

InstantBDEExpress

**porting “trasparente” e “indolore” di
applicazioni BDE a DbExpress**



Panoramica su BDE

- Accesso a dbf e Paradox + accesso a DB SQL
 - Installazione "invasiva" e pesante
 - Accesso a MSSQL via DBLib (Libreria obsoleta)
 - Accesso a DB2 via CLI (Non supporta query prepared)
- Caching dei Dati (DataSet Bidirezionale)
 - Swap su disco
 - Impossibile disattivare il caching
- Cached Updates (Tramite TUpdateSQL) per bufferizzare le modifiche.
 - ApplyUpdates, CommitUpdates e CancelUpdates.
- Nato per database file oriented!



Panoramica su DBExpress

- Connessione solo a motori SQL
 - No Accesso a dbf e Paradox
 - Drivers leggeri e di terze parti
 - Accesso a MSSQL via OLEDb (meglio il driver di Corelab)
 - Accesso a DB2 (driver Borland)
 - Accesso a Oracle (meglio il driver di Corelab)
 - Accesso a Interbase e Firebird (Interbase e UIB)
- DataSet unidirezionale!
 - Uso dei ClientDataSet/Provider per fare caching
 - Cached Updates, Cancel Updates
- Nato per motori SQL e cross platform (Kylix)



Aspetti da considerare per le applicazioni BDE

- Dimensione e complessità dell'applicazione
 - Utilizzo di chiamate BDE a basso livello (Dbixxx)
 - Utilizzo di tabelle Paradox/Dbase
 - Utilizzo di TTable o TQuery
 - Utilizzo di CachedUpdates o live Query.
 - Logica applicativa File-oriented o SQL-oriented
-
- Tutti questi aspetti determinano il tasso di complessità del porting di una applicazione BDE.



Requisiti di pre-porting

- L'uso di TTable massivo si sposa male con i motori SQL
 - Problemi anche con BDE (es.cursori MS-SQL)
- Porting da TTable a TQuery dove possibile:
 - Sfruttare le join e non i campi di lookup
 - Utilizzare CachedUpdates e TUpdateSQL
 - TQuery.UpdateMode: meglio usare wherekeyonly
 - Non esiste più il concetto di locking preventivo!
- Attenzione alla mappatura dei campi:
 - ENABLEBCD varia il comportamento di BDE (consigliato)
- Non scaricare tutti i record lato client!
 - Una lista di troppi record è illeggibile
 - Lavora male sia il server che il client



Un porting “indolore”

- Obiettivo: Usare DBExpress simulando il comportamento del BDE
 - Per modificare il meno possibile le applicazioni BDE
 - Per poter risolvere i problemi a basso livello
 - Per “astrarsi” il più possibile dai diversi database SQL
- Realizzazione:
 - Componenti InstantBDEExpress alternativi a BDE che ne mantengono le caratteristiche di funzionamento
 - Utility di gestione degli alias InstantBDEExpress Administrator (simile a BDEAdministrator)
- Vantaggio:
 - Poter compilare alternativamente con BDE e DbExpress in fase di “porting” di una applicazione



Files del package

- IBDXDbTables.pas (equivalente a DbTables)
- IBDXFields.pas (classi di campi)
- IBDXMetadata.pas (classi per i metadati)
- InstantBDEExpress.dpk (package run-time)
- dclInstantBDEExpress.dpk (package design-time)
 - InstantBDEExpressReg.pas (registra i componenti)
 - IBDXDbTables.dcr (icone componenti)
 - IBDXMetadata.dcr (icone componenti)



L'InstantBDEExpress Administrator

- Interfaccia utente simile a BDEAdministrator
 - Registrazione degli Alias in file ini
 - È una estensione del file dbxconnections.ini + drivers.ini
 - Utilizzo del file drivers.ini solo come template
- Componente TBDXSQLMetadata
 - Mostra i metadati DbExpress in un dataset
 - E' un ClientDataSet con l'utilizzo interno di un TSQLDataSet che richiama SetSchemaInfo.
- Disponibile nel pannello di controllo di Window



Componente TBDXDatabase

- Derivato da TSQLConnection
 - Accesso al database tramite AliasName
 - Mappatura isolation-Level BDE->DbExpress
 - Ridefinizione di TTransIsolation come è prevista nel BDE
 - SessionName (non utilizzato)
 - Gestione della clessidra SQL
 - StoreDefs per compatibilità TDatabase
- Migliorie possibili:
 - SessionName per gestire transazioni diverse



