

SQL Server XAML Multi Desktop
MDI Report Editor .Net Framework
Scheduling n-tier Application ADO.NET M-V-VM
Global Navigation ORM MRP II Oracle
C-Sharp Customizable Database MySQL MRP I
Entity Framework Metro Style 64Bit
ERP WPF CLR
Database Independent 32Bit C# 4 tier
LINQ CRM Enterprise Resource Planning
Trend Extensible Framework Trend Plug-in Pro
Business Intelligence

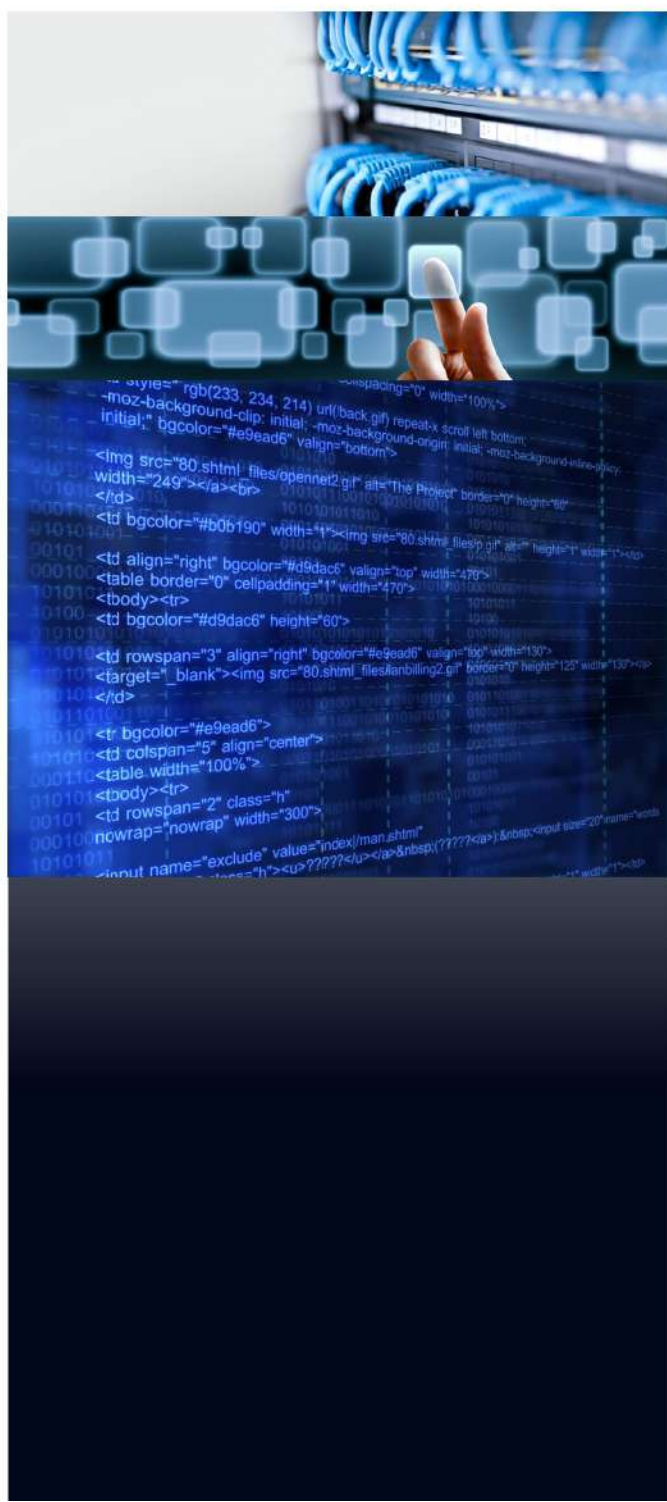
ERP Trend Enterprise Tecnologia

TDAINFORMATICA
Software House

SOLUZIONI GESTIONALI PER LE AZIENDE

SOFTWARE ERP TREND ENTERPRISE: LO STRUMENTO NEL FUTURO DELL'AZIENDA

QUESTO DOCUMENTO È UNA INTRODUZIONE E UNA PICCOLA GUIDA, ALLE TECNOLOGIE CON LE QUALI È STATO PENSATO, PROGETTATO E REALIZZATO L'APPLICATIVO GESTIONALE **ERP TREND ENTERPRISE**.



PREMESSA

Questo documento è un'introduzione o, se volete, una piccola guida, alle tecnologie con cui è stato pensato, progettato e realizzato l'applicativo gestionale ERP Trend Enterprise. Poiché non tutti i lettori di questo documento sono esperti delle terminologie e delle più moderne tecniche di sviluppo software, abbiamo pensato di scriverlo in due forme: una chiara e comprensibile a tutti, una per esperti programmatori.

State per iniziare la lettura per i più esperti... ribaltate la brochure per leggere l'altra.

Lo scopo di questa guida è dimostrare che Trend Enterprise è progettato secondo le ultime tecnologie che, Microsoft e le comunità di sviluppatori, indicano come riferimento per gli anni a venire.

INTRODUZIONE

Trend Enterprise è un software client-server che opera in ambiente .NET, realizzato con una architettura di tipo n-tier o multi-tier, un aggettivo che può essere spiegato come "applicazione a n-strati o livelli". Quali sono, dunque, gli strati che costituiscono un software multi-strato? Teoricamente sono tre (3-tier): Presentazione dei Dati, Accesso ed Elaborazione dei Dati e Sistema di Gestione dei Dati. Il primo è l'interfaccia utente, il secondo è il motore dell'applicazione che trasforma i dati, il terzo è il sistema che memorizza e gestisce i dati, comunemente chiamato database. I vantaggi della stratificazione sono dovuti al fatto che, per apportare modifiche alle procedure, è sufficiente modificare solo il livello interessato alle variazioni, il che rende l'applicazione molto flessibile e le parti del livello intermedio utilizzabili per più di una procedura. L'architettura di Trend Enterprise è di tipo 4-tier.

