

Introduzione

Il TLMobile è nato come successore del TLR600 di cui ne ha ripreso la logica e il flow di funzionamento. Dal TLR600 ha acquisito la principale caratteristica di software di terminale perennemente online con la sua controparte server. A differenza del suo predecessore, che utilizzava un controllo proprietario della Pikolink, il TLMobile affida le comunicazioni tra terminale e il resto del modo a dei web service.

Successivamente sono state implementate altre modalità di funzionamento che vanno però ad integrarsi con la caratteristica principale del software che punta sempre a centralizzare le informazioni sul server.

Struttura

Il sistema TLMobile si compone da più di un software:

- **TLMobileConf**: sito di configurazione del Sistema. Sviluppato con Visual Studio 2010, è basato su Framework 2.0 per poter funzionare con l'UltiDev Standard (che non gestisce Framework 4).
- **TLPosWinWS**: web service il cui scopo è quello di rendere disponibili di dati presenti negli archivi del TLPosWin. Anch'esso come il TLMobileConf è sviluppato con Visual Studio 2010 e Framework 2. Una caratteristica importante è che questo web service deve funzionare sempre a 32bit (a causa dell'utilizzo del driver ODBC Visual Fox Pro che è solo a 32 bit). In caso debba funzionare su sistemi a 64 bit si deve utilizzare UltiDev Pro o IIS 7.5 che permettono agevolmente di configurare siti che funzionino solo a 32 bit.
- **TLWirelessTermWS**: è il web service di gestione di tutti gli elementi peculiari del sistema TLMobile (configurazioni, gestione documenti, esportazioni)
- **TLWTInitialLoad**: è una utility creata per velocizzare l'initial load del DB SQL Server Compact che deve essere copiato sul terminale. Nel tempo è divenuto un passaggio fondamentale del processo dell'installazione perché permette di stabilire altre configurazioni accessorie del terminale (presenti in file XML) e permette di effettuare un debug della struttura web service.
- **TLMobile**: è il software di interfaccia che gira su terminale. E' stato sviluppato con Visual Studio 2008 (ultima versione che permette di sviluppare con Compact Framework
- **BakGroundDBUpdater**: programma sviluppato sempre con Visual Studio 2008 per Compact Framework 3.5 che gira in background sul terminale e il cui compito è quello di mantenere allineati i dati del DB locale con quelli del gestionale (in caso il terminale debba attingere informazioni da questo DB) e di concentrare sul server eventuali collezionamenti memorizzati su DB Locale.

Modalità di funzionamento

Allo stato attuale dello sviluppo il sistema TLMobile può funzionare in 4 differenti modalità:

- **Auto**: il terminale punta a lavorare con i web service (acquisizione dati anagrafici e memorizzazione collezionamenti). Se il collegamento con il server viene a mancare, il TLMobile passa automaticamente in modalità "disconnessa" e utilizza un archivio locale come riferimento per le anagrafiche e per la memorizzazione dei collezionamenti. Quando il terminale passa in modalità "disconnessa", attiva una sequenza temporizzata che ogni 30 secondi verifica se è tornata la connettività con il server. Se la connettività viene ripristinata, il software automaticamente torna ad utilizzare i web service e demanda il trasferimento dei collezionamenti effettuati offline al BackGroundDBUpdater.
- **Batch**: il software di terminale lavora unicamente sull'archivio locale. L'aggiornamento delle anagrafiche e il trasferimento dei collezionamenti al server è demandato al BackGroundDBUpdater.
- **Web Service**: Il software di terminale lavorare unicamente tramite i web service. Se la connettività viene a mancare il software si blocca e segnala un errore.
- **Mista**: il software fa riferimento all'archivio presente sul terminale per le anagrafiche, mentre gestisce i collezionamenti come la modalità "Auto".

TLMobile

Il software è basato sostanzialmente su due classi:

- **TermCode:** in cui sono raggruppati tutti i metodi a cui il software fa riferimento (interfacciamento a web service, accesso a DB locale ed elaborazione)
- **SessionValues:** contiene una serie di variabili che definiscono lo stato del terminale.

