

Guida all'uso di FuzzyReport

Versione 2.0 in Italiano

Sommario

Sommario.....	2
Introduzione.....	3
Caratteristiche.....	4
Installazione.....	5
Nuovi componenti.....	7
Nuovi ambienti.....	8
Concetti base.....	10
Come iniziare.....	11
Le bande.....	12
Primi passi.....	15
I livelli e le relazioni master-detail.....	23
I gruppi.....	24
Le tabelle.....	25
I sottoreport.....	26
Report composti.....	27
Adattamento testo e dintorni.....	28
Approfondimenti sulle sorgenti di dati.....	30
Programmiamo una sorgente di dati.....	32
Report non da database.....	34
Funzioni ed operatori.....	36
Formattazione ed evidenziatura.....	38
L'editor di espressioni.....	40
Opzioni report e pagina.....	42
I metodi del report.....	45
Gli eventi del report.....	47
Le proprietà del report.....	49
I metodi della classe base.....	52
Le proprietà della classe base.....	53
Oggetti standard.....	54
Oggetti avanzati.....	56
Struttura delle pagine del report.....	60
Stralci di codice sorgente.....	61
Creare oggetti a run-time.....	65
Convertire QuickReport.....	66
Licenza d'uso.....	67

Introduzione

L'idea alla base di FuzzyReport, era quella di creare un report che sopperisse alle lacune di QuickReport, aumentandone i pregi. Che fosse semplice, flessibile nell'uso e liberamente distribuibile (sorgenti inclusi) all'intera utenza Delphi.

La possibilità pratica arrivò nel luglio del 1999, quando navigando in rete trovai la versione ufficiale 2.2 di FastReport freeware.

Da quella versione ufficiale si distaccò FuzzyReport, che continuò così a crescere indipendentemente.

La data del rilascio della prima versione risale al 11/08/1999.

Da allora FuzzyReport ha compiuto passi da gigante, proponendosi oggi come valida alternativa non solo a QuickReport, ma a molti strumenti di reporting di tipo commerciale. Diventando il primo report in italiano, interamente freeware e con i sorgenti inclusi, disponibile per l'intera comunità Delphi.

Inoltre dalla versione 2.0, con un formato di pagina compatibile con QuickReport, un componente di conversione automatica, e dei nuovi metodi e proprietà, si è riusciti a rendere il porting (conversione da QuickReport), un'operazione quasi indolore.

Caratteristiche

- Modellatore di report incorporato (possibilità di modificare i report a run-time)
- Anteprima alla MS Word
- Autocomposizione di nuovi report
- Possibilità di gestire variabili/categorie interne al report
- Velocità (40-60% più veloce di QuickReport 2 e 3)
- Compattezza del codice
- Predisposto al multilinguaggio
- Numero illimitato di pagine nei report preparati
- Modellazione ed anteprima in tempo reale
- Modelli di report
- 3 livelli di dettagli (Master/Dettaglio/SottoDettaglio)
- Sottoreport
- Gruppi
- Report a tabella
- Report a pagine multiple
- Report composti alla QRCompositeReport
- Controllo completo della stampa
- Supporto di funzioni matematiche
- Espandibile tramite nuovi componenti, autocomposizioni, librerie di funzioni, ecc.
- Filtri di esportazione (ora disponibili quello testuale, quello RTF – Word compatibile e quello in formato grafico EMF)
- Report a due fasi
- Ricerca testuale nei report preparati, con lampeggiamento del testo trovato
- Ridimensionamento delle pagine del report (Zoom sia in larghezza che in altezza)
- Elevata qualità di stampa, ed anteprima
- Gestione codici a barre
- Completa gestione dell'Euro
- Formato pagina compatibile con QuickReport
- Tre diverse unità di misura selezionabili nel modellatore (con righello)
- Nuovi metodi e proprietà per aumentare la compatibilità con QuickReport
- Formato pagina compatibile con QuickReport
- Nuovo componente per la conversione automatica di report "QuickReport"
- Nuovo componente per la gestione di grafici professionali

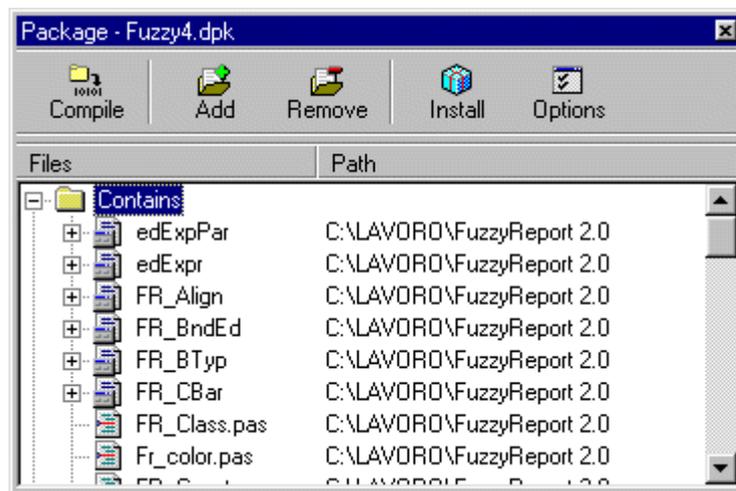
Installazione

L'installazione di FuzzyReport è la stessa per Delphi 3/4/5, l'unica differenza si rileva nel caso dell'installazione sotto Delphi 2.

E' importante eliminare ogni traccia di vecchie versioni di FuzzyReport, prima di procedere con l'installazione, dato che la sovrascrittura di una vecchia versione potrebbe causare seri problemi di compatibilità.

Nel caso d'installazione sotto Delphi 2, dal menù "Component" selezionare la voce "Install...", apparirà una finestra dal titolo "Install Components", cliccare sul tasto "Add...". Apparirà una finestra dal titolo "Add Module", cliccare sul tasto "Browse...". Apparirà un'altra finestra, dalla quale bisognerà selezionare il file "FR_Reg.pas" e poi cliccare su "Apri". Tornata in primo piano, la finestra dal titolo "Install Components", cliccare su "OK".

Se invece l'installazione si sta' effettuando in Delphi 3/4/5, si seguano i seguenti passi...



Dal menù "File" attivare il comando "Open" e dalla cartella in qui è stato copiato FuzzyReport, selezionare il file "FuzzyX" dove per "X" si intende il numero di versione di Delphi utilizzato. Apparirà una finestra come questa:

Nota importante: se si usa QuickReport Pro, si rimuova, dalla sezione "Required", la riga "QrptXX" e si inserisca nei percorsi di ricerca di Delphi la cartella di QuickReport Pro.

Per installare FuzzyReport, premere in sequenza i tasti "Compile" ed "Install".

Terminata questa procedura apparirà una finestra che comunicherà l'avvenuta installazione dei nuovi componenti. Una finestra molto simile alla seguente:

Nuovi componenti

Analizziamo adesso brevemente tutti i nuovi componenti che troviamo nella palette di FuzzyReport.

	Nome	Descrizione
	TfrReport	Componente principale. Basta posizionarlo all'interno di una form e fargli doppio clic per visualizzare il modellatore (piano di lavoro del report) ed incominciare a lavorare.
	TfrCompositeReport	Permette di unire più report in uno unico.
	TfrUserDataset	Permette di collegare vettori, file di testo, ed altre sorgenti di dati, ai report in uso, e di modificarne le caratteristiche di collegamento a run-time.
	TfrDBDataset	Permette di collegare database, ai report in uso, e di modificarne le caratteristiche di collegamento a run-time.
	TfrRichObject	Permette di inserire all'interno di un report oggetti di tipo "RichText"
	TfrDesigner	Permette di inserire all'interno di un programma, la possibilità di modificare a run-time i report disponibili.
	TfrShapeObject	Permette di inserire all'interno di un report oggetti forme geometriche varie.
	TfrCodeBarreObject	Permette di inserire all'interno di un report codici a barre.
	TfrGraphObject	Permette di inserire all'interno di un report grafici professionali.
	TfrOLEObject	Permette di inserire all'interno di un report oggetti OLE (Word, Excel, Power Point, Corel Draw, Paint Shop, ecc...)
	TfrFramedMemoObject	Permette di inserire all'interno di un report dei rettangoli di testo con dei bordi altamente personalizzabili.
	TQRtoFRConverter	Permette di convertire il report "QuickReport" sulla quale è posizionato, in un report "FuzzyReport".
	TfrExportTextObject	Inserito all'interno di un progetto permette di effettuare l'esportazione del report corrente in formato testo.
	TfrExportRichObject	Inserito all'interno di un progetto permette di effettuare l'esportazione del report corrente in formato RTF (Word compatibile).

In seguito ognuno di questi componenti verrà più accuratamente trattato per meglio spiegarne tutti i possibili utilizzi.

Nuovi ambienti

Analizziamo adesso i nuovi ambienti con cui dovremo incominciare a prendere familiarità per lavorare con FuzzyReport. Essi sono principalmente due. Il modellatore e l'anteprima di stampa.

Il primo è un ambiente che permette di costruire e modificare i report, mentre il secondo invece ne permette la visualizzazione a video e l'eventuale stampa.

E' ovvio sottolineare che questa brevissima descrizione non rende giustizia a questi due completi strumenti di sviluppo. Ma il nostro momentaneo intento è solo quello di far prendere familiarità con gli ambienti principali, per poi approfondirne l'utilizzo nelle seguenti apposite sezioni.

Analizziamo adesso come si propongono visivamente questi due nuovi strumenti.

Il modellatore, a cui si accede facendo doppio clic su di un componente di tipo "TfrReport", appare come una finestra del tutto simile alla seguente:

Da questa finestra è possibile seguire tutte le fasi della vita dei report. Dalla creazione alla stampa. E' possibile modificare tutte le opzioni, ed impostare ogni singolo parametro. Spostare componenti, aggiungerne di nuovi ed eliminarne di vecchi. Impostare il formato del foglio ed il verso di stampa. Ingrandire, in larghezza o altezza, le pagine da stampare, e molto, molto altro, ancora.

Sempre dalla finestra del modellatore è possibile avviare l'anteprima di stampa, ossia una visione a schermo pieno, di come sarà la stampa su carta. La finestra dell'anteprima sarà simile alla seguente immagine:

Le funzioni disponibili in questo ambiente vanno dallo scorrimento di tutte le pagine stampate, all'adattamento della dimensione per meglio adattarle alla visualizzazione a schermo, alla ricerca testuale di parti di testo, al salvataggio/caricamento del report generato, all'esportazione dello stesso in formato testo.

Concetti base

Un report non è altro che uno strumento che interfacciato ad una serie di dati (provenienti da database o da altra fonte), ne permette la visualizzazione a video e la stampa su carta, nel modo più flessibile, semplice e professionale, possibile.

Esistono diversi tipi di report in commercio. Ognuno di essi adotta una sua filosofia per la gestione delle informazioni e delle modalità di visualizzazione.

FuzzyReport è un report di tipo "a banda". Ossia per elaborare e visualizzare i dati, utilizza delle bande specifiche, ognuna con un determinato compito.

Ci sarà una banda che verrà visualizzata all'inizio di ogni pagina, una alla fine, un'altra all'inizio di ogni report (prima pagina di tutte quelle generate), una alla fine (ultima pagina).

All'interno di ogni banda, sarà possibile inserire dei diversi tipi di oggetti. Ognuno dei quali permetterà di personalizzare in modo diverso le informazioni da visualizzare.

Quindi riassumendo possiamo dire che mentre gli oggetti permettono di scegliere il modo migliore di visualizzare un dato, le bande ne determinano la posizione.

Un'altra importante caratteristica di FuzzyReport, è quella di essere capace di associare dei nomi di variabili a dei campi di una tabella, a dei valori di sistema, a dei valori attribuibili via codice, o a combinazioni di questi. Per comodità d'uso, le variabili possono poi essere a loro volta suddivise in categorie.

Un esempio di valori di sistema è la variabile [DATE] che restituisce automaticamente il valore della data odierna.

L'utilizzo di variabili e categorie, rende l'uso del modellatore, da parte dell'utente finale, più semplice ed intuitivo. Non obbligandolo ad avere a che fare direttamente con astrusi nomi di campi di tabelle.

Come iniziare

La prima cosa da fare prima di incominciare a progettare un report è quella di decidere come suddividerlo. Ossia quali tipi di bande utilizzare.

I tipi di bande a disposizione sono 21. Alcuni tipi è possibile utilizzarli più volte all'interno dello stesso report, altri una sola volta. Questi ultimi sono normalmente (ma non sempre) quelli che indicano intestazioni o piè di pagina (inizio e fine), dato che non avrebbe senso inserire due diverse intestazioni. Sarebbe sufficiente inserire tutto il dovuto nell'unica presente.

Una volta deciso con che bande lavorare, bisognerà incominciare ad inserire in esse gli oggetti che visualizzeranno i singoli dati.

Come per le bande, una volta inseriti gli oggetti, bisognerà comunicare ad ognuno di essi che informazioni visualizzare.

Questo sarà possibile farlo inserendo in ogni oggetto un'espressione. La funzione delle espressioni è proprio quella di restituire un valore dopo averne fatto un'elaborazione. Ciò significa che oltre con esse, oltre ad indicare ad un oggetto che informazioni visualizzare, sarà anche possibile indicargli il modo di elaborarle.

Per esempio, avendo a disposizione la quantità di un articolo ed il suo prezzo singolo, un'espressione potrà indicare ad un oggetto di visualizzare il prezzo complessivo di tutti gli articoli, semplicemente moltiplicando la quantità per il prezzo singolo.