TREND ENTERPRISE E LA TECNOLOGIA .NET IL FUTURO DEI SOFTWARE GESTIONALI

L'ARCHITETTURA 4-TIER E LA TECNOLOGIA .NET FANNO DI TREND ENTERPRISE UNO STRUMENTO IL CUI FUTURO È ALLINEATO ALLE INDICAZIONI DI MICROSOFT E DELLE COMUNITÀ DI SVILUPPATORI.

TREND ENTERPRISE E I SUOI COLLEGHI

Facciamo alcune considerazioni che ci permetteranno di distinguere **Trend Enterprise** dalla maggior parte degli altri software gestionali, attualmente reperibili nel mercato: la maggior parte di questi sono progettati con una architettura di tipo 2-tier, a due livelli. Solitamente, infatti, uno strato contiene sia le interfacce sia le logiche di elaborazione dei dati e un altro comprende i dati stessi. Si può affermare che quest'architettura, implicitamente, limita l'utilizzo del software o le sue capacità, ad esempio:

- 1- Un software 2-tier sarà un'applicazione desktop o per il web o per un altro tipo d'infrastruttura, ma non entrambe.
- 2- Il database, che contiene i dati, determina lo sviluppo dell'applicazione; ne consegue che non è possibile scegliere tra diversi "database".
- 3- I programmatori, quando sviluppano una nuova interfaccia, devono riscriverla insieme alle logiche per l'elaborazione dei dati e queste non saranno riutilizzabili.

Questo perché i suddetti software sono stati realizzati, probabilmente, con strumenti che risalgono agli anni passati e che non avevano le prerogative che avrebbero permesso ai programmatori di sviluppare diversamente. In questo caso tutti i produttori di software dovranno fare i conti con la politica di Microsoft che, in un periodo di circa 3 o 4 anni, non riproporrà la compatibilità dei nuovi sistemi operativi con questi strumenti. In caso diverso non avrebbe nessuna logica, pensando al futuro, scegliere questo tipo di architettura.



L' ARCHITETTURA 4-TIER DI TREND: TRAZIONE INTEGRALE

I COMPONENTI PRINCIPALI DELL' ARCHITETTURA DI TREND SONO QUATTRO: APPS, MODEL, ORM, RDBMS; SONO QUESTI GLI "STRATI" (TIER) DELL' APPLICAZIONE .NET



L'ARCHITETTURA DI TREND

Abbiamo detto che Trend Enterprise è un software client-server con architettura di tipo n-tier, e precisamente 4-tier; è stato realizzato in ambiente .NET con il linguaggio di programmazione C# (C-Sharp) di Microsoft. Lo schema successivo mostra i quattro strati e, per ognuno di essi, i contenuti tecnologici e "sottostrati" di cui si compongono. Come abbiamo anticipato Vi proponiamo due tipi di guida alla comprensione di questo schema, una molto più tecnica dell'altra: scegliete quella che meglio Vi si addice

TREND FRAMEWORK

Tutte le Apps Standard di Trend Enterprise, sono state sviluppate utilizzando Trend Framework, che può essere utilizzato dai programmatori (siano essi i componenti dei reparti CED oppure i programmatori dei Rivenditori di Trend) per sviluppare nuove APPS o per customizzare quelle già esistenti, col fine di soddisfare le esigenze dell'utente finale.

Trend Framework semplifica notevolmente lo sviluppo di nuove applicazioni; il programmatore si dovrà concentrare esclusivamente sulla APPS che deve realizzare e non su come realizzarla, in quanto Trend Framework gestisce automaticamente tutte le regole di binding, di verifica e controllo (di cosa?), di gestione delle grid e di relazione tra le tabelle. Le stesse routine per la reportistica sono integrate in Trend Framework e, utilizzando l'apposito editor, è possibile modificare le stampe esistenti o crearne di nuove.

I programmatori, sfruttando gli editor dell'Extensible Database Model, possono aggiungere e modificare i campi, le tabelle e le relazioni delle classi di dominio in modo trasparente; queste modifiche saranno automaticamente inviate (a chi ?) aggiornando il database e rese disponibili allo strato Application.



L' AMBIENTE MICROSOFT .NET: UNA GARANZIA PER IL FUTURO

LA TECNOLOGIA CHE MICROSOFT E LE COMUNITA' DI SVILUPPATORI, INDICANO COME RIFERIMENTO PER GLI ANNI A VENIRE.

RDBMS, RELATIONAL DATABASE MANAGEMENT SYSTEM

Lo strato RDBMS è il più semplice da comprendere. Comunemente chiamato database, è identificato, spesso e in maniera semplicistica, con "l'Archivio dei Dati". Con le architetture di tipo 1, 2, 3-tier, i programmatori sono obbligati a sviluppare le logiche dei software in funzione del RDBMS che sarà utilizzato. L'architettura 4-tier, al contrario, permette l'indipendenza dell'applicazione dal RDBMS, poiché uno strato di cui parleremo in seguito, l'ORM, funge da "interprete" tra il database e il livello dell'applicazione stessa, detta CORE oppure MODEL. Ciò è molto importante, perché in questo modo Trend Enterprise diventa "RDBMS Indipendente" e, le aziende che lo adotteranno, potranno scegliere il database in funzione delle proprie esigenze, oppure, eventualmente, utilizzare quello di cui già dispongono.

