

Tibet4DExt

Un Plug-in 4th Dimension[®] per accedere
agli archivi di Tibet[®] in linea

(Versione 1.7.x)

**Manuale
Utente**

Capitolo 1, Informazioni Generali

Informazioni generali su Tibet4DExt

Il plug-in Tibet4DExt serve per acquisire dati dagli archivi di Tibet “in linea”, trasformando cioè una applicazione 4th Dimension in un “client” di una postazione principale di Tibet.

Questo avviene con l'introduzione di diverse funzioni esterne che si aggiungono al linguaggio di programmazione di 4D.

Perché sia possibile l'accesso ai dati, deve avvenire una operazione di “log in”, cioè l'accettazione da parte del server Tibet della postazione derivata rappresentata a tutti gli effetti dall'applicazione 4D; questo comporta una fase di partenza in cui in generale sarà necessario fornire all'applicazione l'indirizzo del server, un codice utente, una parola chiave e un codice di attivazione (fornito da Easy Byte al momento dell'acquisto).

Terminato l'accesso ai dati, oppure quando si ritenga di dover accedere ad un nuovo server, occorre effettuare l'operazione opposta di “log out”, cioè il rilascio delle risorse impegnate per il collegamento.



Con questa versione è supportato solamente il collegamento via TCP/IP con il server.



Nel seguito ci riferirà al server Tibet anche come “postazione principale” o semplicemente “server”

Richieste e Installazione.

Per usare Tibet4DExt occorre avere:

- 4th Dimension©
- Un Macintosh PowerPC con MacOS 8.6 o superiore, e 4Mb liberi per il plug-in.

Inoltre nel server Tibet deve essere presente ed attivato il modulo 'X' per le estensioni.



In questa versione il Plug-in funziona solo su PowerPC

Per installare Tibet4DExt, copiare il plug-in fornito nella cartellina "MAC4DX" che dovrà essere presente nella stessa cartellina del documento di 4D contenente i dati e l'applicazione. Attenzione a non copiare, con nomi differenti, due copie dello stesso plug-in, per evitare errori inaspettati e malfunzionamenti!

Il collegamento con il server

Tibet4DExt, per funzionare correttamente, necessita di essere aperto prima di ogni altra operazione (TPLIOpenPlugin), e chiuso prima di uscire dal programma (TPLIClosePlugin). Login e Logout

Una volta aperto il Plug-in, prima di poter accedere ai dati di un'azienda, si deve eseguire il "login", cioè dichiarare la propria identità al server.

Il login viene effettuato tramite la chiamata TPLILogIn che prevede il passaggio di alcuni parametri:

- (a) Il numero IP della postazione principale (Server Tibet)
- (b) Un codice utente valido
- (c) La "Password" di accesso del suddetto utente
- (d) Un codice di attivazione.

Un utente Tibet già dovrebbe conoscere il significato di (b) e (c); nel dubbio, rivolgersi all'amministratore del sistema. Se sulla postazione principale sono stati disattivati i controlli sugli accessi, questi parametri possono essere lasciati vuoti.

Come dovrebbe essere noto, il codice di attivazione (d) è fornito da Easy Byte; può essere usato un codice di attivazione esplicitamente previsto per le estensioni, oppure un codice per una postazione derivata "normale". In quest'ultimo caso però, una postazione derivata con lo stesso codice non può essere in funzione contemporaneamente all'estensione, ovvero quando è attivo uno non sarà possibile usare l'altra, e viceversa.

Il numero IP del server Tibet (a) necessita di una spiegazione particolare: esso deve essere nel formato standard di un nodo IP, come per esempio "127.0.0.1", ovvero quattro numeri separati da punti.

Il valore da immettere in questo campo può essere richiesto all'amministratore del sistema, oppure ricavandolo dal Pannello di Controllo TCP/IP sulla macchina sui cui è attivo il server Tibet, nel campo chiamato "Indirizzo IP".



Nel caso (comunque scoraggiato) in cui più server Tibet siano attivi sulla stessa macchina, l'unico che può essere usato per le estensioni è il primo che è stato fatto eseguire. Gli altri non possono essere "visti"!

per terminare il collegamento, è necessario chiamare la funzione LogOut:

```
err := TPLILogOut()
```

Se il logout non viene effettuato prima di chiudere l'applicazione, il server considererà l'applicazione come chiusa irregolarmente.

Tra il login e il logout potranno essere effettuate tutte le operazioni di lettura e scrittura desiderate.

