DIR' AND CP_SECTION = 10
```

STEP 2: Creare la Stored Procedure nel DB CASHDATA

Script: 03 Integrazione con Bilancia\00_Creare SP per estrarre dati da inviare alla bilancia.txt

Nota: Lo script nella cartella contiene attualmente la SP PRC_EXPORT_ERP (export chiusure). La SP effettiva per la bilancia e' AY_EXPORT_FILE_BILANCIA, che va creata in base alle specifiche della bilancia utilizzata.

La SP AY_EXPORT_FILE_BILANCIA deve accettare questi parametri:

```
CREATE PROCEDURE [dbo].[AY_EXPORT_FILE_BILANCIA]
    @FIRMA INT,
    @BANCO INT,
    @SOLO_MODIFICATI BIT
AS
BEGIN
    -- Restituisce un DataTable con i campi necessari al formato BENCHCOMM:
    -- Banco, CodicePLU, AliquotaIVA, Prezzo, CodiceEAN,
    -- DescrizioneRiga1, DescrizioneRiga2, GiorniConservazione, Tara, ecc.
    -- L'implementazione varia in base alla bilancia
END
```

STEP 3: Creare la directory di export

Creare la cartella specificata nel parametro EXPORT_BILANCIA_DIR e verificare che l'utente Windows che esegue CASHDATA abbia permessi di scrittura.

STEP 4: Configurare la bilancia

Configurare la bilancia per leggere il file IM00VARP.txt dalla directory di export. Il formato e':

- **Protocollo:** BENCHCOMM
- **Encoding:** ISO-8859-1
- **Lunghezza riga:** 100 caratteri + CR+LF
- **Campi a posizione fissa** (pos 1-99 con dati per banco, PLU, IVA, prezzo, EAN, descrizione, ecc.)

11b.4 Checklist Integrazione Bilancia

- Configurare parametro SCALE_EXPORT_ENABLED = true
- Configurare parametro SCALE_EXPORT_PROTOCOL = BENCHCOMM
- Configurare parametro EXPORT_BILANCIA_DIR con il percorso corretto
- Creare la SP AY_EXPORT_FILE_BILANCIA nel DB CASHDATA (personalizzare per tipo bilancia)
- Creare la directory di export con permessi di scrittura
- Configurare la bilancia per leggere da EXPORT_BILANCIA_DIR\IM00VARP.txt
- Testare l'export dal bottone "Esporta Listino Balance" nella sezione Controllo
- Verificare che il file generato sia corretto (100 char/riga, encoding ISO-8859-1)

11b.5 Troubleshooting Integrazione Bilancia

Errore	Causa	Soluzione
Bottone "Esporta Listino Balance" non visibile	SCALE_EXPORT_ENABLED != true	Impostare parametro a true in CASSA_CONFIG_PARAMETER
"Formato bilancia 'X' non riconosciuto"	Protocollo non supportato	Verificare che SCALE_EXPORT_PROTOCOL = BENCHCOMM (unico formato supportato)
Errore durante export	SP AY_EXPORT_FILE_BILANCIA mancante	Creare la SP nel database CASHDATA
File non generato	Directory non esistente o senza permessi	Verificare che EXPORT_BILANCIA_DIR esista e sia scrivibile
Bilancia non legge i dati	Formato file errato	Verificare encoding ISO-8859-1 e lunghezza righe (100 + CRLF)

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 in CashData.xml
"Login failed"	Credenziali errate	Verificare username/password in CashData.xml
"Database not found"	DB non esiste	Creare database ed eseguire script FASE 1-2

13.2 Errori di Prima Installazione

Questi errori si verificano tipicamente al primo avvio dopo una nuova installazione. La sequenza di avvio dell'app (HomeLogic.cs) verifica molti parametri e tabelle in ordine.

Flusso di avvio (ordine di verifica)

```

1. BackupAndVerifyDataBase()    -- Solo in Release, richiede PCisSqlServer
2. ReadConfigParameter("LanguageCode")
3. ReadConfigParameter("INSTALLFOLDER") -- Prima installazione se vuoto
4. CheckLic()                  -- Verifica licenza
5. ReadDefaultPrinter()        -- Verifica stampante
6. ReadParameter("HOMEMENU_ENABLED")
7. PulisciScontrino()          -- Usa CCO_CASH_NR su CASSA_COUPON
8. LoadParametri()             -- Carica tutti i getParValueByKey()
9. LoadScontrinoDefaultsValues() -- Contatori (NEW_DAYLY_CLOSURE,
RECEIPT_COUNTER_PWD, ecc.)