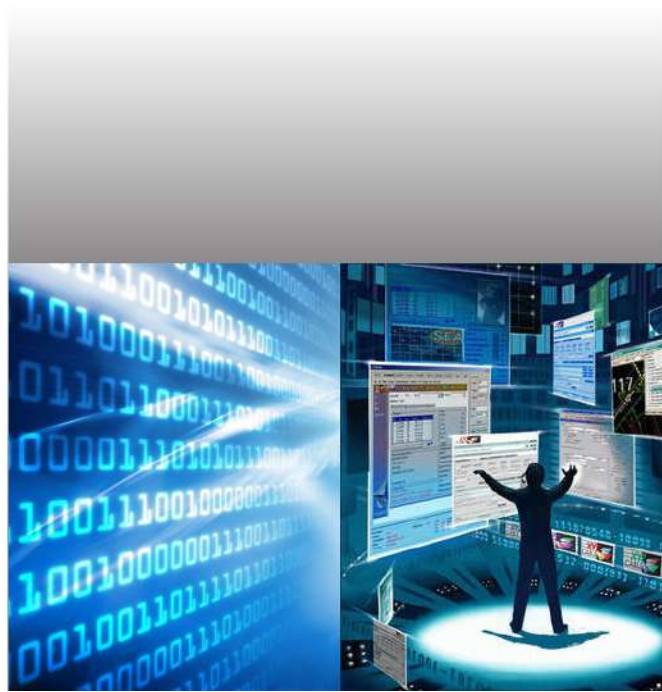
In effetti, purché il RDBMS sia compatibile con le specifiche del mondo ADO.NET potrà essere utilizzato con il nostro applicativo. Si contano ormai oltre cinquanta "database", compatibili con ADO.NET, tra i quali sono compresi SQL, Oracle e MYSQL.



ORM, OBJECT-RELATIONAL MAPPING

Abbiamo già accennato alla funzione di "interprete" dello strato ORM tra il livello MODEL o CORE, e il RDBMS che determina l'indipendenza dell'applicazione .NET dal sistema che gestisce i dati. L'ORM di cui è dotato Trend Enterprise è ENTITY FRAMEWORK, un componente di .NET Framework 4.5 di Microsoft, cui abbiamo implementato TDEF, Trend Dynamics Entity Framework.

EF, ENTITY FRAMEWORK, rende disponibili al MODEL (il motore dell'applicazione) i dati contenuti nel database, secondo la tecnologia ADO.NET. La cosa più interessante è che i programmatori non devono conoscere quale database sarà utilizzato dall'applicazione; questo consente loro di scrivere molto meno codice per l'elaborazione dei dati in quanto l'accesso agli stessi è gestito dall'interprete. L'ORM converte le operazioni richieste dall'applicazione nei comandi specifici del RDBMS cui è connesso. È stato necessario implementare nell'ORM un sottostrato con funzioni specifiche quali, ad esempio, la gestione di architetture multiprocessore, l'ottimizzatore della velocità di accesso ai dati e una gestione avanzata del blocco dei record per una migliore amministrazione dei casi di concorrenza sui dati. Ulteriormente sono state implementate numerose librerie che permettono l'accesso dinamico ai dati.



LO STATO MODEL O CORE DELL' APPLICAZIONE: IL MOTORE POTENTE DI TREND ENTERPRISE

IL LIVELLO CORE, DETTO ANCHE MODEL, DELL' APPLICAZIONE E', A TUTTI GLI EFFETTI, IL MOTORE DELL' ERP TREND ENTERPRISE. QUESTO MOTORE PUÒ ESSERE MODELLATO IN BASE ALLE ESIGENZE DELL' UTILIZZATORE, PER OTTIMIZZARE IL FUNZIONAMENTO.

MODEL O CORE

Lo strato MODEL o CORE è il motore dell'applicazione Trend Enterprise. È un insieme di librerie che sono utilizzate dallo strato APPLICATION per elaborare i dati e esportarli all'utilizzatore. IL CORE colloquia con lo strato ORM e restituisce i dati derivati dall'elaborazione alle APPS di Trend Enterprise.

IL CORE è a sua volta suddiviso in sei differenti aree che interagiscono tra loro, e che vedremo separatamente:

- Extensible Database Model
- Trend Generic Library
- Trend Data Access Library
- Trend Reporting
- Trend Controls
- Trend Extensible Framework

Extensible Database Model

Extensible Database Model permette di modificare la struttura dati dell' applicazione Trend Enterprise in modo semplice ed intuitivo.

Attraverso un apposito editor si possono editare le strutture delle tabelle, delle relazioni, dei campi e così via.

E' possibile ad esempio aggiungere nuovi campi, modificare la dimensione dei campi già presenti, aggiungere nuove tabelle e nuove relazioni.

Ogni modifica effettuata nell' editor viene propagata automaticamente sia al database che alle classi di dominio in modo trasparente all' utente.

Le classi di dominio rappresentano il "ponte" tra il Core di Trend Enterprise ed il Database.

Ogni applicazione di TrendEnterprise (Trend App) colloquia esclusivamente con il Core (Model) che a sua volta attraverso le classi di dominio riceve e/o invia i dati al database.

Questo concetto viene generalmente chiamato "**persistence ignorance**" ovvero il programmatore non deve conoscere la sintassi del database (Trend Enterprise e' infatti database indipendente e ogni database ha un proprio dialetto) e non deve neanche conoscere i vari tools messi a disposizione dai database stessi per modificare, ad esempio, le strutture delle tabelle; lo sviluppatore deve esclusivamente "dialogare" con lo strato applicativo esposto dal Framework di TrendEnterprise.

Tutte le operazioni, generalmente chiamate CRUD (Create Read Update Delete), verso il Database vengono gestite dal Framework stesso.

IL Framework gestisce, oltre alle operazioni CRUD, operazioni di ottimizzazione per rendere l' applicazione e l' accesso ai dati il più prestante possibile; attraverso il parallelismo e componenti LINQ estremamente indicizzati anche su database di grandissime dimensioni si hanno prestazioni vicine al tempo reale.

Modello Relazionale -> Modello Gerarchico Reticolare

Trend Enterprise utilizza come modello di accesso ai dati il

Modello Gerarchico Reticolare.

Nel modello Gerarchico Reticolare **le relazioni tra i dati fanno parte della struttura stessa delle classi** esposte nel dominio; questo significa che lo sviluppatore non deve necessariamente conoscere com'è organizzata la base dati e quindi conoscerne la struttura per scrivere la classica query SQL con le relative join; sarà, invece, il Framework a scriverla per lui evitando errori di scrittura in quanto le query vengono generalmente scritte in modo tipizzato e qualsiasi errore (nome campo errato, navigation property errata ecc.) viene evidenziata a priori già in fase di compilazione e non in fase di esecuzione.