Le bande

Ecco di seguito elencati i 21 diversi tipi di bande, disponibili nella versione corrente di FuzzyReport.

Tipo di banda	Nome della banda	Descrizione
Intestazione	Titolo report	Viene stampata una volta ad inizio report
Piè di pagina	Piè di pagina report	Viene stampata una volta a fine report
Intestazione	Intestazione pagina	Viene stampata all'inizio di ogni pagina
Piè di pagina	Piè di pagina	Viene stampata alla fine di ogni pagina
Intestazione	Intestazione principale	Viene stampata all'inizio di ogni 1° livello
Dati	Dati principali	Dati del 1° livello
Piè di pagina	Piè di pagina principale	Viene stampata alla fine del 1° livello
Intestazione	Intestazione dettagli	Viene stampata all'inizio del 2° livello
Dati	Dati dettagli	Dati del 2° livello
Piè di pagina	Piè di pagina dettagli	Viene stampata alla fine del 2° livello
Intestazione	Intestazione sottodettagli	Viene stampata all'inizio del 3° livello
Dati	Dati sottodettagli	Dati del 3° livello
Piè di pagina	Piè di pagina sottodettagli	Viene stampata alla fine del 3° livello
Sfondo	Copertura	Viene stampata come sfondo della pagina
Dati	Intestazione colonna	Viene stampata all'inizio di ogni colonna
Piè di pagina	Piè di pagina colonna	Viene stampata alla fine di ogni colonna
Intestazione	Intestazione gruppo	Viene stampata all'inizio di ogni gruppo
Piè di pagina	Piè di pagina gruppo	Viene stampata alla fine di ogni gruppo
Intestazione	Intestazione Tabella	Viene stampata all'inizio della tabella
Dati	Dati Tabella	Dati Tabella
Piè di pagina	Piè di pagina tabella	Viene stampata alla fine della tabella

Come si può facilmente constatare esistono tre diversi tipi base di bande. Le bande "Intestazione", quelle "Piè di pagina" ed infine le bande "Dati".

I primi due tipi sono molto simili. Servono a stampare rispettivamente all'inizio e alla fine di una sezione (pagina, report, gruppo o livello). Al contrario le bande di tipo "Dati" determinano la sorgente dei dati da elaborare. Tramite esse è possibile selezionare il database (o altra fonte come vettori, file di testo, ecc...) dalla quale attingere i dati. Queste bande vengono stampate sequenzialmente tante volte quanti sono gli elementi della sorgente di dati selezionata.

Per rendere il concetto più chiaro procediamo con un esempio pratico. Ipotizziamo di dovere stampare una lista di clienti. Il modo migliore di proporre una stampa simile risulterebbe con molta probabilità quello di dividere ogni pagina in tre parti. Una superiore contenente l'intestazione con i nomi di ogni colonna (Ragione sociale, indirizzo, città, telefono, ecc...), una centrale contenente i dati veri e propri, ed una nota a piè di pagina in basso, che contenga il numero della pagina corrente in relazione a quelle totali.

Graficamente un esempio del genere verrebbe molto simile all'immagine seguente:

In questo esempio, si è fatto uso di tutti e tre i tipi di banda. Quella "Intestazione", quella "Dati", e per ultima quella "Piè di pagina".

Risulterà adesso più chiaro quali siano le funzionalità degli specifici tipi di bande, e come essi possano essere adoperati in un unico report per ottenere l'impostazione desiderata.

La cosa ancora poco chiara è probabilmente come sia possibile "programmare" il tipo di banda "Dati". Infatti quest'ultimo tipo di banda ha dei comportamenti diversi dagli altri due. Analizziamo queste differenze in dettaglio per potere capire meglio come utilizzarla.

La prima differenza che risalta all'occhio è di certo il numero variabile di volte che essa viene stampata. La seconda il fatto che ad ogni stampa successiva, variano i valori dei campi ad essa collegati.

La risposta a questi quesiti è una sola. La banda dati, e strettamente collegata alla sorgente di dati in essa selezionata (vedremo più avanti come effettuare questa selezione), e da essa dipende. Quindi verrà stampata tante volte quanti sono gli elementi della sua sorgente. Ed ad ogni successiva stampa, avanzerà di un elemento nella sorgente, inviando alla stampa valori diversi in base all'elemento correntemente selezionato.

Le sorgenti di dati selezionabili sono di due tipi, sorgenti da database, e sorgenti di tipo virtuale. I collegamenti ai primi tipi di sorgente sono immediati, infatti queste sono facilmente selezionabili dall'elenco di quelle presenti nel programma corrente (dataset, tabelle, query, ecc...), e servono per poter stampare database. Il collegamento al secondo tipo di sorgente invece deve essere fatto in modo semi-manuale e serve per poter stampare sorgenti di dati diverse dai database, quali ad esempio vettori, file di testo, ecc...

Alla selezione della sorgente di dati, appare una finestra come questa:

Per selezionare una sorgente da database, basta selezionarla dall'elenco di quelle disponibili nel programma. Per selezionarne una virtuale (che dovrà essere poi programmata in base all'utilizzo) bisognerà selezionare come voce "[Sorgente di dati virtuale]". Per non effettuare nessuna scelta, bisognerà invece selezionare la voce "[Nessuno]".

Una volta selezionata la sorgente di dati voluta, bisognerà provvedere a inserire all'interno di ogni singola banda, gli oggetti che rappresentano i dati da visualizzare.

Tutte queste operazioni sono possibili tramite l'ausilio del modellatore.

Primi passi

Dopo una prima fase teorica, incominciamo adesso a prendere confidenza con questo nuovo strumento, tramite la costruzione di piccoli report di esempio.

Per prima cosa apriamo il nostro Delphi e creiamo una nuova applicazione. Apparirà la classica "Form1". Su di essa posizioniamo un TfrReport, una TTable, un TfrDesigner e due TButton, come in questa figura:

Attribuiamo alle proprietà di questi componenti i seguenti valori:

- ◆ TTable:
 - DataBase = DBDEMOS (*alias dei database distribuiti dalla Inprise insieme a Delphi*)
 - TableName = animals.dbf (*database contenente un elenco di animali*)
 - Active = True

- ◆ TButton1
 - Caption = Modifica

- ◆ TButton2
 - Caption = Visualizza

Adesso facciamo un doppio clic sul componente TfrReport. Apparirà immediatamente a pieno schermo la finestra del modellatore. A questo punto iniziamo a costruire il report.

La prima operazione da fare è quella di selezionare i tipi di banda desiderati. In questo primo esempio utilizzeremo solamente tre tipi di bande. Per potere effettuare questa selezione bisogna prima analizzare un po' più approfonditamente il funzionamento del modellatore. Facciamolo seguendo le indicazioni della seguente figura:

Ad ognuno dei quattro tasti sopra descritti, è associata la possibilità di inserire nel report correntemente in uso, un diverso tipo di oggetto, semplicemente selezionando il tasto sulla barra dei comandi (ossia la barra che contiene tutti i tasti sopra descritti) e lavorando sul report.

Al momento attuale, noi analizzeremo per primo il tasto "Banda", che permette di inserire all'interno del report in costruzione una o più tipi di bande, per poi proseguire rispettivamente con i tasti " Rettangolo di testo" ed "Immagine".

Per poter attivare il comando di "Banda", per prima cosa bisogna premere il relativo tasto sulla barra dei comandi, che così risulterà premuto, evidenziando la funzione momentaneamente attiva.

Fatto questo bisogna selezionare, all'interno del foglio rigato, l'area che si desidera attribuire alla banda, posizionandosi con il mouse sul punto del foglio che rappresenterà il vertice superiore/sinistro della banda, e tenendo premuto il tasto sinistro del mouse (o destro se si sono invertiti i tasti del mouse) trascinandolo fino alla posizione del vertice inferiore/destro, per poi lasciare il tasto del mouse.

A questo punto, apparirà una finestra che permetterà di selezionare il tipo di banda

desiderato. Si tenga presente che i tipi di banda non più utilizzabili (perché già in uso) saranno disabilitati.

Questa finestra sarà uguale alla seguente figura:

Selezionare come tipi di banda “Titolo report”, “Dati principali” e “Piè di pagina report”, che permettono di stampare rispettivamente all’inizio del report, al centro (area dati) ed alla fine.

Posizionandoli rispettivamente in alto, nel centro ed in basso, nel foglio rigato, ed attribuendo ad ognuna di esse rispettivamente dimensioni media, grande e piccola. Si tenga presente che per modificare le dimensioni di una banda, basta selezionarla e trascinare i punti evidenziati (per altezza e larghezza) nella posizione voluta.

A questo punto il modellatore apparirà come questa immagine:

L’operazione successiva è quella di indicare alla “Banda dati” con che sorgente lavorare, ossia da dove reperire i dati. Per far ciò basta fare un doppio clic con il mouse sulla banda implicata. Apparirà una finestra come la seguente:

Selezionare come sorgente dati dall’elenco a scorrimento, la voce “Form1.Table1”, che indica la tabella presente nella nostra form, e poi premere il tasto “OK”.

Una volta selezionata la sorgente dati, passiamo a disegnare sulla banda gli oggetti da stampare. Per effettuare questa operazione utilizzeremo i “ Rettangoli di testo”.

I “ Rettangoli di testo” servono per inserire del testo all’interno del report.

Per selezionarli come comando corrente, basta cliccare con il mouse sulla barra dei comandi (la stessa dalla quale abbiamo selezionato il comando “Bande”). E per inserirli nel report, basta procedere come per le bande.

Nell’esempio corrente useremo questo comando per inserire un rettangolo nella banda “Titolo report” per stampare il titolo, un rettangolo più una immagine nella banda “Dati principali” per stampare i dati di ogni animale più una sua foto, ed un altro rettangolo nella banda “Piè di pagina report” per indicare il numero di animali stampati.

Per inserire ogni singolo oggetto (rettangolo o immagine), procedere come per le bande.

Terminato l’inserimento di tutti gli oggetti, adesso vediamo di assegnare i valori alle variabili e categorie che ci serviranno in seguito. La prima cosa da fare è quella di aprire il menù alla voce “File” e poi eseguire il comando “Lista variabili...”. Apparirà una finestra come questa:

Da questa finestra è possibile gestire interamente le variabili e le categorie.

Per prima cosa aggiungiamo una categoria, poi modifichiamone il nome in “Variabili”. A questo punto aggiungiamo due nuove variabili, a cui daremo rispettivamente per nomi “Dimensione” e “Foto”.

Dalla finestra “Valore” selezioniamo come sorgente di dati “Table1”. Appariranno nell’elenco sottostante tutti i campi di questa tabella.

Selezionando la variabile "Foto", cliccare sul campo "BMP". Questa operazione, assegnerà alla variabile "Foto" il campo "BMP" della tabella "Table1".

Procediamo adesso con la variabile "Dimensione". A rigore di logica dovremmo attribuire a questa variabile il campo "SIZE" della tabella "Table1".

Il problema è che questo valore non è espresso in centimetri, dovremo quindi effettuare un conversione di unità di misura. Esattamente dovremmo attribuire alla variabile "Dimensione" il valore del campo SIZE moltiplicato per la costante di conversione 2.54.

Per fare questo selezioniamo la variabile "Dimensione" e poi dalla finestra "Valore" selezioniamo "Altro". Apparirà un elenco di valori di sistema, fra cui appare la voce "Espressione". Selezionando questa voce si attiverà automaticamente la finestra di testo in basso, con il nome "Espressione".

A questo punto è possibile seguire due diverse strade. La prima è quella di scrivere direttamente in questa finestra, il valore da assegnare alla variabile, ossia "SIZE * 2.54".

La seconda, quella di premere il tasto alla sua destra raffigurante tre punti "...", e di compiere l'operazione di assegnazione tramite l'editor di espressioni, raffigurato in questa immagine:

Qualunque strada si decida di percorrere, si tenga presente che entrambe portano al medesimo risultato.

Terminata la fase di assegnazione delle variabili, chiudiamo la finestra premendo il tasto "OK". Così facendo torneremo al normale ambiente del modellatore.

Lavoriamo adesso sugli oggetti inseriti in precedenza, assegnando ad ognuno le variabili necessarie e modificandone l'aspetto in base alle nostre esigenze.

Per modificare le caratteristiche di ogni oggetto bisogna utilizzare la finestra "Controllo oggetto". Questa finestra visualizza tutte le proprietà dell'oggetto correntemente selezionato, permettendo di modificarle in base alle proprie necessità.

Facendo un doppio clic sulla proprietà "(Testo)" di questa finestra, è possibile modificare il contenuto dell'oggetto correntemente selezionato. All'esecuzione di questo comando apparirà una finestra come la seguente:

Basta digitare il testo che si desidera assegnare all'oggetto in uso e premere "OK". Utilizzando questa tecnica, in ordine inserire all'interno di ogni oggetto i seguenti valori:

Oggetto rettangolo nella banda "Titolo report":

LISTA ANIMALI (*premere il tasto Invio*)
stampa effettuata il [DATE] alle [TIME]

Oggetto rettangolo nella banda "Dati principali":

Nome: [NAME] (*premere il tasto Invio*) (*premere il tasto Invio*)
Dimensione: [Dimensione] cm

Oggetto immagine nella banda "Dati principali":

[Foto]

Oggetto rettangolo nella banda “Piè di pagina report”:

Numero totale di pesci: [Count()]

Terminata questa fase e dopo avere modificato i parametri grafici (colore, dimensione caratteri, attributi, allineamento, ecc...) il report apparirà come la seguente immagine:

Si noti che i valori immessi fra parentesi quadre indicano a FuzzyReport di non visualizzare i valori stessi, ma di calcolarli come campi, variabili o espressioni.