Componente TBDXUpdateSQL

- Reimplementa TUpdateSQL
 - Implementa 3 TSQLQuery per fare le operazioni di insert, update e delete
 - Implementa gli stessi metodi BDE
- Migliorie possibili:
 - Un component-editor come quello BDE per creare automaticamente le istruzioni SQL



Componente (interno)

TBDXSQLResolver

- Reimplementa il modo di fare "update" da parte di un clientdataset
 - Si sostituisce al resolver standard
 - DoUpdate, DoDelete, DoInsert utilizzano l'UpdateSQL se disponibile altrimenti simula il comportamento "live-query" sfruttando il comportamento standard del provider.
- Viene istanziato da un provider di tipo TBDXDataSetProvider



Componente (interno)

TBDXDBDataSet

- Simula TDBDataSet + TBDEDataSet
 - È un ClientDataSet che utilizza internamente:
 - TBDXInternalSQLDataSet (TinternalSQLDataSet)
 - TBDXDataSetProvider
 - Utilizza TBDXDataLink (per la gestione del Master-Details automatico come nel BDE).
 - Rende compatibile il CDS al BDE:
 - property OnUpdateError, OnUpdateRecord, DatabaseName, SessionName, UpdateMode, UpdateObject, AfterApplyUpdate, BeforeApplyUpdates
 - Alcune proprietà non gestite:
 - Exclusive (non serve)
 - Constraints (poco utilizzati)

Componente TBDXQuery

- Simula il comportamento di TQuery
 - property DataSource, ParamCheck, SQL, Params.
 - RequestLive aggiorna subito lato server con cachedUpdate a false.
- Unidirectional fa caching come il BDE ;-)
- Master-Details automatico con uso di parametri nell'istruzione SQL
- Migliorie possibili:
 - Property-editor per DatabaseName
 - Component-editor tipo SQL-Builder
 - Component-editor per Execute Query



Componente TBSDXTable

- Simula il comportamento di TTable
 - Non utilizza internamente un TSQLTable ma sempre un TSQLQuery!
 - property DefaultIndex, IndexDefs, TableName
 - property MasterSource, MasterFields
 - property IndexFieldNames
 - (sono spariti IndexName e IndexFiles)
- Migliorie possibili:
 - Property-editor per DatabaseName
 - Property-editor per TableName



Altri componenti/classi

- TBDXStoredProc:
 - Simula il comportamento di TstoredProc
 - Testato poco!
- TBDXDBRegistry:
 - Gestione registrazione/deregistrazione Alias
 - Utilizzato dal InstantBDEExpress Administrator e dai componenti per recuperare dall'alias l'oggetto di connessione.



Esempio di Porting

- Preparazione al porting:
 - Se si usa Paradox o Dbase utilizzare TDataBase
 - Usare nomi "logici" per TTable
- Porting dell'esempio MastApp:
 - Questo esempio si presta bene al porting perché è già una applicazione orientata all'uso di Paradox e Interbase
 - Basta modificare le classi di accesso ai dati
 - Togliere dipendenze dall'oggetto globale Session del BDE
 - Problema dei campi "date-time"



Porting “progressivo”

- E' possibile affrontare il porting in modo progressivo.
 - Preparare l'applicazione all'uso di componenti “derivati”:
 - TCBTable, TCBQuery, TCBXXXField ecc...
 - Implementare libreria BDE con queste classi che non fanno nulla!
 - Compilare la versione BDE o DbExpress senza modificare l'applicazione ed effettuare Test in parallelo.
 - Attenzione all'uso di classi specifiche anche per i field:
 - Aiuta a risolvere incompatibilità tra i drivers DbExpress
 - Consente di implementare ulteriori feature



Consulenza

- Ethea fornisce le competenze e la consulenza al porting delle applicazioni.
 - Analisi dell'applicazione BDE
 - Formazione sull'uso dei componenti InstantBDEExpress
 - Supporto tecnico durante il periodo di porting
 - Analisi delle migliorie applicabili all'applicazione
 - Sviluppo su commessa dell'intero processo di porting dell'applicazione



Conclusioni

- InstantBDEExpress è una buona base per partire con il porting di applicazioni BDE
 - Considerare un pre-porting in logica SQL se si ha una applicazione "file-oriented".
 - Fatta l'applicazione sfruttare il componente TSQLQuery di DbExpress per elaborazioni finalmente unidirectional!
 - Sfruttare TCBTable e TCBQuery come ClientDataSet (es. InternalCalcFields, Aggregates, Clone, gestione più avanzata dei conflitti di update ecc...).
- InstantBDEExpress è una tecnologia trasparente, i sorgenti sono inclusi e l'assistenza è garantita.