Facciamo un esempio per capire meglio il concetto di Modello Gerarchico Reticolare :

lineaMovMag e' l'oggetto della corrente linea del movimento di magazzino :

Se vogliamo ricavare ad esempio la descrizione della categoria merceologica possiamo scrivere :

```
var desCategoria = lineaMovMag.np_ges_prod.np_cat_merc.  
descrizione;
```

In TrendEnterprise il prefisso np_ indica Navigation Property che, in se, ha le indicazioni di relazione (1..1, 1.. molti oppure molti .. molti) per accedere ai dati in modo gerarchico e reticolare.

Np_ges_prod e' la navigation property che "collega" la corrente linea del movimento di magazzino all' anagrafica prodotti, che a sua volta ha una navigation property, np_cat_merc, che collega il prodotto alla relativa categoria merceologica.

Sarà il framework che automaticamente invia la corretta istruzione SQL al database con le relative join per estrarre la descrizione della categoria.



RPRISE.
MENTO SULLA BASE DELLE PROPRIE ESIGENZE.

Trend Generic Library

Trend Generic Library è la parte del Core di Trend Enterprise che espone classi, metodi e command ad uso trasversale e generale.

Le classi esposte da questa porzione della libreria vengono utilizzate sia dalle Trend Apps che dalle altre parti del Framework stesso.

In questa libreria sono presenti classi di utilizzo generale come ad esempio:

- Apertura e connessione al database.
- Interrogazioni sullo stato di Lock dei Record.
- Esecuzione di query dinamiche (tipizzate e non tipizzate in base all' occorrenza).
- Visualizzazione di form nel corrente desktop selezionato.
- Caricamento di immagini e gestione avanzata dei controlli.
- Esecuzione di animazioni.
- Conoscere lo stato dell' utente loggato (ad es: se e' in Development Mode)
- Conoscere lo stato dell' attuale DB selezionato (Multiazienda)
- Gestione delle Global per accesso ai dati tramite iperlink.
- Metodi e Classi per semplificare l' uso del pattern MWM.
- E così via.

Trend Data Access Library

Trend Data Access è la parte del Core di Trend Enterprise che espone classi, metodi e command riguardanti l' accesso ai dati. Le classi esposte da questa porzione della libreria vengono generalmente utilizzate direttamente dalle Trend Apps.

In questa libreria sono presenti classi che si occupano esclusivamente dell' accesso alle informazioni e della relativa gestione.

Fanno parte ad esempio di questa libreria tutta la gestione del Repository con i vari command, metodi ed eventi esposti per gestire automaticamente le operazioni C.R.U.D. delle interfacce video.

Questa parte della libreria si occupa anche dell' ottimizzazione dell' accesso ai dati sfruttando un motore di oggetti in memoria fortemente indicizzato, viene gestito, ove necessario il parallelismo e l' utilizzo multiprocessore.

Il pattern utilizzato è l' MWM (Model View View-Model) che permette di separare le interfacce grafiche dal codice.

Trend Data Access effettua automaticamente una scansione avanzata dell' object tracking ottimizzando le risorse della memoria attraverso un' indicizzazione degli stessi per un notevole aumento prestazionale.

Trend Data Access è la parte del Core che si occupa del Record Locking gestendo automaticamente le informazioni sul Lock dei record come ad esempio l' utente che sta bloccando il record, il terminale l' ora e la data di blocco.

Il Record Locking, a seconda delle esigenze e della dimensione aziendale, può essere configurato come Ottimistico o Pessimistico. Il Record locking pessimistico, in ogni caso, permette in fase di lock la lettura del record ma non la modifica dello stesso.

Trend Data Access e' composto da un' ulteriore strato chiamato Application all' interno del quale sono presenti le classi ed i metodi strettamente applicativi. Lo strato Application è a sua volta suddiviso in varie classi a seconda del modulo interessato ad es:

- Application.mov_mag : Insieme di classi e metodi utilizzabili dal modulo magazzino (determinazione del prezzo di un prodotto, numerazione del protocollo di magazzino, gestione delle varianti di prodotto ecc. ecc.)
- Application.ges_dbas: Insieme di classi e metodi utilizzabili dal modulo distinte di base (sviluppo a db monolivello, multilivello, gestione delle varianti di distinta, determinazione del prezzo di un prodotto sviluppando la relativa distinta base ecc. ecc.)
- Application.mov_con: Insieme di classi e metodi utilizzabili dal modulo movimenti contabili (creazione automatica di una reg. contabile passando i parametri necessari, totale iva di una determinata registrazione, totale prima nota di una determinata registrazione, saldo scheda contabile di un cliente – fornitore o conto piano dei conti, ecc. ecc.)
- Non elenchiamo qui tutte le suddivisioni dello strato application, ogni modulo di Trend Enterprise ha, generalmente, il proprio strato application.



LO STATO MODEL O CORE DELL' APPLICAZIONE: IL MOTORE POTENTE DI TREND ENTERPRISE

IL LIVELLO CORE, DETTO ANCHE MODEL, DELL' APPLICAZIONE E', A TUTTI GLI EFFETTI, IL MOTORE DELL' ERP TREND ENTERPRISE. QUESTO MOTORE PUÒ ESSERE MODELLATO IN BASE ALLE ESIGENZE DELL' UTILIZZATORE, PER OTTIMIZZARE IL FUNZIONAMENTO.

Trend Reporting

Trend Reporting è la parte del Core di Trend Enterprise che si occupa della gestione della reportistica. Le classi e metodi esposti da questa porzione della libreria vengono generalmente utilizzate direttamente dalle Trend Apps.

Trend Reporting ha un **Viewer di report WYSIWYG** (What You See Is What You Get) integrato che oltre alle funzioni di visualizzazione classiche come zoom, adattamento alla pagina, navigazione sulle pagine, ricerca del testo contenuto nel report, ha funzioni avanzate quali iperlink sul report stesso o report correlati, grafici e kpi.

Il **Viewer** di report è di tipo **vettoriale**, permette quindi ingrandimenti del testo senza perdita di risoluzione, oltre a dare la possibilità di ruotare il testo stesso.