Analizziamo sotto questa nuova luce i valori immessi precedentemente.

Oggetto rettangolo nella banda “Titolo report”:

[DATE] [TIME] *Variabili di sistema che restituiscono rispettivamente la data e l'orario corrente.*

Oggetto rettangolo nella banda “Dati principali”:

[NAME] *Nome di un campo della tabella “Table1”.*

[Dimensione] *Nome di variabile*

Oggetto immagine nella banda “Dati principali ”:

[Foto] *Nome di variabile*

All'oggetto immagine, bisogna sempre assegnare il nome di una variabile. Non è possibile assegnargli direttamente il nome di un campo.

Oggetto rettangolo nella banda “Piè di pagina report”:

[Count()] *Funzione che restituisce il numero di elementi della sorgente della banda dati (o di un'altra se specificata fra parentesi).*

Sarebbe possibile inserire anche intere espressioni. Ad esempio invece della variabile “Dimensione” di potrebbe scrivere direttamente “SIZE * 2.54”.

A questo punto, non rimane altro che salvare il report creato dandogli per nome “Esempio 1”, ed uscire. Se prima di uscire si desidera verificare il lavoro, basta premere il tasto raffigurante una lente d'ingrandimento, per avviare l'anteprima di stampa.

Tornati al Delphi, per terminare il nostro lavoro, basterà inserire il seguente codice sorgente per gli eventi “OnClick” dei due bottoni:

```
procedure TForm1.Button1Click(Sender: TObject);  
Begin  
    // Carica il report da disco  
    frReport1.LoadFromFile('Esempio 1.frf');  
    // Attiva il Modellatore del report  
    frReport1.DesignReport;  
End;  
  
procedure TForm1.Button2Click(Sender: TObject);  
Begin  
    // Carica il report da disco  
    frReport1.LoadFromFile('Esempio 1.frf');  
    // Visualizza l'anteprima del report  
    frReport1.ShowReport;  
End;
```

Adesso si salvi il lavoro svolto, e poi si esegua. Premendo il tasto “Modifica” si entrerà nel modellatore, potendo modificare il report a run-time. Premendo il tasto “Visualizza”, otterremo un’anteprima di stampa uguale a questa immagine:

I livelli e le relazioni master-detail

Dopo questo primo esempio completo, vediamo adesso di capire meglio cosa intendiamo per livelli (già accennati nella sezione “Le bande”).

Un livello non è altro che un singolo database (o sorgente di dati). Per report a più livelli, si intende un report che visualizza dati provenienti da database a cascata, ossia in relazione fra di loro.

In una relazione master-detail il primo livello è il master ed il secondo il detail. In una master-detail-subdetail, il subdetail è il terzo.

FuzzyReport, permette di stampare database fino a tre livelli di relazioni. Per poter fare questo utilizza dei tipi di banda specifici.

La bande “Dati principali”, “Intestazione principale” e “Piè di pagina principale”, servono a gestire la visualizzazione dei “Dati”, della “intestazione” e del “Piè di pagina” del primo livello”.

Il secondo livello è gestito dalle bande “Dati dettagli”, “Intestazione dettagli” e “Piè di pagina dettagli”.

Mentre per il terzo ci sono le bande “Dati sottodettagli”, “Intestazione dettagli” e “Piè di pagina sottodettagli”.

Poniamo ad esempio il caso di avere due database in relazione master-detail, contenenti il primo un elenco di clienti ed il secondo uno di ordini.

Selezionando un record nel primo database, otterremo nel secondo un elenco di record uguale agli ordini effettuati dal cliente in questione.

Il nostro scopo è quello di creare un report capace di visualizzare l’elenco clienti, visualizzando in dettaglio per ogni cliente, l’elenco dei suoi ordini.

Per giungere al nostro scopo basterà usare le bande “Dati principali” e “Dati dettagli” selezionando per ognuna come sorgente di dati il relativo database.

A questo punto bisognerà inserire un rettangolo di testo per ogni campo che si desidera visualizzare nella rispettiva banda (campi master nella banda “Dati principali” e campi detail in quella “Dati dettagli”).

Si tenga presente che la relazione fra i due database (normalmente TDataSet) non dovrà essere impostata dal report, ma come si fa sempre collegando i due database tramite un TDataSource ed impostando le proprietà “MasterSource;MasterFields”.

Nel caso in cui si disponesse di tre database in relazione master-detail-subdetail (ad esempio clienti, ordini ed articoli), le operazioni da seguire sarebbero del tutto analoghe. Bisognerebbe semplicemente aggiungere una terza banda dati “Dati sottodettagli”, dopo avere ovviamente collegato il terzo database al secondo, con il solito TDataSource.

I gruppi

Dopo avere sommariamente descritto i livelli, passiamo a spiegare l'ultima tipologia di insiemi presenti in FuzzyReport. Questi sono i gruppi.

La loro principale funzione è proprio quella di raggruppare fra di loro i record, appartenenti ad un database, aventi dei dati in comune, così da potere dividere la moltitudine di record in gruppi coerenti.

Come per i livelli anche i gruppi possono adoperarsi a cascata, ossia è possibile adoperarne più di uno per ogni pagina del report.

Per capire meglio quale potrebbe essere l'utilità pratica di un gruppo cerchiamo di fare un esempio.

Poniamo il caso di avere un database contenente un elenco di ordini composto dai campi Numero_ordine, Data_ordine, Totale_ordine e Ragione_sociale_cliente e che questo database sia ordinato per Ragione_sociale_cliente.

Poniamo anche il caso di volere stampare un elenco di tutti i clienti con i loro rispettivi ordini.

Dato che in questo caso disponiamo di un solo database, contenente sia le informazioni dei clienti che quelle degli ordini, non potremmo usare le relazioni master (clienti) – detail (ordini).

Per ottenere lo stesso risultato utilizzeremo proprio i gruppi, specificando di volere raggruppare il database in base al campo Ragione_sociale_cliente.

Per fare ciò basterà inserire la banda "Intestazione gruppo" prima di quella "Dati principali".

Alla banda "Dati principali" collegheremo il nostro database, inserendo in essa tre oggetti rettangolo che visualizzeranno rispettivamente i campi Numero_ordine, Data_ordine, Totale_ordine.

Nella banda "Intestazione gruppo" inseriremo invece un oggetto rettangolo che visualizzerà il campo Ragione_sociale_cliente.

Per terminare l'operazione basterà fare un doppio clic sulla banda "Intestazione gruppo". Apparirà una finestra che permetterà di inserire i criteri di raggruppamento (ossia la regola che devono soddisfare i record del database della banda "Dati principali" per essere considerati facenti parte di un gruppo invece che di un altro). In questa finestra inseriremo come regola: Ragione_sociale_cliente. In questo modo i record verranno raggruppati in base al cliente che ne ha fatto l'ordine.

A questo punto non manca che avviare l'anteprima per goderci il risultato.

Le tabelle

In questo capitolo analizzeremo la possibilità di stampare tabelle.

Per far questo bisognerà disporre di due sorgenti di dati. La prima per l'intestazione di ogni riga e la seconda per gli elementi delle righe.

In effetti dobbiamo vedere una tabella come un report a due livelli, dove l'intestazione di ogni riga (le celle della prima colonna per intenderci) sono prelevate dal primo livello (master) del report, mentre gli elementi effettivi delle righe sono presi dal secondo livello (Detail).

Le bande da utilizzare per poter creare una tabella sono fondamentalmente sei.

Ecco di seguito un elenco dei tipi di banda necessari con una dettagliata descrizione dell'impiego...

Banda	Tipo	Descrizione
Intestazione principale	Opzionale	Contiene il titolo delle colonne
Dati principali	Obbligatorio	Deve contenere la sorgente di dati del primo livello (Master), così da identificare le righe della tabella (un master per ogni riga).
Piè di pagina principale	Opzionale	Contiene i totali delle colonne
Intestazione Tabella	Opzionale	Contiene i titoli delle righe
Dati Tabella	Obbligatorio	Deve contenere la sorgente di dati del secondo livello (Detail), così da identificare le colonne della tabella.
Piè di pagina Tabella	Opzionale	Contiene i totali delle righe

I rettangoli di testo contenenti i dati relativi alle singole celle della tabella, dovranno essere inseriti all'intersezione della banda "Dati principali" con la banda "Dati tabella".

I sottoreport

In questo capitolo analizzeremo i sottoreport, ossia oggetti capaci di visualizzare il report in essi contenuto nel report e nella posizione, dove sono stati inseriti.

I sottoreport appaiono come comuni oggetti, e quindi sono selezionabili dalla stessa barra degli strumenti dalla quale sono selezionabili, bande, rettangoli di testo e immagini.

La loro principale peculiarità consiste nel non indicare degli oggetti a se stanti, ma degli altri report.

Esattamente quando si inserisce un sottoreport, la prima cosa che si nota è che il modellatore di FuzzyReport, crea una nuova pagina.

Sarà in questa pagina che dovremo andare a creare il report che verrà poi visualizzato nella posizione in cui è stato inserito il sottoreport.

Infatti quando creiamo un nuovo sottoreport, al suo interno viene scritto il numero di pagina nella quale andare ad inserire il report da visualizzare.

A questo punto non rimane che fare delle prove.

Report composti

In questo capitolo analizzeremo i report composti, ossia la possibilità di stampare contemporaneamente tutto un elenco di report come fossero un unico.

Per fare questo esiste un oggetto apposito, il suo nome è "TfrCompositeReport".

Utilizzarlo è semplice. Basta sapere che possiede tutte le stesse proprietà di un semplice report con un'aggiunta. Possiede in più la proprietà "Report" di tipo "TList".

Tramite questa proprietà è possibile manipolare i report da inserire.

Una volta terminata questa fase, basta operare come sempre.

Segue un breve stralcio di codice che unisce in un report composto due report dai nome "Report1" e "Report2" e ne effettua l'anteprima.

```
// CompositeReport è il nome del report composti.  
CompositeReport.Reports.Clear; // Elimina ogni report presente  
in elenco.  
CompositeReport.Reports.Add (Report1); // Aggiunge all'elenco  
il report 'Report1'.  
CompositeReport.Reports.Add (Report2); // Aggiunge all'elenco  
il report 'Report2'.  
CompositeReport.ShowReport; // Visualizza l'anteprima dei due  
report aggiunti.
```

Una breve nota sul fatto che non è possibile modificare tramite il modellatore un report composto. Bisognerà provvedere alla modifica dei suoi singoli elementi esternamente al report composto.

Adattamento testo e dintorni

In questo capitolo analizzeremo alcune delle principali proprietà dei rettangoli di testo, unitamente a proprietà simili (e ad esse collegate) delle bande.

Lo scopo di questa analisi è quello di rendere possibile la disposizione del testo contenuto nel report (in fase di stampa o anteprima), il più naturale possibile, non dando l'impressione che sia stato precostituito in schemi fissi (i rettangoli di testo), ma dandogli la possibilità di adattarsi autonomamente in base allo spazio localmente disponibile, impaginandosi in automatico.

Per riuscire in questo scopo, analizzeremo cinque semplici proprietà dei principali oggetti. Tre appartenenti ai rettangoli di testo e due appartenenti alle bande dati.

Analizziamo i problemi di disposizione del testo no per volta.

Il primo problema che risolveremo consisterà nel creare dei rettangoli di testo capaci di adattare la propria larghezza a quella effettiva del testo in essi contenuto.

Per fare ciò l'unica operazione da fare consiste nel selezionare dal menu contestuale del rettangolo di testo (selezionabile cliccando su di esso con il tasto destro del mouse) la voce "Larghezza automatica".

Poniamo adesso il caso di volere fare lo stesso con l'altezza del rettangolo di testo.

La procedura risulterà leggermente più complessa. Perché ? Semplice. Per adattare la larghezza basta semplicemente aumentarla o diminuirla in base all'esigenza, per l'altezza bisogna considerare che nel caso in cui essa superi quella della banda che contiene il rettangolo, allora la stessa banda dovrà modificarsi per rendere le modifiche armoniche con l'insieme.

Per risolvere questo secondo problema bisognerà selezionare dal menu contestuale del rettangolo di testo la voce "Adattato", mentre dal menu contestuale della band dati che lo contiene bisognerà selezionare la voce "Adattato".

Si tenga ben presente che solo l'attivazione di entrambe le voci permetterà di raggiungere l'obiettivo prefissatoci.

Il nostro terzo problema consiste nel voler portare a capo il testo di un rettangolo di testo, quando questo supera il bordo estremo (invece di farlo sparire oltre il bordo).

Per perdere questo possibile, basterà attivare, sempre dal menu contestuale del rettangolo, la voce "a capo". Si noti comunque che se il testo a capo aumenta notevolmente l'altezza del testo reale, sarebbe opportuno attivare le voci per il regolamento automatico dell'altezza (come visto nel precedente esempio).

Il quarto problema consiste nel fare in modo che rettangoli di testo molto alti, non vengano stampati alla pagina successiva quando non vi è più spazio alla fine di quella corrente, ma che alla fine di quella corrente vi si stampi solo una parte del rettangolo di testo, stampando la restante nella pagina successiva.