Capitolo II, Le funzioni per il programmatore

Le funzioni a disposizione del programmatore

Le funzioni ad uso del programmatore sono 17:

```
TPLIOpenPlugin  
TPLIClosePlugin  
TPLIListaServer  
TPLILogin  
TPLILogout  
TPLIReadRecord  
TPLIReadField  
TPLIUpdateRecord  
TPLIUpdateField  
TPLIGetFieldNames  
TPLIStartScan  
TPLIEndScan  
TPLIScanNext  
TPLIScanLocate*  
TPLIIPAddress2String  
TPLIString2IPAddress  
TPLIMakeIPAddress  
TPLIVersion  
TPLILockRecord*  
TPLIUnlockRecord*  
TPLIGetNewRecord*  
TPLIAddMasterDetail*
```

* funzioni disponibili solo dalla versione 1.2.4

TPLIOpenPlugin, apertura del plug-in

Tibet4Dext necessita di essere aperto prima di ogni altra operazione, chiamando questa funzione viene allocata la memoria necessaria per il funzionamento del plug-in e vengono inizializzati alcuni parametri interni.

```
err := TPLIOpenPlugin()
```

TPLIClosePlugin, chiusura del plug-in

Questa funzione dovrà essere chiamata alla chiusura dell'applicazione o comunque quando l'utente ha terminato di utilizzare le funzioni del plug-in. TPLIClosePlugin rilascia la memoria allocata all'apertura del plug-in dalla funzione TPLIOpenPlugin.

```
err := TPLIClosePlugin()
```

TPLIListServer, elenco dei server Tibet

Questa funzione deve essere utilizzata quando non si conosca l'indirizzo IP del server o dei server Tibet presenti.

```
err := TPLIListaServer(nome; protocollo; indirizzo; porta)
```

Questa funzione riempie gli array "nome", "protocol", "address", "port" con i valori di tutti i server Tibet presenti nella rete locale. Se sono presenti dei server Tibet, il risultato della funzione è maggiore di zero ed è il numero dei server che sono stati trovati. Se il valore tornato da questa funzione è uguale a zero, significa che non sono presenti server Tibet sulla rete locale. Un valore negativo significa che si è verificato un errore. I valori di protocollo, indirizzo e porta saranno necessari per eseguire il login sul server.

TPLILogin, connessione ad un server

Esegue il login sul server specificato nei parametri. E' necessario indicare una chiave di attivazione, un nome utente e una parola chiave validi.

```
err := TPLILogin(chiaveDiAttivazione; utente; password;
                  protocollo; indirizzo; porta)
```

Se il valore tornato è uguale o maggiore di zero, la connessione è andata a buon fine, altrimenti il valore tornato contiene l'errore.

TPLILogout, sconnessione da un server

```
err := TPLILogout()
```

Esegue il logout dal server attivo. Il risultato tornato, se negativo, contiene un errore.

TPLIReadRecord, lettura di un record

Per ottimizzare i tempi di accesso, Tibet4DExt suddivide l'accesso in due fasi: lettura del "record" e lettura dei campi; l'accesso alla rete avviene solo nella prima fase, facendo sì che la lettura dei singoli campi possa essere comodamente e velocemente eseguita localmente.

```
err := TPLIReadRecord (nrazienda, nomedb; nrchiave; chiave; record)
```

i parametri hanno il seguente significato:

nrazienda: quale azienda leggere (normalmente 1);

nomedb: quale archivio leggere;
nrchiave: quale chiave usare;
chiave: il valore della chiave;
record: Una variabile 4D in cui leggere il record; sarà una variabile TEXT.

L'archivio da cui leggere (nomedb) è una stringa di quattro caratteri: per esempio 'DBCF' indica l'archivio 'Clienti-Fornitori'. L'elenco completo compare nel manuale "Informazioni Tecniche sugli Archivi di Tibet" che accompagna questa documentazione.

Ogni archivio può essere acceduto tramite una o più chiavi (nrchiave); quale chiave usare è indicato tramite un numero a partire da 1: per esempio il codice di un cliente o fornitore è la chiave primaria di accesso all'archivio, ed ha il numero 1. L'elenco completo delle chiavi per ogni archivio e i relativi numeri è riportato nel manuale "Informazioni Tecniche sugli Archivi di Tibet" già citato.

Il valore della chiave è normalmente una stringa, e corrisponde al valore della chiave da cercare per accedere al relativo record.

Infine, il record letto sarà depositato in una variabile di 4D di tipo TEXT (record) per la successiva lettura dei campi del record.

TPLIReadField, lettura di un campo

```
err:=TPLIReadField (record; nome; contenuto)
```

Come detto, l'accesso ai campi si effettua tramite una variabile TEXT di 4D in cui sia stato preventivamente letto un record.

Per ricavare un campo, che è sempre convertito in stringa (contenuto), occorre passare il nome del campo richiesto e la variabile di 4D in cui è stato letto il record.