Altre funzioni interessanti sono l' **esportazione nativa** dei report nei seguenti formati:

- PDF
- HTML
- MHT
- RTF
- XLS (Excel)
- XLSX (Excel nuovo formato)
- CSV
- Text
- Image (png, bmp, emf, wmf, tiff, jpg ecc. ecc.)
- XPS

Possibilità di **inviare automaticamente una Mail** allegando il report nei formati sopra indicati.

Salvataggio del Report nel formato nativo, per riaprirlo successivamente dal viewer stesso.

Oltre al viewer Trend Reporting integra un **Editor di Report** che permette di modificare i report standard di Trend in base alle esigenze dell' utente finale e/o di creare nuovi report da zero.

L' editor di Report si attiva automaticamente quando l' utente (con gli appositi privilegi) attiva la "modalità sviluppo".

Questo permette di accedere al report che si desidera modificare direttamente dall' interfaccia di Trend (menù, tool bar ecc. ecc.). Anche l' editor di report e' di tipo **WYSIWYG**, l' utente, quindi, modifica il report con una comoda interfaccia utente e tramite il Drag & Drop degli oggetti grafici.

Il dataBinding dei dati viene controllato dal modulo Trend Data Access; quindi l' utente per aggiungere e/o modificare la sorgente dei dati sfrutta le navigation property esposte dalle classi di dominio, in questo modo, automaticamente, **tutte le relazioni vengono gestite automaticamente** dal core di Trend Enterprise.

Logicamente, come già spiegato, Trend Data Access colloquia con L' Extensible Database Model, per cui, se il programmatore ha aggiunto tabelle, campi e/o relazioni automaticamente queste vengono viste dall' editor stesso con la possibilità, quindi, di aggiungerle sia sui report standard di Trend Enterprise che sui nuovi report.

Trend Enterprise ha inoltre la possibilità di selezionare gli **stili della reportistica** per modificare in una sola parte i colori, i font, le dimensione ecc. ecc. di tutti i report di Trend.

Gli stili gestiscono la gerarchia di classe, per cui, se non si modificano le proprietà tutti gli oggetti del report ereditano dallo stile correntemente selezionato a meno che l' operatore voglia impostare una proprietà differente.

Sono già presenti nel setup di Trend Enterprise diversi stili standard, l' operatore ha poi la possibilità di creare stili personalizzati e/o di modificare quelli standard.

Trend Controls

Trend Controls è un' insieme di Controlli (Custom Control e User Control) che vengono utilizzati dalle Apps di Trend Enterprise.

Il Programmatore può scegliere di utilizzare questi controlli oppure può anche optare per l' utilizzo di controlli standard o di terze parti.

I controlli di Trend hanno varie caratteristiche che permettono di velocizzare notevolmente lo sviluppo di applicazioni gestionali e/o di customizzare quelle già esistenti.

Una delle principali caratteristiche e' il Binding avanzato, ogni controllo espone una DependencyProperty (TrendControlSource) che automaticamente gestisce il Binding con il controllo stesso.

Se ad esempio vogliamo visualizzare su una Grid le linee di dettaglio della testata l' unica cosa che deve fare lo sviluppatore e' quella di valorizzare nel controllo teGridControl la proprietà (DependencyProperty) TrendControlSource. Ad esempio :

Questa istruzione inserita nello Xaml o selezionata nella finestra proprietà di Visual Studio:

TrendControlSource="Record.np_tab_vl"

gestisce automaticamente il Binding bidirezionale tracciando le modifiche e tutte le operazioni CRUD verso il database di tutte le linee dell' oggetto Record che e' esposto automaticamente nel Repository dalla libreria Trend Data Access.

Utilizzando i controlli standard del Framework si evita, quindi, di gestire tutta la mole di informazioni e codice inserito normalmente nel ViewModel quali ad esempio le ObservableCollection binding ecc. ecc. in quanto e' il Framework stesso che effettua questo lavoro per noi.

RPRISE.
MENTO SULLA BASE DELLE PROPRIE ESIGENZE.

Ci sono diversi controlli esposti dalle librerie quelli maggiormente utilizzati sono :

- teBrowseSearch	- teEditBoxValidate
- teButton	- teGridControl
- teButtonGlobal	- teGridColumn
- teButtonTransparent	- teGridViewCard
- teCheckBox	- teGridViewTable
- teCheckBoxDimmer	- teGridViewTreeList
- teCheckBoxPower	- teLabel
- teCheckBoxSlider	- teMemoEdit
- teComboBoxEdit	- teTabControl
- teDateEdit	- teTabItem
- teEditBox	- teWindow

Trend Extensible Framework

Trend Extensible Framework è la parte del Framework che si occupa della possibilità di personalizzare Trend Enterprise.

L'utente e/o il programmatore che possiede gli appositi privilegi ha la possibilità di attivare una funzione che noi chiamiamo modalità sviluppo. Quando Trend Enterprise è in modalità sviluppo permette di customizzare l'applicazione standard di Trend. Tutte le modifiche effettuate con la modalità sviluppo vengono posizionate in un sottostrato dell'applicativo ed eseguite a run-time. In questo modo si ha l'enorme vantaggio di non dover variare i sorgenti dell'applicazione permettendo l'aggiornamento alle successive release di Trend senza dover reinsieri le modifiche effettuate per l'utente finale.

Ogni modifica effettuata può essere rimossa, in questo caso, Trend riutilizzerà le funzioni standard dell'applicativo.

Attraverso Trend Extensible Framework si possono modificare le proprietà degli oggetti standard di Trend (colori, dimensioni, sorgente dati ecc. ecc.). Si possono creare nuovi oggetti nell'interfaccia sia esposti dal framework di Trend che oggetti standard del .NET Framework e/o di terze parti.