Detto in parole povere, voliamo fare i modo che i rettangoli di testo molto alti non

vengano stampati per intero, ma vengano spezzettati fra la fine della pagina corrente e l'inizio di quella successiva, onde rendere più uniforme la distribuzione globale del testo.

Per ottenere questo (apparentemente complesso) risultato bisognerà attivare due voci. Una appartenente al rettangolo di testo e l'altra alla banda dati che lo contiene.

La prima voce da attivare è "Adattato", mentre la seconda è "Banda interrotta".

Si tenga ben presente che solo l'attivazione di entrambe le voci permetterà di raggiungere l'obiettivo prefissatoci.

Questa proprietà è utilizzabile contemporaneamente alle altre sopra esposte.

Approfondimenti sulle sorgenti di dati

Lo scopo che ci prefiggiamo di raggiungere in questo capitolo e quello di analizzare in modo più approfondito, tutte le opzioni disponibili tramite la maschera di selezione delle sorgenti di dati.

Ma procediamo per passi. Per prima cosa analizziamo attentamente la maschera che appare quando ci viene chiesto di selezionare una sorgente di dati per la banda corrente.

La finestra è uguale alla seguente immagine.



Le voci in essa disponibili sono sei. Divise in due gruppi principali.

Dal primo è possibile selezionare sia la sorgente di dati (da database o virtuale) che l'intervallo di selezione della stessa.

Tramite il secondo è invece possibile applicare alla sorgente selezionata (solo nel caso di sorgenti da database) un filtro complesso.

Per filtro complesso si intende (a differenza dei semplici filtri per database) un filtro che permetta l'utilizzo di tutte le funzioni e gli operatori disponibili in FuzzyReport.

Analizziamo attentamente il primo gruppo. La sua prima voce è "Sorgente di dati" che permette di effettuare tre diversi tipi di selezione.

Si può selezionare una sorgente di dati database (ossia una tabella o DataSet di progetto).

Si può selezionare la voce "[Sorgente dati virtuale]", che permette di utilizzare come sorgente di dati informazioni provenienti da vettori, file di testo, ed altro.

L'ultima selezione disponibile è "[Nessuno]". Quest'ultima selezione disattiva l'utilizzo di ogni sorgente di dati.

Le voci successive sono "Inizio", "Fine" e "Numero".

La prima permette di scegliere l'inizio della selezione degli elementi della sorgente di dati. I suoi possibili valori sono "Primo elemento" ed "Elemento corrente".

La seconda permette di scegliere la fine della selezione degli elementi della sorgente di dati. I suoi possibili valori sono "Ultimo elemento", "Elemento corrente" ed "Elementi contati". Quest'ultima (qualora selezionata) attiva la terza voce, permettendo di selezionare un esatto numero di elementi per la sorgente di dati.

Il secondo gruppo di voci (Filtro) si descrive da sé. Permette di selezionare l'espressione che deve verificarsi affinché ogni elemento della sorgente di dati venga considerato o meno. E' possibile disattivare il filtro senza la necessità di eliminare l'espressione che lo compone, tramite la voce "Attiva filtro".

Programmiamo una sorgente di dati

In questo capitolo cercheremo di capire come sia possibile tramite oggetti e metodi specifici, programmare via codice le sorgenti di dati.

Il primo passo da seguire per poter programmare in modo semplice una sorgente di dati via codice, consiste nell'aggiungere nel progetto il controllo TfrDBDataSet  (o TfrUserDataSet  nel caso in cui si vogliono programmare sorgenti di dati virtuali).

Le proprietà di questo controllo sono identiche alle voci che appaiono nella finestra di selezione della sorgente di dati.

Ecco di seguito un dettagliato elenco delle proprietà di questi controlli...

Proprietà	Tipo	Descrizione
DataSet (solo TfrDBDataSet)	TDataSet	Indica il database da collegare alla sorgente di dati.
Filter (solo TfrDBDataSet)	String	Contiene l'espressione che verrà utilizzata per filtrare gli elementi della sorgente di dati. Verranno utilizzati solamente quelli che la verificheranno.
Filtered (solo TfrDBDataSet)	Boolean	Indica se attivare o meno il filtro.
RangeBegin	TRangeBegin	Permette di selezionare il primo elemento delle serie (rbFirst=primo elemento, rbCurrent=elemento corrente).
RangeEnd	TRangeEnd	Permette di selezionare l'ultimo elemento delle serie (reLast=ultimo elemento, reCurrent=elemento corrente, reCount=elementi contati)
RangeEndCount	Integer	Nel caso in cui la proprietà corrente sia settata a "reCount", questa proprietà indica il numero di elementi della serie da utilizzare.

Una volta impostate queste proprietà basterà associare la nuova sorgente di dati appena creata alla banda desiderata.

Per farlo basta utilizzare un metodo del report, nato proprio per questo scopo.

Il metodo in questione è "SetDataSetBand".

Ponendo il caso che il controllo usato sia di tipo TfrDBDataSet e che abbia per nome "frDBDataSet1", vediamo di seguito uno stralcio di codice sorgente che spiega in modo completo come programmare via codice una sorgente di dati (in questo caso da database)

...

```

// Report è il report che contiene l'oggetto.
With frDBDataSet1 Do
  Begin
    DataSet := Table1; // Tabella da collegare alla sorgente di
    dati.
    Filter := ''; // Non inserisce nessuna espressione nel
    filtro.
    Filtered := False; // Disattiva il filtro.
    RangeBegin := rbFirst; // Seleziona come primo elemento della
    serie il primo elemento della tabella.
    RangeEnd := reLast; // Seleziona come ultimo elemento della
    serie l'ultimo elemento della tabella.
  End;
  Report.SetDataSetBand ('Banda',frDBDataSet1); // Assegna alla
  banda di nome 'Banda' la nuova sorgente di dati.

```

Si tengano presenti due cose.

La prima è che non ci sarebbe stato bisogno di assegnare tutte le proprietà, ma solo quella dal nome “DataSet”, dato che a tutte le altre abbiamo assegnato i valori di default.

La seconda cosa da tenere presente è che una volta assegnata la nuova sorgente di dati alla banda sarà sempre possibile continuare a modificarne le proprietà.

E' possibile programmare ulteriormente il comportamento di queste sorgenti di dati, utilizzando i loro eventi o l'evento equivalente “OnDataSetWord” del report.

La differenza fondamentali fra gli eventi delle sorgenti di dati programmabili e l'evento generale del report è che quello generale viene chiamato sempre per ogni evento della sorgente di dati e per ogni sorgente di dati, mentre gli eventi delle sorgenti di dati programmabili sono singoli e specifici.

Di base non esiste una reale differenza, dipende tutto da come si desidera impostare il sistema di programmazione. Tenendo presente comunque che l'uno non esclude l'altro.

Analizzeremo adesso dettagliatamente gli eventi delle sorgenti di dati programmabili, ricordando che quelli dei report sono argomento di un capitolo specifico successivo.

Ecco di seguito un dettagliato elenco degli eventi...

Evento	Descrizione
Procedure OnCheckEOF (Sender: TObject; var EOF: Boolean)	Permette di programmare il momento in cui si arriva alla fine della sorgente di dati, impostando a True la variabile “EOF”.
Procedure OnFirst (Sender: TObject)	Permette di programmare il comportamento della sorgente di dati alla richiesta di posizionarsi al suo primo elemento.
Procedure OnNext (Sender: TObject)	Permette di programmare il comportamento della sorgente di dati alla richiesta di posizionarsi all'elemento successivo a quello corrente.

Report non da database

In questo capitolo cercheremo di capire come ci si debba comportare quando si desidera stampare dei dati non appartenenti a dei database.

Può essere il caso della stampa di un vettore, di un file di testo, di semplici variabili, o di ogni altra diversa sorgente di dati.

I passi fondamentali per riuscire in questo intento sono tre.

Il primo consiste nell'associare alla sorgente dati della banda non un database ma la dicitura "[Sorgente di dati virtuale]". Questo predispone FuzzyReport a prelevare i dati da visualizzare da sorgenti alternative.

Il secondo passo consiste nel definire le variabili tramite le quali si effettueranno questi passaggi di dati. Una per ogni valore da passare, con l'unico riguardo di assegnare ad ognuna di esse il valore predefinito "[Nessuno]". I valori reali verranno assegnati utilizzando l'evento "OnGetValue" che riceve in ingresso il nome della variabile interessata restituendone in uscita il valore.

Il terzo ed ultimo passo sarà quello di programmare via codice il comportamento di questa nuova sorgente di dati. Questo è possibile utilizzando l'evento "OnDataSetWork" che ricevendo in ingresso l'operazione corrente sulla sorgente di dati (apertura, chiusura, spostamento al primo elemento, spostamento all'elemento successivo, e controllo di fine elenco) ne permette di programmarne il comportamento.

Di seguito è riportato un estratto del codice di questi due eventi, più parte della dichiarazione del programma che ne fa uso.

Ecco il codice della dichiarazione delle costanti e della sorgente di dati virtuale (ilvettore)...

```
Const Lunghezza=10; // Lunghezza sorgente di dati  
        // Vettore contenente i dati (sorgente di dati)  
        Vettore:Array [1..Lunghezza] Of String=  
        ('Uno', 'Due', 'Tre', 'Quattro', 'Cinque', 'Sei', 'Sette',  
        'Otto', 'Nove', 'Dieci');  
        // L'indice del vettore. Indica il dato correntemente letto  
Var Indice:Integer;
```

Ecco il codice dell'evento "OnDataSetWork" che permette di specificare i comportamenti di questa sorgente di dati...

```

Procedure TForm1.frReport1DataSetWork (BandName:String; DataSet:
TDataSet;var RecordCount:Integer; Operation:TfrDataSetOperation);
Begin
  // Si usa il vettore in base all'operazione (Operation) in corso
  Case Operation Of
    // Setta la lunghezza del vettore
    OpInit : RecordCount := Lunghezza;
    // Porta il vettore al primo elemento
    OpFirst : Indice:=1;
    // Sposta il vettore all'elemento successivo
    OpNext : Inc(Indice);
  End;
End;

```

Ecco, in fine, il codice dell'evento "OnGetValue" che permette di specificare i i valori da attribuire alle variabili...

```

Procedure TForm1.frReport1GetValue (const ParName: String;
var ParValue: Variant);
Begin
  // In base al nome della variabile richiesta (in questo caso una
sola) si restituisce il valore corrispondente del vettore
  If ParName='Nome' Then ParValue := Vettore[Indice]
End;

```

Funzioni ed operatori

In questo capitolo cercheremo di spiegare il funzionamento di tutte le funzioni ed operatori presenti in FuzzyReport.

Effettivamente ormai si è giunti ad averne disponibili un elevato numero.

Ecco di seguito l'elenco degli operatori...

Tipo operatori	Operatori
Logici	>, <, OR, AND, NOT, =, <>, >=, <=
Matematici	-, *, +, MOD, /, DIV

Ecco di seguito, diviso per categorie, l'elenco delle funzioni...

Categoria	Funzione	Descrizione
Conversione	FORMATDATETIME (<X>, <Y>)	Restituisce la data <Y> con il formato <X>
""	FORMATFLOAT (<X>, <Y>)	Restituisce il numero <Y> con il formato <X>
""	STR (<X>)	Restituisce una stringa convertita dal numero <X>
""	STRTODATE (<X>)	Restituisce la data contenuta nella stringa <X>
""	STRTOTIME (<X>)	Restituisce l'orario contenuto nella stringa <X>
""	VAL (<X>)	Restituisce la conversione numerica di <X>
Logiche	IF (<X>, <Y>, <Z>)	Restituisce <Y> se l'espressione <X> è verificata altrimenti restituisce <Z>
Matematiche	ABS (<X>)	Restituisce il valore assoluto di <X>
""	ARCTAN (<X>)	Restituisce l'arcotangente di <X>
""	COS (<X>)	Restituisce il coseno di <X>
""	EXP (<X>)	Restituisce l'esponenziale di <X>
""	FRAC (<X>)	Restituisce la parte decimale del numero <X>
""	INT (<X>)	Restituisce la parte intera del numero <X>
""	LN (<X>)	Restituisce il logaritmo naturale di <X>
""	PI	Restituisce il valore di pigreco
""	POWER (<X>, <Y>)	Restituisce il numero equivalente a <X> elevato ad <Y>
""	ROUND (<X>)	Restituisce il numero intero corrispondente all'arrotondamento di <X>
""	SIN (<X>)	Restituisce il seno di <X>
""	SQR (<X>)	Restituisce il quadrato di <X>
""	SQRT (<X>)	Restituisce la radice quadrata di <X>
""	TRUNC (<X>)	Restituisce il numero intero corrispondente al troncamento di <X>
""	CEIL (<X>)	Restituisce il numero intero corrispondente all'arrotondamento per eccesso di <X>
""	FLOOR (<X>)	Restituisce il numero intero corrispondente all'arrotondamento per difetto di <X>
Statistiche	AVG (<X>)	Restituisce la media della serie <X>
""	COUNT ()	Restituisce il numero di elementi della serie
""	MAX (<X>)	Restituisce il valore minimo della serie <X>
""	MIN (<X>)	Restituisce il valore massimo della serie <X>

