

Occorre distinguere tra block e rblock. Un rblock (recursive block) e' la descrizione di un block.

Quando si crea un SDF, si creano tutti i block, ed ad ogni block sono settati i parametri del corrispondente rblock.

Gli rblock sono sequenziali, per cui un rblock puo' essere fisso o variabile (cioe' ha sempre lo stesso numero di blocchi oppure questi variano), ma non si possono mescolare block di rblock diversi.

Ogni rblock ha una parte fissa ed una variabile. La parte fissa e' composta sempre dallo stesso numero/tipo di field, la parte variabile ha elements (che sono a loro volta rblock) che si ripetono per un numero di volte fisso o variabile.

Il SourceDataField e' **omposto** da un unico rblock. Quindi l' **SDF** e' un contenitore per un unico rblock, da cui poi si costruiscono tutti gli altri.

Caratteristiche rblock

Indica il numero di rblock:

```
int numberOfRBlocks;
```

Indica la lista di blocks

```
SDFRBlock* block;
```

Parametri esterni rblock

Il file name

```
char** rblockFilename;
```

Indica se un rblock ha un numero fisso o variabile di block.

```
bool* numberOfBlockFixed;
```

Indica il numero massimo di blocchi presenti per un certo rblock, letto da file.

```
word* maxNumberOfBlock;
```

Indica il numero effettivo di block presenti per un certo rblock, settato durante il run time se numberOfBlockFixed = false, settato durante la creazione se numberOfBlockFixed = true.

```
word* numberOfRealDataBlock;
```

Per quanto riguarda l'identificazione della posizione in cui e' presente il campo che indica il numero di blocks per un certo rblock nel caso di numberOfBlockFixed = false

```
word* headerLevelOfNBlockIndex
```

```
word* indexOfNBlock;
```

```
word* subFromNBlock;
```

Quindi un rblock e' composto dalla parte fissa, da un indicatore che dice quanti rblock sono presenti e da una lista di block e dai loro parametri esterni.

Caratteristiche dei block

Un block

Indica se la parte fissa e' presente
bool fixedPresent;

Indica se la parte variabile e' presente:

bool variablePresent;

Indica se l'rblock ha gli element in numero fisso o variabile:

bool rBlockVariable;

Indica il numero massimo di blocchi possibili

int nblockmax;

Costruzione

Per prima cosa bisogna costruire il SDF, che e' sua volta un rblock con i seguenti parametri esterni:

```
numberOfRBlocks=1  
numberOfBlockFixed[0]=true  
maxNumberOfBlock[0]=1  
numberOfRealDataBlock[0] = 1  
indexOfNBlock[0] = 0 (default)  
subFromNBlock[0] = 0;  
rblockFilename="Source Data Field"
```

block -> si crea un solo rblock

Si leggendo i suoi parametri dalla sezione [RBlock Configuration] del .packet. La costruzione e' ricorsiva.

Si costruiscono tutti i block caricandone i parametri a partire dall' rblock corrispondente.

A questo punto si esce per permettere alla routine di lettura di leggere gli identifiers.

Costruzione rblock

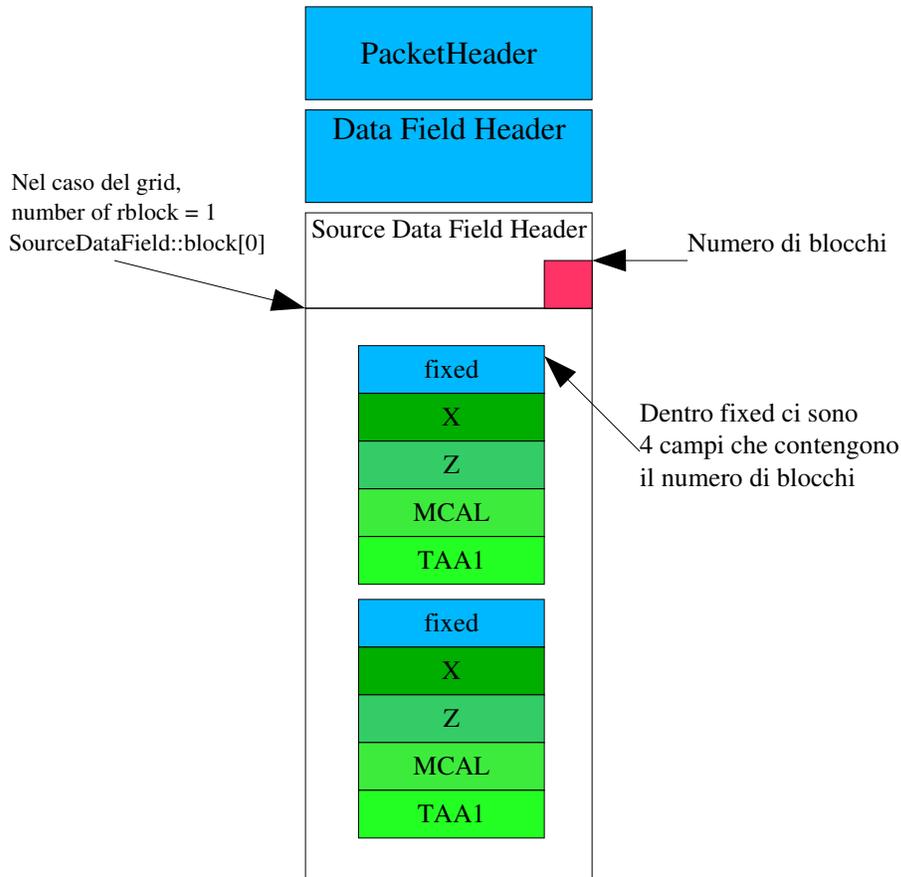
SI deve leggere se il blocco ha una parte fissa e una parte variabile.

Poiche' un rblock deriva da un SourceDataField, questo comporta i seguenti vantaggi:

- si eredita la struttura per la gestione di strutture organizzate a blocchi di diverso tipo (rblock)

- si eredita il concetto di elements (il numero di elementi per la parte variabile di un rblock, che sono a loro volta rblock)

Struttura di esempio: GRID packet



Attualmente non è possibile ripetere più volte il blocco bianco, quello che rappresenta effettivamente il SDF, ma dal punto di vista della struttura generale non cambia nulla, perchè è possibile ottenere lo stesso effetto mettendo a zero la parte fissa, lavorando solo su quella variabile e puntando come contatore di blocchi (nel caso variabile) ad un campo del Data Field Header.

Numero reali di blocks

Il metodo

```
void SourceDataField::setNumberOfRealDataBlock(word number, word rblockIndex)
```

consente di settare il numero di blocchi nel caso in cui numberOfBlockFixed[i]=false e non deve superare maxNumberOfBlock[i].

Come detto in precedenza, il SDF dispone di un unico blocks di un determinato tipo rblock. Questo block e' poi composto da altri blocks (dei proprio rblock type).

In sostanza, occorre vedere questo tipo di SDF come un insieme di blocks strutturati ricorsivamente. Tuttavia in questa visione e' presente un problema. Durante la creazione sono stati creati un numero di blocchi pari alla numero totale possibile, quindi e' necessario settare il numero reale di blocchi per ogni rblock.

Considerando inoltre che nel SDF e' presente un solo blocco, il suo accesso deve essere trasparente, nel senso che se uno chiama

```
p->dataField->sourceDataField->setNumberOfRealDataBlock(10, 1)
```

occorre settare il valore dell' rblock (unico) contenuto nel SDF.