

Suite 2/599 Doncaster Road Email: abmsupport@advancedbusinessmanager.com.au

Doncaster, VIC 3108

Tel: (03) 9532 4199 Fax: (03) 9840 1799

ACN 073 061 677

Website: www.advancedbusinessmanager.com.au

Support Bulletin # 91

Table of Contents

Using Views in Crystal Reports.....	2
What is a view?	2
How to create a view.....	2
Create a select statement with the desired fields named in the result set	2
Create View syntax	2
Call the View	2
Using views in a customized Crystal Report: Real World Example	3
Problems:.....	3
Solution:.....	3
SQL Script:.....	4
Crystal Report changes	4

*THIS BULLETIN IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. The information contained in this document represents the current view of Advanced Business Manager on the issues discussed as of the date of publication. Because Advanced Business Manager must respond to change in market conditions, it should not be interpreted to be a commitment on the part of Advanced Business Manager and Advanced Business Manager cannot guarantee the accuracy of any information presented after the date of publication.

INFORMATION PROVIDED IN THIS DOCUMENT IS PROVIDED 'AS IS' WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND FREEDOM FROM INFRINGEMENT.

The user assumes the entire risk as to the accuracy and the use of this document.

Using Views in Crystal Reports

What is a view?

A view is like a table, but it doesn't physically exist. In affect its a virtual table. A view is created by a using a query joining one or more tables together to form a result set.

How to create a view

Views can be simple or complex, it just depends on what you are trying to accomplish. The core of a view is a SQL select statement. If you can create a select statement, you can create a view.

Create a select statement with the desired fields named in the result set

```
select transactionid, transactiontype, batchnumber, accountingdate
from transheaders
where transactiontype = 'sx'
```

	transactionid	transactiontype	batchnumber	accountingdate
1	SXPI00001<XXXXXXXXXXXXXXXXXX	SX	1	2009-11-09 00:00:00.000
2	SXPI00003<XXXXXXXXXXXXXXXXXX	SX	6	2009-12-01 00:00:00.000
3	SXPI00004<XXXXXXXXXXXXXXXXXX	SX	9	2009-12-03 00:00:00.000
4	SXPI00005<XXXXXXXXXXXXXXXXXX	SX	10	2009-10-14 00:00:00.000
5	SXPI00006<XXXXXXXXXXXXXXXXXX	SX	16	2009-12-10 00:00:00.000
6	SXPI00007<XXXXXXXXXXXXXXXXXX	SX	19	2010-01-22 00:00:00.000

Create View syntax

```
CREATE VIEW NewViewName AS
select transactionid as tranid,
       transactiontype as trantype,
       batchnumber as batchnum,
       accountingdate as accdate
from transheaders
where transactiontype = 'sx'
```

Call the View

Once the view is created in the database it can be used just like a table. You can create SQL scripts and create links to it in Crystal Reports but unlike a static table, the result set of a view is dynamically built when called.

```
Select * from NewViewName
```

Using views in a customized Crystal Report: Real World Example

Linking to custom views in Crystal Reports may solve many difficult or even impossible relationships such as conditional links.

Request is that the Client wants to print the payment method on the Supplier Remittance.

Problems:

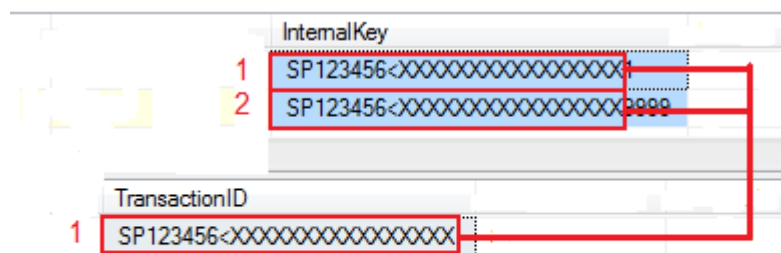
First problem is that the Supplier Remittance Z table has no link to the transdetails table. The internal key in the zsupplier_remittance table has some extra numbers on the end of transaction id, so it's impossible to link the two tables together. We need to link the first 25 characters of the two fields together.

ount	_RemainingAmount	InternalKey	UserNo	Mec
1	74.00	SP123456<XXXXXXXXXXXXXXXXXXXX1	1	P
2	NULL	SP123456<XXXXXXXXXXXXXXXXXXXX9999	1	P

UniqueID	TransactionID	TransactionType	AccountID
187	174	SP	S1XXXXXXXXXX

Only the first 25 characters of the zsupplierremittance.internalkey is a valid link to the transdetails.transactionid.

The second problem is that even if we could establish a relationship, it is not a one to one relationship and we will end up with multiple lines in the result set. A symptom of this type of link relationship is when the report prints multiple copies of one transaction line. We need to setup a one to one relationship between the two tables.



Even if we could link these two tables the relationship is a one to many, not one to one.

Solution:

Create a view with the relationship between the zsupplier table and the transheaders table using the first 25 characters and insure the result set has only one result for each transaction.

We can even eliminate having to add the transheaders table all together by adding the payment method field to our view. This way we only need to add the view to the report.

SQL Script:

```

Create view bms_SupplierRemitMethod as
SELECT    TRANSHEADERS.TransactionID AS Tansid,
          ZSUPPLIER_REMITTANCES.InternalKey AS SuppTranid,
          TRANSHEADERS.PayMethod AS paym
FROM      TRANSHEADERS INNER JOIN ZSUPPLIER_REMITTANCES
ON        LEFT(ZSUPPLIER_REMITTANCES.InternalKey,25) = TRANSHEADERS.TransactionID
WHERE     (RIGHT(RTRIM(dbo.ZSUPPLIER_REMITTANCES.InternalKey), 1) = '1')

```

Select Statement:

This is what fields we want in our result set.

```

TRANSHEADERS.TransactionID AS Tansid,
ZSUPPLIER_REMITTANCES.InternalKey AS SuppTranid,
TRANSHEADERS.PayMethod AS paym

```

Join Statement:

The join is on from the first 25 characters of the Internal Key field on the ZSupplier_Remittance to the transheader transactionid. We are using the Left function to do this.

```
LEFT(ZSUPPLIER_REMITTANCES.InternalKey,25) = TRANSHEADERS.TransactionID
```

Where Statement:

The where clause gives us one record for each invoice from the ZSupplier_Remittance table by linking only on the Header Record. We know that the zsupplier_Remiattance table data always has a header record with a value of '1' at the end.

```
(RIGHT(RTRIM(dbo.ZSUPPLIER_REMITTANCES.InternalKey), 1) = '1')
```

Crystal Report changes

You can now add the view, create a link, and use the fields in the report just like a normal table in Crystal Reports.