""	SUM (<X>)	Restituisce la somma della serie <X>
Stringa	COPY (<X>,<Y>,<Z>)	Restituisce una sotto-stringa copiata dalla stringa <X> a partire dalla posizione <Y> per <Z> caratteri
""	LENGTH (<X>)	Restituisce la lunghezza della stringa <X>
""	LOWERCASE (<X>)	Restituisce <X> in minuscolo
""	NAMECASE (<X>)	Restituisce <X> in minuscolo eccetto che per il primo carattere in maiuscolo
""	NUMBERTOLETTER (<X>)	Restituisce l'equivalente in lettere del numero <X>
""	PHRASECASE (<X>)	Restituisce <X> in minuscolo eccetto che per le iniziali di ogni parola contenuta nel testo
""	POS (<X>,<Y>)	Restituisce il numero corrispondente alla posizione della sotto-stringa <X> nella stringa <Y>
""	TRIM(<X>)	Restituisce la stringa <X> ripulita dagli spazi presenti ai suoi estremi
""	TRIMLEFT(<X>)	Restituisce la stringa <X> ripulita dagli spazi presenti nel suo estremo sinistro
""	TRIMRIGHT(<X>)	Restituisce la stringa <X> ripulita dagli spazi presenti nel suo estremo destro
""	UPPERCASE (<X>)	Restituisce <X> in maiuscolo
""	FILLSTR (<X>)	Restituisce una stringa formata da <X> volte la sotto-stringa <Y>
Valuta	EURO (<X>)	Restituisce una stringa contenente il numero <X> formattato come euro
""	EUROTOMONEY (<X>)	Restituisce una stringa contenente il numero <X> convertito da euro alla moneta corrente e formattato
""	EUROTOMONEY (<X>)	Restituisce il numero <X> convertito da euro alla moneta corrente
""	MONEY (<X>)	Restituisce una stringa contenente il numero <X> formattato come valuta nel sistema corrente
""	MONEYTOEURO (<X>)	Restituisce una stringa contenente il numero <X> convertito dalla moneta corrente in euro e formattato
""	MONEYTOEURO (<X>)	Restituisce il numero <X> convertito dalla moneta corrente in euro

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">Formattazione ed evidenziazione

Adesso vedremo un esempio sull'utilizzo della formattazione delle variabili. L'utilizzo di questa tecnica permette di personalizzare la visualizzazione di ogni singola variabile nel modo che più si ritiene opportuno. Questo può essere molto comodo, ad esempio quando si desidera formattare una variabile per visualizzarla come moneta, unità di misura o altro.

Per visualizzare l'editor di formato cliccare con il tasto destro del mouse su di un rettangolo, apparirà un menu, cliccare sulla voce "Formatta variabile...".

Apparirà una finestra come questa:



La prima casella indica il tipo di variabile (Testo, Numero, Data, Orario, Logico). La seconda seleziona il tipo di formato di visualizzazione.

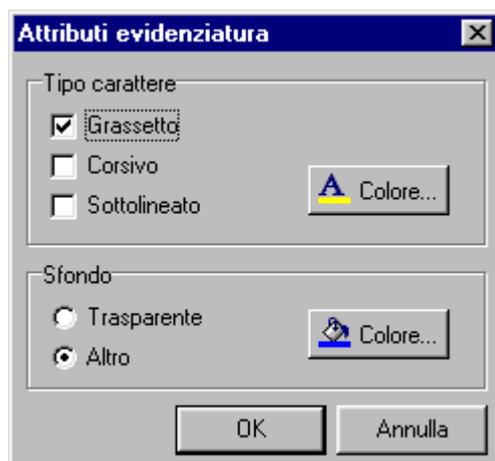
Il tipo "Testo" non consente alcun tipo di formattazione, mentre il tipo "Numero" consente di selezionare la formattazione, il numero di cifre decimali ed il simbolo di frazione.

A parte i tipi "Testo" e "Logico", è possibile selezionare un tipo di formattazione personalizzata. Utilizzando la stessa sintassi comune alle formattazioni in ambiente DELPHI.

E' anche possibile includere il concetto di evidenziazione, che è molto usato per dare risalto al report. Per fare ciò basta specificare la condizione da applicare alle variabili che dovranno essere evidenziate nello spazio apposito. Utilizzando come indicativo generico della variabile, il testo "Valore", (o "Value" se si desidera internazionalizzare il proprio report), come si vede in figura.

E' anche possibile personalizzare l'evidenziazione per ogni rettangolo. Per fare questo basta selezionare il rettangolo sulla quale si desidera effettuale la personalizzazione e premere il tasto sulla barra degli strumenti raffigurante .

Apparirà una finestra come questa:

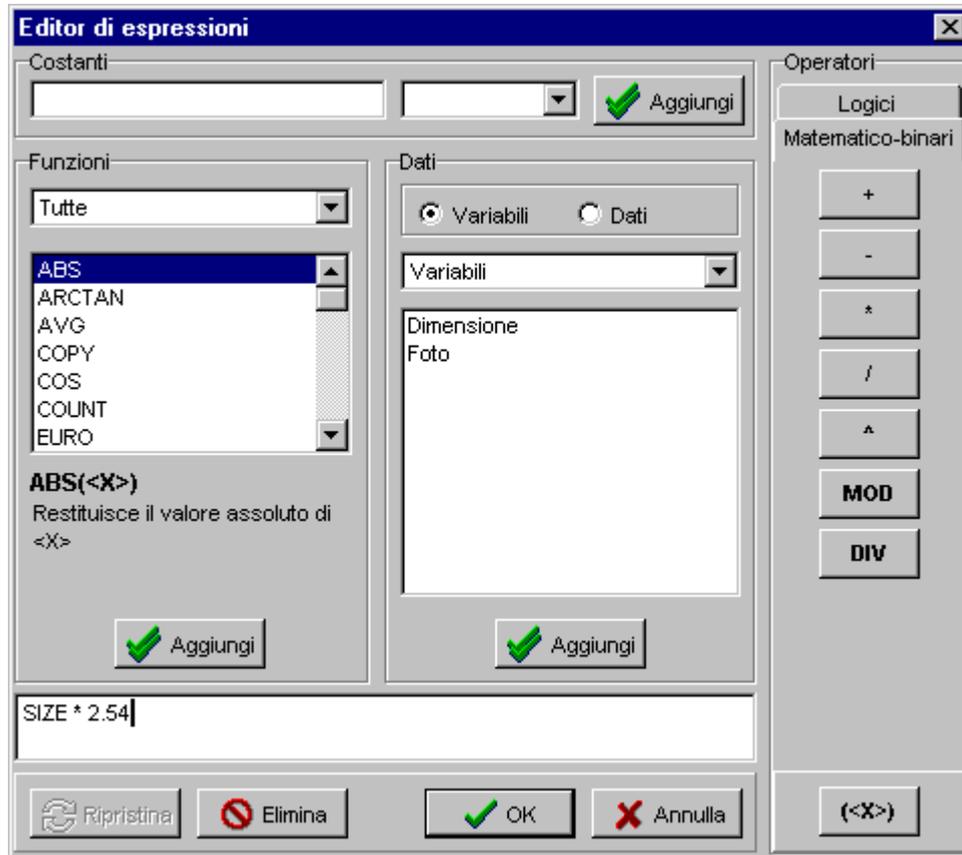


Tramite questa finestra è possibile selezionare lo stile ed i colori dell'evidenziatura, oltre a decidere se applicarla ad uno sfondo uniforme (in base al colore selezionato) o in modo trasparente.

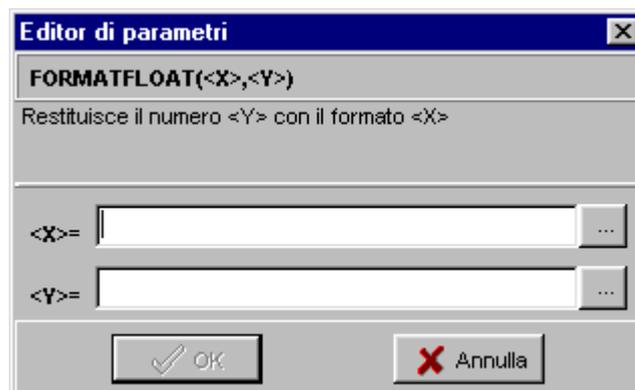
L'editor di espressioni

Quando si visualizza l'elenco delle variabili, questa è la finestra che appare:

Quando ad una variabile si attribuisce "Altro/Espressione", in basso vengono attivate una casella di testo, per l'inserimento manuale ed un bottone "[...]" per la composizione guidata. Cliccando su questo tasto appare una finestra come questa:



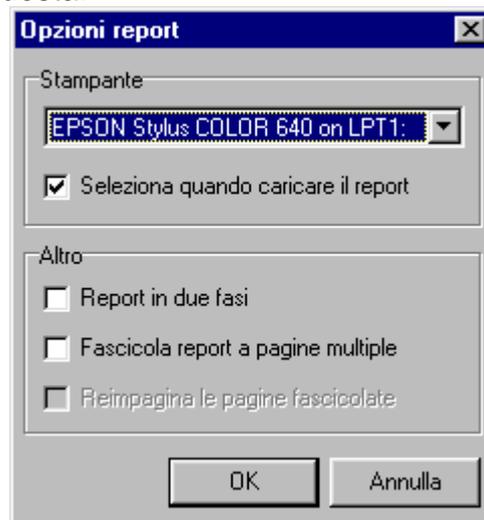
Tramite questa finestra è possibile utilizzare tutte le funzioni disponibili (incluse quelle aggiunte dall'utente), visualizzandole per categoria di appartenenza. Cliccando sul tasto "Aggiungi" appare la seguente finestra:



Questa finestra permette all'utente di inserire i parametri della funzione selezionata. Essi potranno essere inseriti manualmente, digitandoli nelle apposite caselle di testo, o tramite l'editor dell'espressioni, premendo i tasti "[...]".

Opzioni report ssss<big><big>Opzioni report e pagina

Selezionando dal modellatore di report il menù file e poi la voce Opzioni report... apparirà una finestra come questa:



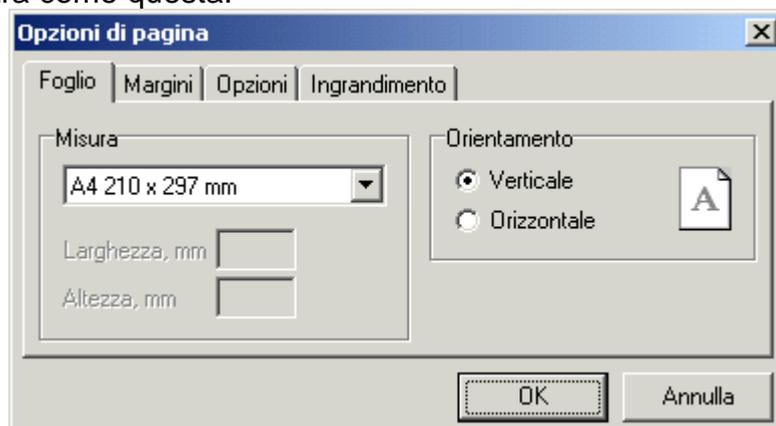
Dalla maschera "Stampante" è possibile selezionare la stampante da utilizzare per stampare il report corrente, mentre dalla maschera "Altro" è possibile selezionare delle opzioni molto interessanti.

La prima 'Report in due fasi' elabora il report in due fasi permettendo di calcolare anticipatamente il numero totale delle pagine generate. Quando questa opzione è attiva, è possibile utilizzare la variabile di sistema "[TotalPages]" per indicare il numero totale di pagine presenti nel report.

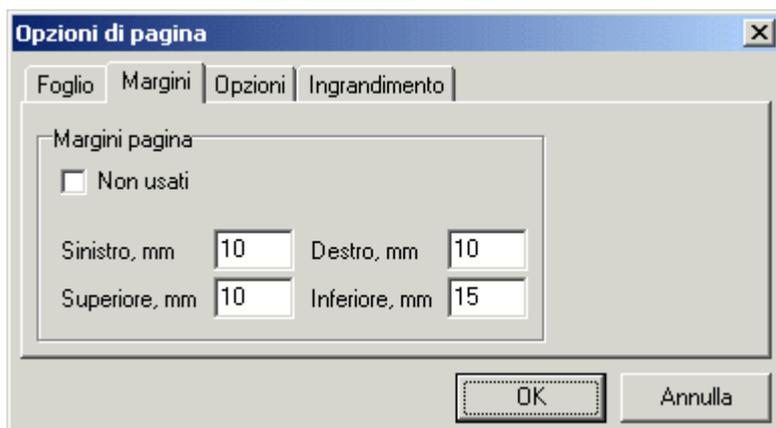
La seconda opzione permette di fascicolare un report multipagina (un report costituito da più pagine, ognuna delle quali definisce un sottoreport a se stante). Un report multipagina standard le pagine del report finale sono così suddivise (1°primo, 2°primo, 3° primo, 1°secondo, 2° secondo, 3° secondo, 1° terzo, 2° terzo, 3° terzo), mentre in un report multipagina fascicolato le pagine del report finale verranno così suddivise (1° primo, 1°secondo, 1°terzo, ...2°primo, 2°secondo, 2° terzo, ...).

Attivando questa opzione sarà possibile (tramite la terza opzione) decidere se reimpaginare i numeri di pagina (ossia rispettare la posizione effettiva della nuova fascicolatura), o se lasciare l'ordine originario, nella stampa dei numeri di pagina.