Si possono inoltre creare dei Plug-in (con qualsiasi linguaggio del .NET Framework) che modificano il comportamento standard delle App di Trend Enterprise. I plug-in possono modificare sia le "regole di calcolo" standard (ad es: se si vuole modificare il metodo di calcolo provvigionale e/o le regole di applicazione di ricarica dei prezzi ecc. ecc.) che le regole di interfaccia utente (ad es: si vuole inibire la cancellazione dei clienti agli utenti del gruppo magazzino, oppure si vuole abilitare la visualizzazione dell'IBAN della banca di appoggio dei clienti ai soli utenti del gruppo contabilità ecc. ecc.). Trend Extensible Framework permette in definitiva di modificare le applicazioni standard di Trend Enterprise per soddisfare le esigenze dell'utente finale senza doversi poi preoccupare della manutenzione delle stesse con i futuri aggiornamenti di release.

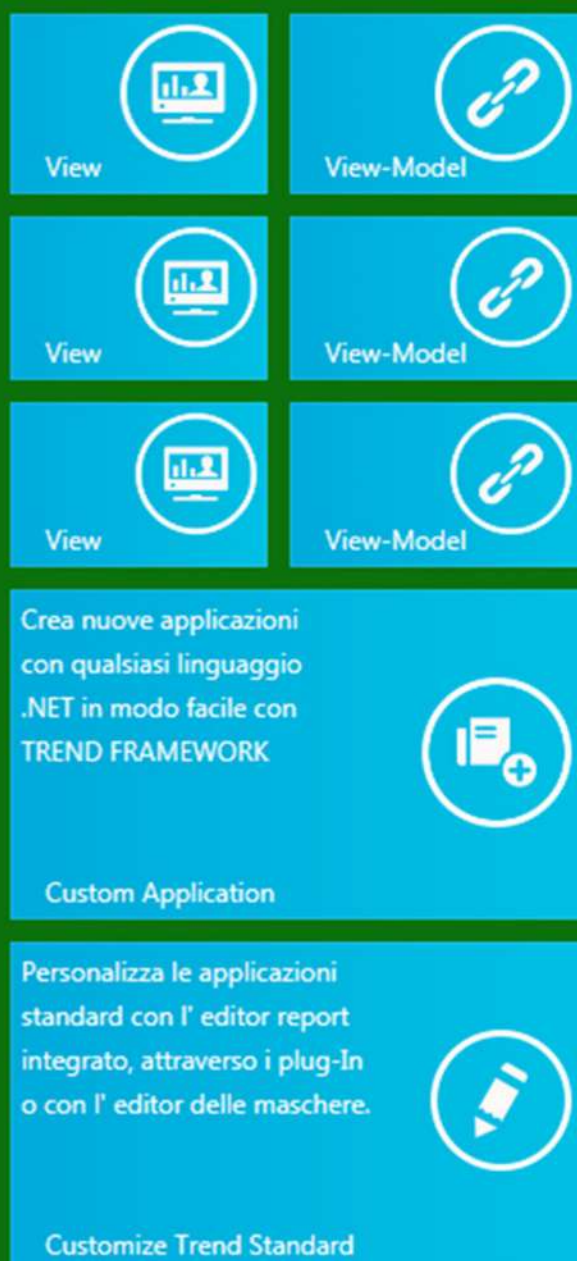
APPLICATION

Lo strato APPLICATION è quello che l'utilizzatore considera come il vero e proprio software. In questo livello si trovano le interfacce grafiche, che si collegano ai componenti del MODEL, per l'esposizione dei risultati delle elaborazioni.

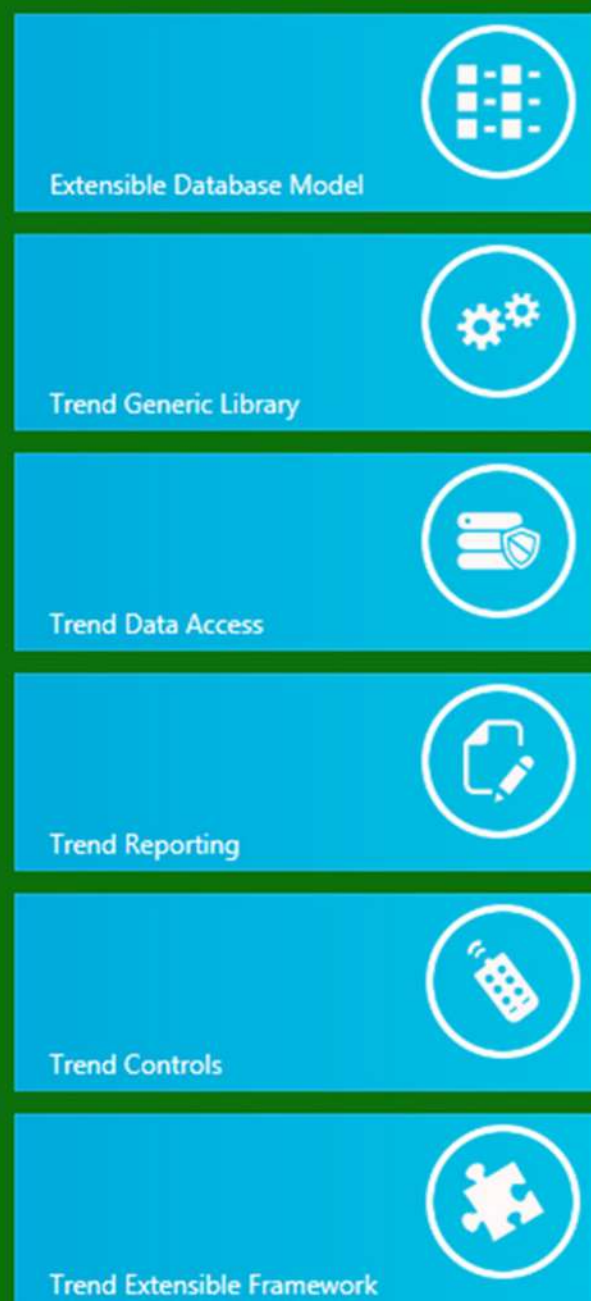
Due, soprattutto, sono le caratteristiche importanti di questo livello. Una è che le APPS possono essere scritte in uno qualsiasi dei linguaggi che supportano la tecnologia .NET. Questo permette ai programmatori di costruire nuove applicazioni per gli utenti sfruttando le competenze di cui, eventualmente, già dispongono. Non meno importante, come leggerete più avanti, il fatto che sia possibile sviluppare applicazioni per infrastrutture diverse e non solo applicazioni desktop per Windows, senza perdere il supporto del framework di Trend. Le Trend APPS Standard sono sviluppate con il linguaggio di markup XAML (Extensible Application Markup Language) di Microsoft, utilizzando il pattern di programmazione MVVM (Model View - View Model) che permette di separare l'interfaccia dal codice. Il MODEL, come abbiamo visto, è il motore o CORE dell'applicativo, la VIEW è l'interfaccia, la APP. Il VIEWMODEL è il collegamento tra queste. MVVM è diventato un pattern indispensabile a causa della crescita della complessità delle interfacce grafiche e della necessità dell'utenza di utilizzare lo stesso prodotto software con diverse infrastrutture, come il web, dispositivi desktop e i diversi apparati mobile per la telefonia. La separazione dal codice permette, semplicemente, di "ridisegnare" l'interfaccia secondo l'esigenza infrastrutturale, riutilizzando lo stesso codice.