Il nome del campo è, nel caso più semplice, una stringa di quattro caratteri; Tibet però supporta sottocampi e vettori, per cui il nome può diventare una sequenza di nomi di quattro caratteri, separata da punti e con la possibile indicazione di elementi di vettori come un numero a partire da 1 racchiuso fra parentesi quadre. Possibili esempi sono:

CFDE (Il campo descrizione nel record Clienti-Fornitori)

CFPC[1].DARE (Il sottocampo DARE del primo elemento del vettore CFPC)

Come al solito, valori e significati dei nomi dei campi sono riportati nel manuale "Informazioni Tecniche sugli Archivi di Tibet".

TPLIUpdateRecord, scrittura di un record

err:=TPLIUpdateRecord (nrazienda, record)

i parametri hanno il seguente significato:

nrazienda: quale azienda aggiornare (normalmente 1);

record: Una variabile 4D in cui sia presente un record (TEXT).

Il record può essere stato riempito da una "TPLIReadRecord" o da una "TPLIGetNewRecord", eventualmente seguite da chiamate alla "TPLIUpdateField" per assegnare un valore ai vari campi. Questa funzione aggiorna il record nell'archivio di Tibet: se il record deriva da una precedente chiamata "TPLIGetNewRecord" si tratterà di una aggiunta altrimenti, nel caso che derivi da una "TPLIReadRecord", verrà modificato il record originale. Sebbene questa funzione esegua automaticamente un lock del record prima di scrivere delle modifiche, è auspicabile che, nel caso di modifica, venga eseguita una chiamata a "TPLILockRecord" immediatamente dopo aver letto il record da modificare, quindi rileggere il record dopo la lock e, di conseguenza eseguire una chiamata a "TPLIUnlockRecord" dopo avere eseguito l'update.

TPLIUpdateField, scrittura di un campo

err:=TPLIUpdateField (record; nome; contenuto)

Questa funzione permette di modificare il contenuto del campo nome all'interno di un record valido.

Per aggiornare un campo occorre passare il nome del campo richiesto, la variabile di 4D in cui è stato letto il record e una stringa che contenga il valore da assegnare (contenuto).

Il nome del campo è, nel caso più semplice, una stringa di quattro caratteri; Tibet però supporta sottocampi e vettori, per cui il nome può diventare una sequenza di nomi di quattro caratteri, separata da punti e con la possibile indicazione di elementi di vettori come un numero a partire da 1 racchiuso fra parentesi quadre. Possibili esempi sono:

CFDE (Il campo descrizione nel record Clienti-Fornitori)

CFPC[1].DARE (Il sottocampo DARE del primo elemento del vettore CFPC)

Come al solito, valori e significati dei nomi dei campi sono riportati nel manuale "Informazioni Tecniche sugli Archivi di Tibet".

TPLIGetFieldNames, per ottenere la lista dei campi

err:=TPLIGetFieldNames (record; nomi; db_offsets)

Questa funzione riempie l'array di stringhe nomi con la lista dei nomi dei campi del record. Oltre ai nomi dei campi, verrà compilato anche l'array db_offsets con gli

offset corrispondenti ai singoli campi; questa informazione potrebbe essere utile per accedere ai campi direttamente tramite la variabile TEXT, questa tecnica è comunque sconsigliata: per accedere ai campi è supportata solo la chiamata "TPLIReadField".

**TPLIStartScan,
TPLIEndScan,
TPLIScanNext
TPLIScanLocate, per scorrere il contenuto di un archivio**

Le funzioni di scansione permettono di scorrere i record di un archivio sequenzialmente, partendo dal primo o ricercando direttamente il record da cui iniziare. Per iniziare la scansione dei record è necessario eseguire una chiamata alla funzione "TPLIStartScan" che legge anche il primo record.

```
err:=TPLIStartScan (nrazienda, nomedb; nrchiave; record)
```

nrazienda: quale azienda leggere (normalmente 1);

nomedb: quale archivio leggere;

nrchiave: quale chiave usare per la scansione (vedere la "TPLIReadRecord");

record: Una variabile 4D in cui leggere il primo record; sarà una variabile TEXT.

Per leggere il record successivo si deve utilizzare la funzione "TPLIScanNext" passandogli lo stesso record riempito dalla " TPLIStartScan "

```
err:=TPLIScanNext (record)
```

Per leggere un record specifico si può utilizzare la " TPLIScanLocate " che legge il primo record corrispondente alla chiave specificata o, se la chiave non esiste, quello subito successivo.

```
err:=TPLIScanLocate (record, chiave)
```

Il valore della chiave è normalmente una stringa, e corrisponde al valore della chiave da cercare per accedere al relativo record.