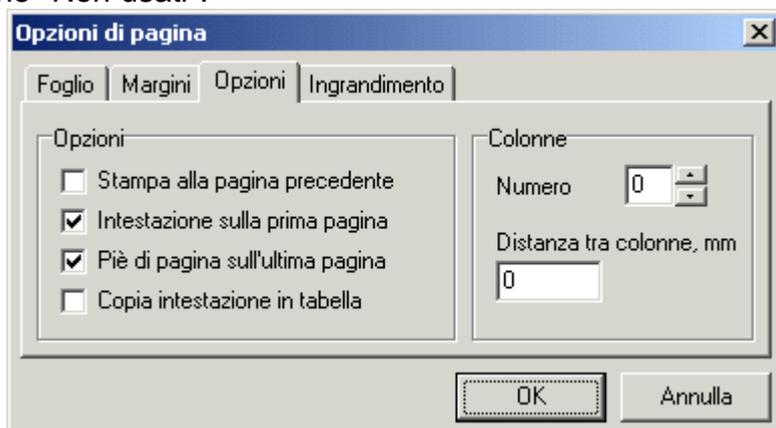
Selezionando dal modellatore di report il menù file e poi la voce Opzioni pagina... apparirà una finestra come questa:



Da questa maschera "Foglio" è possibile selezionare la misura del foglio, selezionando il tipo di foglio. O selezionando come tipo di foglio quello personalizzato, inserire le misure desiderate.



Da questa maschera "Margini" è possibile scegliere i margini che si desiderano utilizzare. Nel caso in cui non si desiderino utilizzare i margini è allora possibile selezionare l'opzione "Non usati".



Da questa maschera "Opzioni" è possibile selezionare uno svariato insieme di opzioni molto interessanti. Queste opzioni possono essere personalizzate per ogni pagina del report, ovviamente questo solo nei report a pagine multiple.

Dalla sottomaschera "Colonne" è possibile selezionare il numero di colonne che si desidera avere nel report e la distanza in mm che ognuna deve avere dalla successiva. I report a più colonne permettono di dividere il foglio in più parti, utilizzandone ognuna come se fosse un singolo foglio.

Dalla sottomaschera "Opzioni" è possibile selezionare quattro diverse opzioni. Andiamo a scoprirle in ordine.

La prima "Stampa alla pagina precedente" funziona solo con i report a pagine multiple. Se attiva permette di visualizzare la pagina corrente nella pagina precedente, creando così l'effetto della sovrapposizione delle due pagine. Questo può essere comodo quando si desiderano comporre elaborazioni di pagine differenti.

La seconda "Intestazione alla prima pagina", se attiva, stampa l'intestazione alla prima pagina del report.

La Terza "Piè di pagina all'ultima pagina", se attiva, stampa il piè di pagina, all'ultima

pagina del report.

La Quarta "Copia intestazione in tabella", se attiva, copia l'intestazione della tabella di un report visualizzato in più pagine, in tutte le pagine che compongono la tabella stessa.

Da questa maschera "ingrandimento" è possibile selezionare l'ingrandimento (o rimpicciolimento) di ogni singola pagina del report, lavorando sia sull'altezza che sulla larghezza.



I metodi del report

In questo capitolo analizzeremo i metodi utilizzabili in fase di programmazione in FuzzyReport.

Ecco l'elenco dei principali metodi...

Metodo	Descrizione
Function FindVariable(Variable:String): Integer	Restituisce l'indice corrispondente alla variabile cercata all'interno delle variabili del report.
Function FindObject(Name:String):TfrView	Restituisce il puntatore all'oggetto del report (rettangolo, immagine, banda, ecc...) cercato in base al nome. Se non trova l'oggetto restituisce un puntatore a Nil.
Function ObjectByName(Name:String): TfrView	Identico a FindObject ma nel caso in cui non trovi l'oggetto cercato solleva un'eccezione.
Procedure DeleteObject (T:TfrView);	Elimina dal report l'oggetto puntato dalla variabile "T".
Procedure InsertObject (T:TfrView;PageNo,Index:Integer);	Inserisce nel report l'oggetto puntato dalla variabile "T", nella pagina "PageNo" (a base 0), alla posizione "Index" (a base 0).
Procedure SetDataSetBand (BandName:String;DataSet:TfrDataSet)	Permette di collegare una banda del report ad un TfrDataSet (oggetto che permette di modificare a run-time le caratteristiche di una sorgente di dati).
Function ChekBandType (BT:TfrBandType;PageNo:Integer):Boolean	Restituisce True se il tipo di banda cercata è presente alla pagina specificata. Altrimenti restituisce False.
Procedure GetVarList(CatNo:Integer; List:TStrings)	Restituisce l'elenco delle variabili associate ad una determinata categoria (identificata dal suo indice numerico).
Procedure GetCategoryList(List:TStrings)	Restituisce l'elenco delle categorie.
Procedure LoadFromStream (Stream:TStream)	Carica il report da uno stream.
Procedure SaveToStream(Stream:TStream)	Salva il report in uno stream.
Procedure LoadFromFile(FName:String)	Carica il report da un file.
Procedure SaveToFile(Fname:String)	Salva il report in un file.
Procedure LoadFromDB(DataSet:TDataSet; DocN:Integer)	Carica il report da un database composto da due campi. Il primo (numerico) è la chiave di accesso al report. Il secondo (Blob) lo contiene.
Procedure SaveToDB(DataSet:TDataSet; DocN:Integer)	Salva il report in un database con la struttura uguale a quello di LoadFromDB.
Procedure LoadTemplate(fname:String; comm:TStrings; Bmp:Tbitmap; Load:Boolean)	Carica un modello di report restituendo True alla variabile "Load", il commento alla variabile "comm", e l'immagine raffigurativa nella variabile "Bmp".
Procedure SaveTemplate(fname:String; comm:TStrings; Bmp:TBitmap)	Salva un modello di report, attribuendogli "comm" come commento e "Bmp" come immagine descrittiva.

Procedure LoadPreparedReport (FName:String)	Carica un report già preparato. Ossia l'equivalente di un'anteprima. Non carica quindi la struttura del report.
Procedure SavePreparedReport (FName:String)	Salva un report già preparato. Ossia l'equivalente di un'anteprima. Non salva quindi la struttura.
Procedure DesignReport	Permette di entrare nel modellatore del report. Per effettuare modifiche alla sua struttura. Per potere adoperare questo comando a run-time, bisognerà inserire nel progetto il componente TfrDesigner.
Procedure NewReport	Cancella il contenuto del report, lasciando solo una pagina vuota.
Function PrepareReport: Boolean	Prepara il report, senza effettuarne ne stampa ne anteprima.
Procedure ExportTo(Filter:TClass; FileName:String)	Esporta il report nel formato del filtro selezionato (testo, ...).
Procedure ShowReport	Prepara un report e ne visualizza l'anteprima.
Procedure ShowPreparedReport	Visualizza l'anteprima di un report già preparato.
Procedure PrintPreparedReport (FromPage,ToPage,n:Integer; Collate:Boolean)	Stampa un report già preparato.
Procedure PrintReport;	Prepara e stampa un report.
Function ChangePrinter (OldIndex,NewIndex:Integer):Boolean	Seleziona una stampante diversa per stampare il report.

Gli eventi del report

In questo capitolo analizzeremo gli eventi utilizzabili in fase di programmazione in FuzzyReport.

Ecco l'elenco dei principali eventi...

Evento	Descrizione
Procedure OnAfterPrint	Viene sollevato al termine della stampa del report.
Procedure OnBeforePrint (var Continue: Boolean)	Viene sollevato all'inizio della stampa del report, ed in base al valore restituito dalla variabile "Continue" decide se continuare od annullare la stampa. Per default il valore della variabile "Continue" è impostato a "True" (continua la stampa), ma se impostato a "False" annulla l'esecuzione della stampa in corso.
Procedure OnBeginDoc	Viene sollevato all'inizio della preparazione del report.
Procedure OnBeginPage (pg: Integer)	Viene sollevato all'inizio della preparazione di una pagina del report. Dove per pagina si intende non quella del report preparato (quello che si vede in anteprima), ma quella della struttura del report.
Procedure OnDataSetWork (BandName: String; DataSet: TDataSet; var RecordCount: Integer; Operation: TfrDataSetOperation)	Viene sollevato quando il report lavora con le sorgenti di dati, indicando sia la banda della sorgente (bandName), l'eventuale database ad essa collegato (TDataSet), il numero di elementi da considerare (RecordCount) e l'operazione richiesta (Operation). Questo evento è normalmente utilizzato per gestire le sorgenti di dati virtuali o per programmare comportamenti particolari delle sorgenti collegate a database.
Procedure OnDrawObject (View: TfrView)	Viene sollevato prima del disegno di ogni oggetto, anche di quelli invisibili (proprietà "Visible" settata a False).
Procedure OnEndDoc	Viene sollevato alla fine della preparazione del report.
Procedure OnEndPage (pg: Integer)	Dove per pagina si intende non quella del report preparato (quello che si vede in anteprima), ma quella della struttura del report.
Procedure OnGetValue (const ParName: String; var ParValue: Variant)	Viene sollevato ogni qualvolta si incontra una variabile con valore nullo (settata a "[Nessuno]"). Viene normalmente adoperato per attribuire valori particolari alle variabili del report.

Procedure OnProgress (n: Integer)	Viene sollevato dopo la preparazione di ogni pagina del report. Questo evento si solleva sia durante la fase di preparazione del report che durante quella di stampa.
Procedure OnUserFunction (const name: String; p1, p2, p3: Variant; var Val: Variant)	Viene sollevato quando all'interno del report si trovano chiamate a funzioni sconosciute. Questo evento rende possibile la definizione di nuove funzioni, alle quali è possibile passare fino a tre diversi parametri (p1, p2, p3). Il risultato della funzione deve essere restituito assegnandolo alla variabile "Val".

Le proprietà del report

In questo capitolo analizzeremo le proprietà pubbliche (tecnicamente publi, modificabili solo a run-time, via codice) e pubblicate (tecnicamente published, modificabili anche a design-time tramite l'object-inspector) utilizzabili in fase di programmazione in FuzzyReport.

Ecco l'elenco delle principali proprietà, divise per categoria...

Categoria	Nome	Tipo	Descrizione
Public	FileName	String	Contiene il nome del file dalla quale si è caricato il report. Questo nel caso in cui il report sia stato caricato da disco.
""	Build	Boolean	E' impostata a True quando il report è in preparazione ed a False in caso contrario.
""	CanRebuild	""	E' impostata a True quando in memoria è oresente la struttura del report (che puo' quindi essere preparato) e non solo il report preparato, ce può essere usato autonomamente.
""	HasBlobs	""	E' impostata a True quando il report usa campi (TField) di tipo Blob, tipo immagini, memo od altro.
""	Terminated	""	E' impostata a True quando è stata terminata dall'utente la fase di preparazione.
""	DoublePass	WordBool	E' impostata a True nel caso in qui il report debba essere eseguito in due fasi. Settare questa proprietà, permette di utilizzare all'interno del report la variabile di sistema [TotalPages] che restituisce il numero totale di pagine del report.
""	Fascicoli	""	Impostata a True, permette di fascicolare un report a più pagine.
""	PrintToDefault	""	Impostata a True, stampa il report nella stampante predefinita.
""	Reimpose	""	Impostato a, reimpagina un report fascicolato. Attenzione vâ usata solo ed esclusivamente quando le proprietà "Fascicoli" e "DoublePass" sono impostate a True.
""	Variables	TStrings	Contiene l'elenco delle variabili del report.

""	Pages	TfrPages	Contiene la struttura delle pagine del report da preparare (quella che si vedono nel modellatore).
""	EMFPages	TfrEMFPages	Contiene la struttura delle pagine del report già preparate (quelle che si vedono in anteprima).
Published	DisableDataSetControls	Boolean	Impostata a True, permette di disabilitare automaticamente tutti i controlli associati ai database collegati al report.
""	Scaled	""	Impostata a True forza il ridimensionamento dei rettangoli di testo in caso di risoluzioni diverse da quella originale (da usare unitamente alla proprietà "PixelsPerInch").
""	ShowProgress	""	Impostata a True, permette di visualizzare la fase di preparazione del report, tramite una finestra che visualizza il numero della pagina correntemente in preparazione.
""	ModalPreview	""	Impostata a True, permette di visualizzare l'anteprima di stampa come finestra modale, viceversa come finestra indipendente.
""	AutoLoad	String	Indica il nome del file che contiene il report da caricare automaticamente all'atto dell'apertura del programma.
""	ReportDir	""	Se impostata, indica la cartella dalla quale caricare i report e nella quale salvarli.
""	TemplateDir	""	Indica la cartella dalla quale prelevare i modelli di report da utilizzare nel modellatore.
""	Title	""	Indica il titolo del report corrente.
""	PixelsPerInch	Integer	Indica la risoluzione originale con cui è stato creato il report (da usare unitamente alla proprietà "Scaled").

“”	DesignerOptions	TfrDsgOptions	<p>Contiene un insieme di opzioni applicabili al modellatore.</p> <p>L'unica opzione al momento disponibile è “doSaveConfirm” che se impostata attiva la finestra di richiesta di salvataggio. In caso contrario permette l'uscita dal modellatore o l'utilizzo di un altro report, senza chiedere conferma di salvataggio per quello corrente (comunemente da usare nel caso in cui si vogliono salvare i report su destinazioni diverse dai files, come ad esempio i DataBase).</p>
“”	Values	TfrValues	<p>Permette di creare, modificare e cancellare, sia categorie che variabili. Il tutto tramite una piacevole ed intuitiva interfaccia grafica.</p>

I metodi della classe base

In questo capitolo analizzeremo i metodi della classe base da cui derivano tutti i componenti utilizzabili in FuzzyReport (rettangoli, immagini, bande, sottoreport, ecc.).