SCHEMA APPLICATIVO DI TREND ENTERPRISE SVILUPPATA IN MANAGED CODE PER .NET

APPLICATION (MVVM PATTERN)



MODEL (CORE)



PRISE: APPLICAZIONE ERP, 4-TIER, T FRAMEWORK DI MICROSOFT

ORM



EntityFramework



Trend Dynamic EntityFramework

DATABASE



SQL Server



Oracle



My SQL



Altri database

SQL Server XAML Multi Desktop
MDI Report Editor .Net Framework
Scheduling n-tier Application ADO.NET M-V-VM
Global Navigation ORM MRP II Oracle
C-Sharp Customizable Database MySQL MRP I
Entity Framework Metro Style 64Bit
ERP WPF CLR
Database Independent 32Bit C# 4 tier
LINQ CRM Enterprise Resource Planning
Trend Extensible Framework Trend Plug-in Pro
Business Intelligence

ERP Trend Enterprise Tecnologia

TDAINFORMATICA
Software House

SOLUZIONI GESTIONALI PER LE AZIENDE

SOFTWARE ERP TREND ENTERPRISE: LO STRUMENTO NEL FUTURO DELL'AZIENDA

QUESTO DOCUMENTO È UNA INTRODUZIONE, UNA PICCOLA GUIDA, ALLE TECNOLOGIE CON LE QUALI È STATO PENSATO, PROGETTATO E REALIZZATO L'APPLICATIVO GESTIONALE **ERP TREND ENTERPRISE**.



PREMESSA

Questo documento è un'introduzione o, se volete, una piccola guida alle tecnologie con cui è stato pensato, progettato e realizzato l'applicativo gestionale ERP Trend Enterprise. Poiché non tutti i lettori di questo documento sono esperti delle terminologie e delle più moderne tecniche di sviluppo software, abbiamo pensato di scriverlo in due forme: una chiara e comprensibile a tutti, una per esperti programmatori.

State per iniziare la lettura rivolta a tutti... ribaltate la brochure per leggere l'altra.

Lo scopo di questa guida è dimostrare che Trend Enterprise è progettato secondo le ultime tecnologie che, Microsoft e le comunità di sviluppatori, indicano come riferimento per gli anni a venire.

INTRODUZIONE

Trend Enterprise è un software client-server che opera in ambiente .NET, realizzato con una architettura di tipo n-tier o multi-tier, un aggettivo che può essere spiegato come "applicazione a enne-strati o livelli". Quali sono, dunque, gli strati che costituiscono un software multi-strato? Teoricamente sono tre (3-tier): Presentazione dei Dati, Accesso ed Elaborazione dei Dati e Sistema di Gestione dei Dati. Il primo è l'interfaccia utente, il secondo è il motore dell'applicazione che trasforma i dati, il terzo è il sistema che memorizza e gestisce i dati, comunemente chiamato database. I vantaggi della stratificazione sono dovuti al fatto che, per apportare modifiche alle procedure, è sufficiente modificare solo il livello interessato alle variazioni, il che rende l'applicazione molto flessibile e le parti del livello intermedio utilizzabili per più di una procedura. L'architettura di **Trend Enterprise è di tipo 4-tier.**



TREND ENTERPRISE E LA TECNOLOGIA .NET

IL FUTURO DEI SOFTWARE GESTIONALI

L'ARCHITETTURA 4-TIER E LA TECNOLOGIA .NET FANNO DI TREND ENTERPRISE UNO STRUMENTO IL CUI FUTURO È ALLINEATO ALLE INDICAZIONI DI MICROSOFT E DELLE COMUNITÀ DI SVILUPPATORI.

TREND ENTERPRISE E I SUOI COLLEGHI

Facciamo alcune considerazioni che ci permetteranno di distinguere Trend Enterprise dalla maggior parte degli altri software gestionali, attualmente reperibili nel mercato: la maggior parte di questi sono progettati con una architettura di tipo 2-tier, a due livelli. Solitamente, infatti, uno strato contiene sia le interfacce sia le logiche di elaborazione dei dati e un altro comprende i dati stessi. Si può affermare che quest'architettura, implicitamente, limita l'utilizzo del software o le sue capacità, ad esempio:

- 1- Un software 2-tier sarà un'applicazione desktop o per il web o per un altro tipo d'infrastruttura, ma non entrambe.
- 2- Il database, che contiene i dati, determina lo sviluppo dell'applicazione; ne consegue che non è possibile scegliere tra diversi "database".
- 3- I programmatori, quando sviluppano una nuova interfaccia, devono riscriverla insieme alle logiche per l'elaborazione dei dati e queste non saranno riutilizzabili.

Questo perché i suddetti software sono stati realizzati, probabilmente, con strumenti che risalgono agli anni passati e che non avevano le prerogative che avrebbero permesso ai programmatori di sviluppare diversamente. In questo caso tutti i produttori di software dovranno fare i conti con la politica di Microsoft che, in un periodo di circa 3 o 4 anni, non riproporrà la compatibilità dei nuovi sistemi operativi con questi strumenti. In caso diverso non avrebbe nessuna logica, pensando al futuro, scegliere questo tipo di architettura.