Per terminare la sessione di scansione dei record si deve utilizzare la "TPLIEndScan" passandogli l'ultimo record valido letto all'interno della scansione.

```
err:=TPLIEndScan (record)
```

TPLIIPAddress2String, per convertire un indirizzo IP in stringa

```
err:= TPLIIPAddress2String (ipnum; ipstr)
```

Dato un indirizzo IP in forma di numero (ipnum) lo converte in stringa.

TPLIString2IPAddress, per convertire una stringa in indirizzo IP

```
err:= TPLIString2IPAddress (ipstr; ipnum)
```

Data una stringa che contiene un indirizzo IP, la converte in forma binaria, nel numero ipnum.

TPLIMakeIPAddress, per creare un indirizzo IP

```
TPLIMakeIPAddress (ipnum;194;183;6;252)
```

Dati i quattro componenti di un indirizzo IP, questa funzione li converte in un numero binario unico (ipnum).

TPLIVersion, versione del plug-in

Si può ricavare la versione del plug-in richiamando questa funzione:

```
Versionstr := TPLIVersion( )
```

dà come risultato la stringa (o una corrispondente): "Tibet4DExt v1.0 PPC".

TPLILockRecord

TPLIUnlockRecord

```
err := TPLILockRecord(nrazienda, record)  
err := TPLIUnlockRecord (nrazienda, record)
```

Queste funzioni permettono di eseguire nell'azienda nrazienda lock e unlock del record passato come parametro (il record deve essere valido: letto da una "TPLIReadRecord" o da una scansione).

TPLIGetNewRecord, per ottenere un nuovo record vuoto

```
err := TPLIGetNewRecord (nomedb; record)
```

Questa funzione crea un nuovo record vuoto per l'archivio nomedb e lo assegna alla variabile record (TEXT). Essa dovrà essere utilizzata per preparare il contenuto di un record che si desidera aggiungere all'archivio di Tibet, normalmente sarà seguita da una serie di chiamate a "TPLIUpdateField" e quindi da una "TPLIUpdateRecord" per scrivere il nuovo record.

TPLIAddMasterDetail, per aggiungere una scheda di tipo master/detail

```
err := TPLIAddMasterDetail(nrazienda;nomedb; var;  
mrecord, drecords, numdetails)
```

nrazienda: quale azienda (normalmente 1);
nomedb: quale archivio master;
var: variant (normalmente 0 [zero]);
mrecord: il record master (TEXT);
drecords: i record detail (TEXT ARRAY);
numdetails: il numero dei record detail.

Questa funzione crea una nuova scheda di tipo master/detail nell'archivio Tibet (ad esempio una registrazione di prima nota o una partita aperta). Per preparare il contenuto dei record, normalmente essa sarà preceduta da almeno due chiamate a "TPLIGetNewRecord" (una per il master e almeno una per i detail), e quindi una serie di chiamate a "TPLIUpdateField" per compilare i vari record.

Dalla versione 1.7.0 e successive

Il risultato della funzione è minore di zero in caso di errore, altrimenti il risultato (positivo) corrisponde al numero interno assegnato alla registrazione da Tibet.

Appendice A, Codici di Errore

I codici sono i seguenti:

-1	Mancanza di memoria, aumentare la memoria per 4D
-2	Parametro illegale, rivedere il formato dei parametri
-3	Login impossibile
-4	Autenticazione fallita, rivedere il codice utente e/o la Password
-5	Codice di archivio errato
-6	Il server è sconnesso
-7	Memoria insufficiente, aumentare la memoria per 4D
-8	Chiave troppo lunga
-9	Manca la descrizione dell'archivio
-10	Indirizzo IP scorretto
-11	Campo non esistente
-12	Conversione scorretta
-13	Errore interno (manca Sy2Image)
-14	Errore interno (dimensione del record)
-15	Errore nella scrittura
-16	Errore nello scan
-17	Mancano i servizi esterni
-18	Versione scorretta
-19	Numero di chiave scorretto
-20	Server non trovato
-21	Campo illegale
-22	Azienda illegale
-23	Nome archivio master errato
-24	Nome archivio detail errato
-25	Sono stati passati troppo pochi elementi per i detail
-26	Impossibile eseguire il lock del record (record in uso)
-27	La versione del server è incompatibile con le extensions
-28	Errore interno di memoria
-29	Errore nel caricamento della libreria
-30	Errore interno di memoria
-31	Errore interno di memoria
-32	Errore interno di memoria
-33	Errore durante l'allocazione di una nuova riga SQL
-34	Numero di riga non valido
-35	La versione della extensions lib è incompatibile
-36	mgacli.dll non trovata

Per i codici di errore che non sono presenti in questa lista, occorre fare riferimento agli errori che possono essere generati dal server Tibet:

< <http://www.easybyte.it/tech2.php> >