Il nome della classe è TfrView, da essa derivano sia tutti i componenti standard di FuzzyReport, che quelli avanzati.

Ecco di seguito l'elenco dei metodi...

Metodo	Descrizione
Procedure Assign (From:TfrView)	Permette di assegnare un oggetto ad un altro.
Procedure Draw (Canvas:Tcanvas)	Disegna l'oggetto corrente nel canvas selezionato.
Procedure LoadFromStream (Stream:Tstream)	Permette di caricare l'oggetto da uno stream.
Procedure SaveToStream (Stream:Tstream)	Permette di salvare l'oggetto in uno stream.
Procedure DefinePopupMenu (Popup:TpopupMenu)	Permette di definire il menù contestuale che appare cliccando con il tasto destro sull'oggetto nel modellatore. Serve normalmente ad attribuire nuove proprietà all'oggetto in fase di modellazione.
Procedure SetAllFrames;	Attiva tutti i bordi (alto, destro, basso e sinistro) dell'oggetto.
Procedure ResetAllFrame	Disattiva tutti i bordi (alto, destro, basso e sinistro) dell'oggetto.

Le proprietà della classe base

In questo capitolo analizzeremo le proprietà della classe.

Eccone di seguito un dettagliato elenco...

Proprietà	Tipo	Descrizione
Left	Integer	Indica la posizione (in pixel) del margine sinistro dell'oggetto.
Top	""	Indica la posizione (in pixel) del margine superiore dell'oggetto
Width	""	Indica la larghezza (in pixel) dell'oggetto,
Height	""	Indica l'altezza (in pixel) dell'oggetto.
FrameWidth	""	Indica lo spessore (in pixel) del bordo dell'oggetto.
FrameColor	TColor	Indica il colore de bordo dell'oggetto.
FrameStyle	TPenStyle	Indica lo stile del bordo dell'oggetto.
Color	TColor	Indica il colore dell'oggetto.
Format	Integer	Indica il formato di visualizzazione dell'oggetto (è un indice numerico che individua il valore fra una serie di formati precostituiti).
FormatStr	String	Stringa che contiene la formattazione personalizzata dell'oggetto.
Memo	TStringList	Contiene il testo dell'oggetto.
Visible	Boolean	Impostata a True visualizza l'oggetto viceversa per il False.
Name	ShortString	Contiene il nome assegnato all'oggetto.
ParentReport	TfrReport	Restituisce il report a cui l'oggetto è associato.
Caption	String	Indica il contenuto del report come stringa unica.
DrawFrameTop	Boolean	Impostata a True abilita il disegno del bordo alto dell'oggetto.
DrawFrameBottom	""	Impostata a True abilita il disegno del bordo basso dell'oggetto.
DrawFrameLeft	""	Impostata a True abilita il disegno del bordo sinistro dell'oggetto.
DrawFrameRight	""	Impostata a True abilita il disegno del bordo destro dell'oggetto.

Oggetti standard

In questo capitolo analizzeremo le proprietà degli oggetti standard di FuzzyReport, ossia quelli il cui codice sorgente è integrato con quello di FuzzyReport. Questi sono i rettangoli, le immagini e le bande. Gli altri oggetti sono da considerarsi oggetti avanzati e quindi da analizzare in diversa sede.

Non viene fatto un dettagliato elenco dei loro metodi, perché ridefiniscono solamente quelli della classe base, non aggiungendone mai di nuovi.

Ecco di seguito un dettagliato elenco delle loro proprietà, diviso per classe...

Classe	Proprietà	Tipo	Descrizione
TfrMemoView (Rettangolo)	Highlight	TfrHighlightAttr	Contiene gli attributi grafici da attribuire al testo da evidenziare.
""	HighlightStr	String	Contiene l'espressione che deve verificarsi per evidenziare un testo.
""	UseHighlight	WordBool	Indica se attivare o meno (True/False) l'evidenziatura.
""	Font	TFont	Indica il tipo di font dei caratteri che compongono il testo del rettangolo.
""	Alignment	TfrAlignment	Indica il tipo di allineamento orizzontale del testo (frLeftJustify = sinistro, frRightJustify = destro, frCenter = centrato, frWidthJustify = totalmente giustificato).
""	VerticalAlignment	""	Indica il tipo di allineamento verticale del testo (tvaUp = superiore, tvaCenter = centrato, tvaDown = inferiore).
""	Rotate90	Boolean	Impostata a True ruota di 90°, in senso antiorario, il rettangolo, incluso il testo in esso contenuto.
""	AutoStretch	""	Adatta l'altezza del rettangolo in base a quella effettiva del testo in esso contenuto.
""	AutoSize	""	Adatta la larghezza del rettangolo in base a quella del testo in esso contenuto.
""	WordWrap	""	Porta a capo il testo che oltrepassa il bordo destro del rettangolo.

TfrBandView (Banda)	AutoStretch	Boolean	Attivata insieme alla stessa proprietà degli oggetti (che la posseggono), permette di adattare in modo automatico l'altezza degli oggetti in base al loro contenuto.
""	ForceNewPage	""	Impostata a True, dopo ogni stampa, forza la reazione e lo spostamento ad una nuova pagina.
""	PageBreak	""	Impostata a True, permette di stampare la gli oggetti contenuti nella banda spezzettandone il contenuto fra le varie pagine. In altre parole se un oggetto che dovrebbe essere stampato a fine pagina non entra per interno, con questa opzione disattivata verrà stampato nella pagina successiva, con questa opzione attivata, verrà stampato un pezzo nella pagina corrente, ed il restante ella successiva. Funziona solo con gli oggetti che supportano questa caratteristica.
""	PrintifSubsetEmpty	""	Impostata a True, stampa la banda anche se il suo livello inferiore (Detail o SubDetail) non contiene elementi.
""	BandType	TfrBandType	Indica il tipo di banda.
TfrPictureView (Immagine)	Picture	TPicture	Contiene immagine associata all'oggetto.
""	Center	Boolean	Impostata a True centra l'immagine.
""	KeepRatio	""	Impostata a True, mantiene costanti le proporzioni dell'immagine.
""	Stretch	""	Impostata a True, permette di ridimensionare l'immagine a piacimento.

Si tenga presente che queste stesse proprietà sono accessibili dal modellatore, attraverso il menu contestuale che appare cliccando con il tasto destro su di ogni oggetto.

Oggetti avanzati

In questo capitolo analizzeremo gli oggetti avanzati presenti in questa versione di FuzzyReport.

Al momento esistono solo sei di questi oggetti, il RichText (testo formattato), lo Shape (forme geometriche), il Codice a barre, il Grafico, l'oggetto OLE (per inserire documenti Word, Excel, Corel Draw, o altri, all'interno di un report) ed il rettangolo di testo con bordi personalizzabili.

Per potere utilizzare questi oggetti all'interno del modellatore (e quindi del report) è necessario collocarne uno (di ogni tipo di oggetto avanzato usato) all'interno del progetto.

Infatti a differenza degli oggetti standard (quelli nati con FuzzyReport ed il cui codice è integrato nel suo stesso nucleo), il codice necessario per l'utilizzo degli oggetti avanzati, non viene mai compilato con il progetto, tranne nel caso in cui il componente interessato non venga inserito nel progetto.

Questa operazione infatti include l'intero codice sorgente della libreria che contiene l'oggetto inserito, all'interno del progetto. Rendendo disponibile a run-time il codice necessario per la gestione di questo nuovo componente.

Nel caso in cui si usino dei report contenenti oggetti avanzati, con dei progetti che non contengono gli stessi oggetti inseriti, FuzzyReport, genererà un errore in fase di preparazione del report, indicando quale oggetto avanzato deve essere inserito nel progetto.

Ecco di seguito un dettagliato elenco delle loro proprietà, diviso per classe...

Classe	Proprietà	Tipo	Descrizione
TfrRichView	RichEdit	TRichEdit	Memorizza il contenuto dell'oggetto (testo formattato, allineato, ecc...).
""	AutoStretch	Boolean	Adatta l'altezza dell'oggetto in base a quella effettiva del testo in esso contenuto.
""	CalcExpr	""	Impostata a True, non elabora le espressioni contenute nel testo fra parentesi quadre "[]", ma le lascia inalterate. Utile nel caso di testi scientifici.
TfrShapeView	TypeShape	TfrShape	Indica il tipo di forma geometrica da disegnare (shCircle = cerchio, shRectangle = rettangolo, shRoundRect = rettangolo con i bordi arrotondati, shFirstLine = linea con il primo vertice il alto a sinistra, shSecondLine = linea con il primo vertice in basso a sinistra).

TfrCodeBarreView	Param	TfrCodeBarre	<p>Permette di selezionare ogni tipo d'informazione riguardante il codice a barre desiderato. Il tipo di dati ha la seguente struttura:</p> <pre>TfrCodeBarre = Record CcheckSum : Boolean; CshowText : Boolean; CshowCheckSumText : Boolean; Ccadre : Boolean; CtypeBarre : TBarcodeType; Cmodul : Integer; Cratio : Double; Cangle : Double; End;</pre> <p>CcheckSum = visualizza il codice a barre del codice di controllo CheckSum. CshowText = visualizza il codice numerico del codice a barre. CshowCheckSumText = visualizza il codice numerico del codice di controllo (CheckSum). Ccadre = non usato. CtypeBarre = indica il tipo di codice a barre. Cmodul = indica l'ingrandimento del codice a barre (1=normale, 2=doppio, ecc.). Cratio = indica la proporzione fra larghezza ed altezza del codice a barre (non accessibile tramite il modellatore). Cangle = non usato.</p>
TfrGraphView	XTitle	ShortString	Contiene il titolo dell'asse delle ascisse.
""	YTitle	""	Contiene il titolo dell'asse delle ordinate.
""	FieldsTitle	""	Contiene l'espressione che genererà i titoli di ogni gruppo di campi.
""	GraphTitle	""	Contiene il titolo del grafico.
""	FormatY	""	Contiene la stringa di formattazione per i valori delle ordinate.
""	LegendView	Boolean	Impostato a 'True' visualizza la leggenda dei campi.
""	TitleView	""	Impostato a 'True' visualizza il titolo del grafico.
""	XView	""	Impostato a 'True' visualizza il titolo dell'asse delle ascisse.

""	YView	""	Impostato a 'True' visualizza il titolo dell'asse delle ordinate.
""	GraphType	TfrGraphType	<p>Indica il tipo di grafico da visualizzare. Il tipo di dati ha la seguente struttura:</p> <pre>TfrGraphType= (Istograms,Istograms_Lines,Triangles,Triangles_Lines,Points,Points_Lines,Lines);</pre> <p>Istograms = Grafico a istogrammi Istograms_Lines = Grafico ad istogrammi e linee Triangles = Grafico a triangoli Triangles_Lines = Grafico a triangoli e linee Points = Grafico a punti Points_Lines = Grafico a punti e linee Lines = Grafico a linee</p>
""	Memo	TStrings	In questo caso la proprietà 'Memo' non serve a contenere il testo dell'oggetto ma contiene (a gruppi di tre righe) stringhe che indicano in ordine: l'espressione del valore del campo, il colore e l'espressione della legenda del campo.
TfrOLEView (Oggetto OLE)	OleContainer	TOleContainer	Contiene l'oggetto OLE associato.
""	Center	Boolean	Impostata a True centra l'immagine.
""	KeepRatio	""	Impostata a True, mantiene costanti le proporzioni dell'immagine.
""	Stretch	""	Impostata a True, permette di ridimensionare l'immagine a piacimento.

TfrFramedMemoView	Frame	Array[0..3] of TfrFrameLine	<p>Indica il tipo di bordi da utilizzare. In ordine 0=destra, 1=sinistra, 2=inferiore, 3=superiore;</p> <p>Il tipo TfrFrameLine è definito:</p> <pre>TfrFrameLine = packed record Typ: Byte; Width: Byte; end;</pre> <p>Dove per "Typ" si intende il tipo di bordo (0=singolo, 1=doppio) e per "Width" lo spessore.</p>
""	Uguali all'oggetto standard TfrMemoView	Uguali all'oggetto standard TfrMemoView	Uguali all'oggetto standard TfrMemoView.

Si tenga presente che queste stesse proprietà sono accessibili dal modellatore, attraverso il menu contestuale che appare cliccando con il tasto destro su di ogni oggetto.

Struttura delle pagine del report

Struttura delle pagine del report

In questo capitolo analizzeremo come sono strutturate le pagine del report prima di essere preparate (la classe ha il nome di `TfrPage`). Vedremo sia i metodi che le proprietà.

I metodi principali sono due. Rispettivamente sono "Clear" che svuota la pagina corrente e "Delete (n:Integer)" che elimina la pagina selezionata.

Essendo più numerose le proprietà, eccone di seguito un dettagliato elenco...