10. Login()                    -- Richiede utenti e gruppi

```

Tabella errori e soluzioni

Errore	Causa Radice	Soluzione
"Errore lettura parametro: INSTALLFOLDER"	Parametro mancante in CASSA_CONFIG_PARAMETER sezione 30	Eseguire 03_FIX_PARAMETRI_MANCANTI.sql oppure verificare che 02_DB_CREATION_TABLES.sql sia aggiornato
"First installation procedure invalid. Contact Software House !!!"	I campi LIC_SOSerial, LIC_Volume, LIC_ACTIVATE in CASSA_CONFIG_PARAMETER (sezione 30) contengono valori template non vuoti, ma il sistema rileva una prima installazione	UPDATE CASSA_CONFIG_PARAMETER SET CP_VALUE = WHERE CP_NAME IN ('LIC_SOSerial', 'LIC_Volume', 'LIC_ACTIVATE') AND CP_SECTION = 30

Errore	Causa Radice	Soluzione
“Autorizzazione CREATE DATABASE negata” / “Errore durante creazione backup database”	BackupAndVerifyDataBase() esegue BACKUP DATABASE in modalita Release, ma l'utente SQL non ha permessi BACKUP	Impostare PCisSqlServer = false in CashData.xml. Impostare true solo se l'utente SQL ha permessi di backup
“La stampante non e' installata” (loop infinito)	CASSA_DOC_PRINTER.DPR_DEFAULTPRINTER e' NULL per tutti i tipi documento. Il codice in Utility.cs:1899 ha un bug nel check IsDBNull	UPDATE CASSA_DOC_PRINTER SET DPR_DEFAULTPRINTER = 'Microsoft Print to PDF' (gia' incluso nello script di creazione aggiornato)
“Nessun gruppo configurato nel sistema”	Tabelle CASSA_USER_GROUP e CASSA_USER vuote	Gia' incluso nello script di creazione aggiornato. Per DB esistenti: inserire manualmente gruppi e utenti (vedi sezione 3.1 FASE 6)
“Errore lettura parametro: HOMEMENU_ENABLED”	Parametro mancante in tabella CASSA_PARAMETER (non CONFIG_PARAMETER!)	INSERT INTO CASSA_PARAMETER (CP_FIRMA, CP_NAME, CP_VALUE, CP_TEXT) VALUES (10, N'HOMEMENU_ENABLED', N'0', N'Home menu enabled')
“Il nome colonna CCO_CASH_NR” a HomeLogic.cs:1497	Colonna CCO_CASH_NR mancante nella tabella CASSA_COUPON	ALTER TABLE [CASSA_COUPON] ADD [CCO_CASH_NR] [int] NULL (gia' incluso nello script di creazione aggiornato)
“Errore lettura parametro: NEW_DAYLY_CLOSURE”	Parametro letto con ReadParameter(“NEW_DAYLY_CLOSURE”, 0, Config.CASHNUMBER) quindi va in CASSA_PARAMETER_PC (non CASSA_PARAMETER!)	INSERT INTO CASSA_PARAMETER_PC (CP_FIRMA, CP_NAME, CP_VALUE, CP_TEXT, CASH_NR) VALUES (10, N'NEW_DAYLY_CLOSURE', N'1', N'Daily closure counter', 1)
“Errore lettura parametro: RECEIPT_COUNTER_PWD”	Parametro letto con ReadParameter(“RECEIPT_COUNTER_PWD”, 0) dalla tabella CASSA_PARAMETER. Il valore deve essere non vuoto	INSERT INTO CASSA_PARAMETER (CP_FIRMA, CP_NAME, CP_VALUE, CP_TEXT) VALUES (10, N'RECEIPT_COUNTER_PWD', N'tex01', N'Password reset contatore')
“Errore lettura parametro: X” (generico)	Parametro X mancante. Individuare la tabella corretta verificando il codice sorgente	Se ReadParameter(name, 0) → CASSA_PARAMETER. Se ReadParameter(name, 0, Config.CASHNUMBER) → CASSA_PARAMETER_PC. Se getParValueByKey(name) → CASSA_CONFIG_PARAMETER

13.3 Come Distinguere le Tabelle dei Parametri

Quando si incontra un errore “Errore lettura parametro”, e' fondamentale capire **quale tabella** deve contenere il parametro:

Metodo di lettura nel codice	Tabella DB	Identificato da
getParValueByKey(“NOME”)	CASSA_CONFIG_PARAMETER	Errore da Utility.cs:3808