L' ARCHITETTURA 4-TIER DI TREND: TRAZIONE INTEGRALE

I COMPONENTI PRINCIPALI DELL' ARCHITETTURA DI TREND SONO QUATTRO: APPS, MODEL, ORM, RDBMS; SONO QUESTI GLI "STRATI" (TIER) DELL' APPLICAZIONE .NET



L' ARCHITETTURA DI TREND

Abbiamo detto che Trend Enterprise è un software client-server con architettura di tipo n-tier, e precisamente 4-tier; è stato realizzato in ambiente .NET con il linguaggio di programmazione C# (C-Sharp) di Microsoft. Lo schema successivo mostra i quattro strati e, per ognuno di essi, i contenuti tecnologici e "sottostrati" di cui si compongono. Come abbiamo anticipato Vi proponiamo due tipi di guida alla comprensione di questo schema, una molto più tecnica dell'altra: scegliete quella che meglio Vi si addice.

RDBMS, RELATIONAL DATABASE MANAGEMENT SYSTEM

Lo strato RDBMS è il più semplice da comprendere. Comunemente chiamato database, è identificato, spesso e in maniera semplicistica, con "l'Archivio dei Dati". Con le architetture di tipo 1, 2, 3-tier, i programmatori sono obbligati a sviluppare le logiche dei software in funzione del RDBMS che sarà utilizzato. L'architettura 4-tier, al contrario, permette l'indipendenza dell'applicazione dal RDBMS, poiché uno strato di cui parleremo in seguito, l'ORM, funge da "interprete" tra il database e il livello dell'applicazione stessa, detta CORE oppure MODEL. Ciò è molto importante, perché in questo modo Trend Enterprise diventa "RDBMS Indipendente" e, le aziende che lo adotteranno, potranno scegliere il database in funzione delle proprie esigenze, oppure, eventualmente, utilizzare quello di cui già dispongono.

In effetti, purché il RDBMS sia compatibile con le specifiche del mondo ADO.NET potrà essere utilizzato con il nostro applicativo. Si contano ormai oltre cinquanta "database", compatibili con ADO.NET, tra i quali sono compresi SQL, Oracle e MYSQL.



L' AMBIENTE MICROSOFT .NET: UNA GARANZIA PER IL FUTURO

LA TECNOLOGIA CHE MICROSOFT E LE COMUNITÀ DI SVILUPPATORI, INDICANO COME RIFERIMENTO PER GLI ANNI A VENIRE.

ORM, OBJECT-RELATIONAL MAPPING

Abbiamo già accennato alla funzione di "interprete" dello strato ORM tra il livello MODEL o CORE, e il RDBMS che determina l'indipendenza dell'applicazione .NET dal sistema che gestisce i dati. L'ORM di cui è dotato Trend Enterprise è ENTITY FRAMEWORK, un componente di .NET Framework 4.5 di Microsoft, cui abbiamo implementato TDEF, Trend Dynamics Entity Framework.

EF, ENTITY FRAMEWORK, rende disponibili al MODEL (il motore dell'applicazione) i dati contenuti nel database, secondo la tecnologia ADO.NET. La cosa più interessante è che i programmatori non devono conoscere quale database sarà utilizzato dall'applicazione; questo consente loro di scrivere molto meno codice per l'elaborazione dei dati in quanto l'accesso agli stessi è gestito dall'interprete. L'ORM converte le operazioni richieste dall'applicazione nei comandi specifici del RDBMS cui è connesso. È stato necessario implementare nell'ORM un sottostrato con funzioni specifiche quali, ad esempio, la gestione di architetture multiprocessore, l'ottimizzatore della velocità di accesso ai dati e una gestione avanzata del blocco dei record per una migliore amministrazione dei casi di concorrenza sui dati. Ulteriormente sono state implementate numerose librerie che permettono l'accesso dinamico ai dati.

MODEL O CORE

Lo strato CORE è il motore dell'applicazione Trend Enterprise. È un insieme di librerie che sono utilizzate dallo strato APPLICATION per esporre i dati. È necessario descrivere questo strato in modo meno semplicistico, considerandolo dal punto di vista degli sviluppatori: essi vedono il MODEL unito all'ORM come il supporto per creare le APPS di Trend Enterprise. Questa unione è TREND FRAMEWORK, TF.

TF, in pratica, è una collezione di comandi, routine, tools, controlli, classi e librerie che gli sviluppatori hanno a disposizione per implementare le APPS che, in definitiva, sono ciò che l'utilizzatore considera come il software vero e proprio. TEEF unisce le logiche generiche di .NET Framework 4.5 e del suo componente Entity Framework, a quelle specifiche, e necessarie per una applicazione gestionale, di Trend SDK e Trend Dinamic Entity Framework. La caratteristica più importante del MODEL è che il set di comandi, routine, controlli e classi che comprende, è disponibile per più di un'applicazione, è riutilizzabile. Questo permette ai programmatori di non riscrivere tali componenti per ogni applicazione, il che riduce il tempo di sviluppo e limita la possibilità di errori. Queste parti, inoltre, possono essere implementate e/o sostituite per accrescere la versatilità e flessibilità del framework senza che ne risentano gli altri strati.

APPLICATION

Lo strato APPLICATION è quello che l'utilizzatore considera come il vero e proprio software. In questo livello si trovano le interfacce grafiche, che si collegano ai componenti del MODEL, per l'esposizione dei risultati delle elaborazioni.

Due, soprattutto, sono le caratteristiche importanti di questo livello.

Una è che le APPS possono essere scritte in uno qualsiasi dei linguaggi che supportano la tecnologia .NET. Questo permette ai programmatori di costruire nuove applicazioni per gli utenti sfruttando le competenze di cui, eventualmente, già dispongono. Non meno importante, il fatto che sia possibile sviluppare applicazioni per infrastrutture diverse, non solo quindi applicazioni desktop per Windows, ma anche per il Web e per sistemi mobile senza perdere il supporto del framework di Trend.