Proprietà	Tipo	Descrizione
PgSize	Integer	Indica il tipo di foglio utilizzato dalla pagina. Indice numerico dei fogli disponibili.
PgWidth	""	Indica la larghezza della pagina in decimi di mm.
PgHeight	""	Indica l'altezza della pagina in decimi di mm.
ZoomX	""	Indica le proporzioni della larghezza della pagina (100 = normale, 50 = metà, 200 = doppio).
ZoomY	""	Indica le proporzioni dell'altezza della pagina (100 = normale, 50 = metà, 200 = doppio).
ColCount	""	Indica il numero di colonne della pagina. Un numero maggiore di 1 divide la pagina in colonne. Facendo in modo che giunti alla fine della pagina si torni a stampare nella parte superiore destra della successiva colonna, fino a terminare le colonne della pagina corrente. Un po' come per le colonne dei quotidiani.
ColGap	""	Indica la distanza fra ogni colonna (in pixel).
PgMargins	TRect	Serve ad impostare i margini della pagina corrente (in pixel).
PgOr	TPrinterOrientation	Serve ad impostare l'orientamento della pagina corrente (poPortrait = verticale, poLandscape = orizzontale).
PhonFirst	WordBool	Impostata a True, visualizza la banda "intestazione pagina" sulla prima pagina (ovviamente solo qualora fosse inserita nel report).
PfonLast	""	Impostata a True, visualizza la banda "Piè di pagina" sull'ultima pagina (ovviamente solo qualora fosse inserita nel report).
CHCopy	""	Impostata a True, copia l'intestazione della prima pagina di un report a tabella, nelle successive.
PrintToPrevPage	""	Impostata a True, unisce l'ultima pagina preparata, di un report a multipagina, con la prima pagina preparata della pagina successiva.
UseMargins	""	Impostata a True, utilizza i margini impostati, altrimenti considerati come fossero tutti 0.
Objects	TList	Elenco di tutti gli oggetti (TfrView) presenti nella pagina.

Stralci di codice sorgente

In questo capitolo analizzeremo qualche piccolo esempio di codici sorgente che, utilizzando alcuni fra i metodi e le proprietà visti in precedenza, ci permetteranno di capire meglio come poter interagire via codice con FuzzyReport.

Nel seguente esempio imposteremo il margine sinistro di un oggetto rettangolo...

```
// Report è il report che contiene l'oggetto.  
// ObjectByName è un metodo del report che restituisce l'oggetto  
cercato in base al nome 'Nome oggetto'.  
// Left è la proprietà dell'oggetto base che ne indica il margine  
sinistro.  
Report.ObjectByName('Nome oggetto').Left := 100;
```

Nel seguente esempio imposteremo il bordo dell'oggetto come tutto attivo (superiore, inferiore, destro e sinistro)...

```
// Report è il report che contiene l'oggetto.  
// ObjectByName è un metodo del report che restituisce l'oggetto  
cercato in base al nome 'Nome oggetto'.  
// SetAllFrames è il metodo dell'oggetto base che ne attiva tutti  
e quattro i bordi.  
Report.ObjectByName('Nome oggetto').SetAllFrames;
```

Nel seguente esempio renderemo invisibile un oggetto rettangolo...

```
// Report è il report che contiene l'oggetto.  
// ObjectByName è un metodo del report che restituisce l'oggetto  
cercato in base al nome 'Nome oggetto'.  
// Visible è la proprietà dell'oggetto base che indica se  
visualizzarlo o meno.  
Report.ObjectByName('Nome oggetto').Visible := False;
```

Nel seguente esempio caricheremo un report da file, poi lo modificheremo con il modellatore, poi lo salveremo su file...

```
// Report è il report che contiene l'oggetto.  
Report.LoadFromFile ('Prova.frf'); // Carica da file il report.  
Report.DesignReport; // Visualizza il modellatore di report.  
Report.SaveToFile (Report.FileName); // Salva il report su file,  
lasciando inalterato il nome del file di origine.
```

Nel seguente esempio caricheremo da database un report per poi avviarne l'anteprima...

```
// Report è il report che contiene l'oggetto.  
Report.LoadFromDB (Table,10); // Carica dalla tabella 'Table' il  
report memorizzato alla posizione 0.  
Report.ShowReport; // Prepara il report e ne fa l'anteprima.
```

Nel seguente esempio caricheremo da stream un report per poi stamparlo sulla stampante predefinita...

```
// Report è il report che contiene l'oggetto.
Report.LoadFromStream (Stream); // Carica il report dallo stream
'Stream'.
Report.PrintReport; // Prepara il report e lo stampa sulla
stampante predefinita.
```

Nel seguente esempio caricheremo da file un report e ne modificheremo la proprietà di preparazione in due fasi (DoublePass), impostandola a True....

```
// Report è il report che contiene l'oggetto.
Report.LoadFromFile ('Prova.frf'); // Carica da file il report.
Report.DoublePass := True; // Imposta a proprietà che specifica la
preparazione in due fasi, a True.
```

Nel seguente esempio raddoppieremo la larghezza della prima pagina di un report, dimezzandone l'altezza...

```
// Report è il report che contiene l'oggetto.
Report.Pages[0].ZoomX := 200; // Raddoppia la larghezza della
prima pagina del report.
Report.Pages[0].ZoomY := 50; // Dimezza la larghezza della prima
pagina del report.
```

Nel seguente esempio modificheremo i margini della prima pagina del report...

```
// Report è il report che contiene l'oggetto.
Report.Pages[0].pgMargins.Left := 10; // Imposta a 10 (pixel) il
margine sinistro della prima pagina del report.
Report.Pages[0].pgMargin.Top := 50; // Imposta a 50 (pixel) il
margine superiore della prima pagina del report.
Report.Pages[0].pgMargin.Right := 10; // Imposta a 10 (pixel) il
margine destro della prima pagina del report.
Report.Pages[0].pgMargin.Bottom := 50; // Imposta a 50 (pixel) il
margine inferiore della prima pagina del report.
```

Nel seguente esempio svuoteremo la prima pagina del report...

```
// Report è il report che contiene l'oggetto.
Report.Pages[0].Clear; // Cancella il contenuto della prima
pagina del report.
```

Nel seguente esempio elimineremo la seconda pagina del report...

```
// Report è il report che contiene l'oggetto.
Report.Pages.Delete (1); // Elimina la seconda pagina del report.
```

Nel seguente esempio ridimensioniamo un'immagine modificandone le dimensioni ed adattandola a questa nuova forma...

```

// Report è il report che contiene l'oggetto.
// P indica la variabile che conterrà l'immagine 'Immagine',
trovata tramite il metodo 'ObjectByName'.
Var P: TfrPictureBox;
Begin
  P := Report.ObjectByName('Immagine');
  P.Width := 100; // Impostiamo la larghezza a 100 pixel.
  P.Height := 200; // Impostiamo l'altezza a 200 pixel.
  P.Stretch := True; // Attiviamo il ridimensionamento automatico
in base alla nuova forma.
End;

```

Nel seguente esempio cambieremo il colore e l'allineamento orizzontale di un rettangolo di testo...

```

// Report è il report che contiene l'oggetto.
// M indica la variabile che conterrà il rettangolo di testo
'Rettangolo', trovata tramite il metodo 'ObjectByName'.
Var M: TfrMemoView;
Begin
  M := Report.ObjectByName('Rettangolo');
  M.Color := $00FFCC; // Impostiamo il colore.
  M.Alignment := frCenter; // Impostiamo l'allineamento
orizzontale come centrato.
End;

```

Nel seguente esempio impostiamo il contenuto di un rettangolo di testo su di una singola riga, utilizzando la proprietà Caption...

```

// Report è il report che contiene l'oggetto.
// M indica la variabile che conterrà il rettangolo di testo
'Rettangolo', trovato tramite il metodo 'ObjectByName'.
Var M: TfrMemoView;
Begin
  M := Report.ObjectByName('Rettangolo');
  M.Caption := 'Riga unica di testo.' // Impostiamo il testo come
riga unica.
End;

```

Nel seguente esempio impostiamo il contenuto di un rettangolo di testo su più righe, utilizzando la proprietà Memo...

```

// Report è il report che contiene l'oggetto.
// M indica la variabile che conterrà il rettangolo di testo
'Rettangolo', trovata tramite il metodo 'ObjectByName'.
Var M: TfrMemoView;
Begin
  M := Report.ObjectByName('Rettangolo');
  M.MemoClear; // Cancliamo il contenuto.
  M.Memo.Add ('Prima riga'); // Aggiungiamo la prima riga.
  M.Memo.Add ('Seconda iga'); // Aggiungiamo la seconda riga.
  M.Memo.Add ('Terza riga'); // Aggiungiamo la terza riga.
End;

```

Nel seguente esempio controlliamo la presenza o meno di un oggetto dal nome "Oggetto" all'interno del nostro report...

```

// Report è il report che contiene l'oggetto.
// O indica la variabile che conterrà l'oggetto cercato, tramite
l'evento FindObject, che, in caso di fallimento restituisce un
puntatore a Nil.
Var O: TfrMemoView;
Begin
  M := Report.FindObject('Oggetto');
  If M<>Nil Then Beep; // Se viene trovato l'oggetto suona un
Beep.
End;

```

Creare oggetti a run-time

In questo capitolo analizzeremo come creare degli oggetti non dal modellatore ma in fase di esecuzione del programma (a run-time).

La tecnica da adoperare per riuscire in questo intento non è affatto complessa, ed è per di più molto simile per tutti i tipi di oggetti che si desiderano creare.

Per creare oggetti bisogna adoperare la funzione *Function frCreateObject(Typ:Byte; const ClassName:String):TfrView;*

I parametri da passare sono due, il primo parametro indica il tipo di oggetto che si desidera creare, mentre il secondo (usato solo nel caso di oggetti avanzati) indica il nome della classe dell'oggetto.

I possibili valori per il primo parametro sono i seguenti:

```
gtMemo = Rettangolo di testo
gtPicture = Immagine
gtBand = Banda
gtSubReport = SottoReport
gtAddIn = Oggetto avanzato
(In questo caso si userà frCreateObject (gtAddIn, NomeClasseOggettoAvanzato))
```

Ecco di seguito un esempio di codice sorgente che crea un rettangolo di testo e ne setta le principali proprietà...

```
Var Memo: TfrMemoView;
Begin
  // Pagina 0 del report frReport1
  With frReport1.Pages.Pages[0] Do
    Begin // frCreateObject crea un nuovo oggetto.
      // gtMemo = Rettangolo di testo
      Objects.Add (frCreateObject(gtMemo, ''));
      // Assegna a Memo l'ultimo oggetto inserito nel report.
      Memo := Objects.Last;
      Memo.Left := 37;           // Setta il margine sinistro.
      Memo.Top := 54;           // Setta il margine superiore.
      Memo.Width := 719;        // Setta la larghezza.
      Memo.Height := 21;        // Setta l'altezza.
      Memo.Caption := 'Testo';  // Setta il valore del testo.
      Memo.Alignment := frCenter; // Setta l'allineamento.
      Memo.Color := clTeal;     // Setta il colore di fondo.
      Memo.Font.Size := 12;     // Setta la dimensione del font.
      Memo.Font.Style := [fsBold]; // Setta lo stile del font.
      Memo.Font.Color := clWhite; // Setta il colore del font.
    End;
End;
```

Convertire QuickReport

In questo capitolo analizzeremo come convertire i report generati con QuickReport, in report FuzzyReport.

Dalla versione 2.4 FuzzyReport è dotato di un componente che effettua la conversione automatica del QuickReport su cui è posto.

Molte modifiche sono servite per permettere una elevata compatibilità fra i due sistemi di reporting.

Per poter effettuare la conversione basta semplicemente posizionare sul QuikReport interessato il componente TQRtoFRConverter  e poi effettuare un doppio clic su di esso.

Apparirà automaticamente il modellatore tipico di FuzzyReport, con il report completamente convertito.

Si tenga presente comunque, che la conversione non è esattamente del 100% anche se è molto vicina ad esserlo, per il semplice motivo che non ostante le numerose modifiche apportate a FuzzyReport per renderne la struttura interna il più possibile compatibile con quella di QuickReport, la compatibilità non è totale.

Si potranno quindi notare piccole imperfezioni o mancanze nel caso di report molto complessi.

Licenza d'uso

Distribuzione del manuale originale. La distribuzione libera di questo manuale, inteso come originale, è consentita, anzi consigliata ed incentivata.

Distribuzione di copie modificate del manuale originale. La distribuzione di copie modificate di questo manuale, implica l'obbligo di rendere queste modifiche di libero dominio, inviando, prima dell'inizio della distribuzione delle copie modificate, una copia del testo modificato all'autore del manuale originale (Dell'Aria Fabio). Solo dopo avere ricevuto dall'autore, l'autorizzazione a diffondere la copia modificata, sarà possibile procedere alla divulgazione. Questo non per limitare chi desideri migliorare e diffondere questo manuale, ma solo per evitare inutili decentralizzazioni e diminuire al minimo la possibilità di inserire, all'interno del manuale modificato, concetti poco chiari o peggio inesatti, che nuocerebbero gravemente alla diffusione dell'oggetto di questo manuale.

Limiti di responsabilità. In nessun caso l'autore di questo manuale, o gli autori delle modifiche ad esso apportate, saranno responsabili per i danni (inclusi, senza limitazioni, il danno per perdita o mancato guadagno, interruzione dell'attività, perdita di informazioni o altre perdite economiche) derivanti dall'uso o dalla incapacità di fare uso del manuale in oggetto, anche nel caso in cui l'autore in questione sia stato avvertito della possibilità di tali danni.

Tacito consenso. L'uso di questo manuale (per qualsiasi scopo, dalla copia alla semplice lettura) è da considerarsi come approvazione tacita dei termini di questo documento.

Controversie legali. Questo contratto è disciplinato dalle leggi italiane. Per ogni controversia resta competente il Foro di Palermo.

Variazioni. Qualsiasi variazione ai termini di questo documento è valida soltanto se prodotta per iscritto dall'autore della copia originale del manuale (Dell'Aria Fabio).