ReadConfigParameter(“NOME”, releasedfor, section)	CASSA_CONFIG_PARAMETER	Errore da HomeLogic.cs:734
ReadParameter(“NOME”, 0) (2 argomenti)	CASSA_PARAMETER	cash_nr=0 (default)

Metodo di lettura nel codice	Tabella DB	Identificato da
ReadParameter("NOME", 0, Config.CASHNUMBER) (3 argomenti)	CASSA_PARAMETER_PC	cash_nr != 0

13.4 Errori di Configurazione

Errore	Causa	Soluzione
"Parametro mancante: X"	Parametro non configurato	Identificare la tabella corretta (vedi 13.3) e inserire il parametro
"ReleaseFor non valido"	PC non registrato	Configurare ReleaseFor in CashData.xml
Parametri con valore vuoto	Alcuni parametri (es. RECEIPT_COUNTER_PWD) richiedono un valore non vuoto	Verificare che CP_VALUE non sia vuoto per i parametri che lo richiedono

13.5 Errori Operativi

Errore	Causa	Soluzione
"Differenza $\neq$ 0"	Conteggio errato	Verificare contanti inseriti
"Pagamenti non verificati"	Manca spunta	Completare verifica pagamenti
"Nuovo fondo negativo"	Prelievo eccessivo	Ridurre importo prelievo

13.6 Log e Diagnostica

- **Log accessi:** Tabella CASSA_USER_OP
- **Log operazioni:** Consultare con report CASHPROTOCOL
- **Errori applicazione:** Verificare file di log in cartella installazione
- **Debug SQL:** Controllare i parametri direttamente nel DB con:

```
-- Verificare parametri di configurazione
SELECT * FROM CASSA_CONFIG_PARAMETER WHERE CP_NAME = 'NOME_PARAMETRO'
-- Verificare parametri runtime
SELECT * FROM CASSA_PARAMETER WHERE CP_NAME = 'NOME_PARAMETRO'
-- Verificare parametri per-PC
SELECT * FROM CASSA_PARAMETER_PC WHERE CP_NAME = 'NOME_PARAMETRO'
```

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 + validazione DDT per ERP
UpdateUtility.cs	~1600	Importazione dati
FunctionsForm.cs	~1800	Sezione Controllo (export ERP, bilancia)
ScaleExport_Benchcomm.cs	~100	Export listino balance formato BENCHCOMM

B.2 Script Database - Installazione Base

File	Cartella	Descrizione
00_DB_CREATION_CREATEDB.sql	RELEASE\01\	Creazione database
01_DB_CREATION_PROCEDURE.sql	RELEASE\01\	Stored procedures e viste (V_CASSA_CHIUSURE, V_CASSA_VENDITE_ARTICOLI)
02_DB_CREATION_TABLES.sql	RELEASE\01\	Tabelle, dati iniziali, utenti, gruppi, stampanti
03_DB_CREATION_WITH_PROMO.sql	RELEASE\01\	Dati promozioni
02 Reimposta dati.sql	Cose da fare...\	Configurazione iniziale (personalizzare prima dell'esecuzione)
03_FIX_PARAMETRI_MANCANTI.sql	Cose da fare...\	Fix parametri mancanti per DB esistenti

B.3 Script Integrazione ERP2

File	Cartella	Target DB	Descrizione
00_Creare linked server.txt	RELEASE\02...\	master	Linked server verso DB ERP2
01_Tabelle&Viste da creare su db ERP.txt	RELEASE\02...\	DB ERP	Tabelle AY_CASSA_*, EXT_ORDINI_*, vista lotti, login
02_Da fare su DB CashData.txt	RELEASE\02...\	DB CASHDATA	Sinonimi ERP_*, viste V_ERP_*, SP AY_CARICA_DDT_DA_ERP, PRC_EXPORT_ERP

B.4 Script Integrazione Bilancia

File	Cartella	Target DB	Descrizione
00_Creare SP per estrarre dati.txt	RELEASE\03...\	DB CASHDATA	SP AY_EXPORT_FILE_BILANCIA (template, personalizzare per tipo bilancia)

Documento generato automaticamente dall'analisi del codice sorgente CASHDATA.

Ultimo aggiornamento: Febbraio 2026

Versione: 1.2 - Aggiunta integrazione ERP2 e bilancia

From:

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

Permanent link:

<http://kb.analytic.it/cashdata>

Last update: **2026/02/11 14:54**