TermCode

La classe può essere suddivisa in:

- Metodi che gestiscono l'accesso a web service: gestiscono la logica di istanziazione degli oggetti di accesso a web service.
 - Get_TLPosWinWS_Articles_Settings, Get_TLWirelessTerm_Conf_Settings, Get_TLWirelessTerm_Docs_Settings: funzioni utilizzate per leggere dal file XML i settaggi riguardanti il web service su cui sono verticalizzate.
 - Init_WirelessTerm_Conf, Init_WirelessTerm_Docs, Init_TLPosWinWS_Articles: funzioni in cui è implementata la logica di istanziazione\collegamento ai web service. In questi metodi è implementata la logica fondamentale di gestione dei metodi di funzionamento del terminale
 - Check_WirelessTermConf, Check_WirelessTermDocs, Check_TLPosWinWS_Articles: funzioni accessorie che simulano un accesso a web service per verificarne la presenza.
 - ChangeConnectivityStatus: questo metodo viene lanciato quando la connettività con i web service viene a mancare o viene ripristinata. Valorizza le variabili di stato del terminale e attiva\disattiva il timer che sovrintende al check dei passaggi di stato.
- Metodi di accesso base al DB locale di SQL Server Compact: sono i metodi base di accesso al DB. Sovrintendono l'esecuzione di stringhe di codice SQL, la creazione di oggetti datareader e popolamento di datatable.
 - OpenDBConn: restituisce una connessione aperta al DB locale
 - GetDataTable: popola una datatable in base al comando SQL che viene passato come parametro
 - ExecuteReader: restituisce un datareader agganciato agli elementi identificati dal comando SQL passato come parametro
 - ExecuteScalar: esegue un comando SQL che restituisce un unico risultato
 - ExecuteNonQuery: esegue un comando SQL che non produce risultati (viene utilizzato per fare insert, update o delete sulle tabelle del database).
- Metodi di accesso ai dati memorizzati nel DB Locale: sono i metodi che vengono chiamati quando debbono essere acquisiti\memorizzati dati dal DB locale (tutti i metodi iniziano con il prefisso "LocalDB_"). Per quanto possibile riproducono la medesima logica di acquisizione dati dei web service.
- Metodi legati all'identificazione utente e alle sue abilitazioni: sono metodi che sovrintendono l'identificazione utente (tramite web service o DB locale) e che rendono disponibili le abilitazioni utente
 - UserLogin: funzione utilizzata per gestire l'identificazione utente. Questo è un metodo ad alto livello che implementa la logica di discriminazione della fonte dati da utilizzare webservice\db locale.
 - UserPermissionCheck, UserFlagsCheck: sono i metodi che verificano se un utente è abilitato ad eseguire un'operazione sul terminale o meno.
- Metodi legati alla gestione delle informazioni articolo, fornitore, documenti e settaggi. Sono i metodi che sovrintendono alla gestione dei dati (acquisiti da web service o DB locale)

frmAcqInfo

È il form principale del programma. I terminali sono sistemi hardware con capacità elaborative "limitate" e i passaggi grafici tra diversi form mettono in crisi l'hardware (causando lentezza nelle transizioni tra differenti form). Conseguentemente si è dovuto applicare una tecnica di programmazione "vecchia" basata sui pannelli. Il flow di visualizzazione dei pannelli e le informazioni in esse visualizzati sono domandati dai metodi presenti in TermCode e dalla valorizzazione delle variabili di SessionValues.

BackgroundDBUpdater

La struttura è molto semplice. E' basata su una classe "Sincronizzatore" che contiene tutti i metodi di sincronizzazione dei dati del DB Locale. A seguito delle singole sincronizzazioni da TLPosWinWS verso DB Locale, per ridurre al minimo la quantità di dati scambiati con i web service, viene memorizzata in un file XML la "DataOraUPD" più recente di ogni singolo tipo di record. Viceversa eventuali collezionamenti presenti su DB locale vengono inviati al web service TLWirelessTermWS per essere memorizzati su server.