

## Creating a Balance Forward Report

---

**Product:** ARPEGGIO™  
R&R Report Writer®  
**Version:** ARPEGGIO 2.0  
R&R 8.0  
**Host:** N.A.

**NIC:** N.A.  
**Interface:** N.A.  
**Oper Sys:** Microsoft® Windows® 95  
Microsoft Windows NT®

---

### Summary

This technical bulletin provides an example for creating a balance forward report. The following statement shows a sample balance forward statement.

STATEMENT				
Dr. Arnold Toothacher, D.M.D. 974 Edgell Road Marlboro, MA 01752 (508) 974-9266				
Mr. Phil Molar 144 Juniper Hill Lane Framingham, MA 01701				
Date	Professional Service	Charge	Payment	Balance
06/01/94	Starting Balance			146.00
06/06/94	Adult Prophy	60.00		206.00
06/06/94	Full Mouth x-ray	15.00		221.00
06/15/94	Check #436		100.00	121.00
06/15/94	Silver Amalgam, Tooth 12	55.00		176.00
06/15/94	Silver Amalgam, Tooth 13	55.00		231.00
	Pay this amount			231.00
*** Please disregard this bill if payment has been made. ***				

Figure 1. Balance Forward Statement

### Balance Forward Accounting

In balance forward accounting, you maintain a list of charges and payments for each account. To find out the balance at any point in time, you add the charges, add the payments, then

subtract total payments from total charges. A billing statement is sent out every month with any balance carried forward from the previous statement.

## Storing Data

To follow this example, you first need to know a bit about the structure, contents, and relationships of the database tables. Then the special features related to the balance forward report will be easier to understand.

The patient's name and address are stored in a table named Patient. The charges and payments are stored in a table named Items. The Patient table is related to the Items table by storing the patient record number in each item record. In this example, the Items table contains the following records for patient Phil Molar.

PATIENT	DATE	SERVICE	AMOUNT
1	04/05/94	Oral Surgery	555.00
1	05/03/94	Insurance Payment	-409.00
1	06/06/94	Adult Prophy	60.00
1	06/06/94	Full Mouth X-Ray	15.00
1	06/15/94	Check #436	-100.00
1	06/15/94	Silver Amalgam, Tooth 12	55.00
1	06/15/94	Silver Amalgam. Tooth 13	55.00

Figure 2. The Item database table.

## Calculating Dates

In order to print the statements on a monthly basis, Dr. Toothacher created calculated fields to identify the starting and ending dates. He can modify these calculations each month. The following calculated fields hold the starting and ending dates.

```
Start_Date = CTOD("06/01/94")  
End_Date = CTOD("06/30/94")
```

The problems in creating a balance forward report are how to include in the starting balance all of the items before the starting date, and how to print only those items that fall between the starting and ending dates. In this example, the starting and ending dates are described in the calculations above.

The essential features of the report are contained on two record lines. Let's first discuss the second record line, which will print each item between the starting and ending dates. The Date

and Service fields come directly from the database. The Charge and Payment fields are calculated from the Amount field.

In the database table shown in Figure 2, note that charges are entered as positive amounts and payments are entered as negative amounts. This allows you to create a calculated field expression that tests whether a number belongs in the Charge or Payment column. The following calculated fields perform this function.

```
Charge = IIF(AMOUNT>0, AMOUNT, 0)
Payment = IIF(AMOUNT<0, AMOUNT, 0)
```

After inserting the fields on the report, choose the Format menu and select Field. take the "x" out of the Show-Zero Option for each of these fields. This makes the Charge field blank for Payment records and the Payment field blank for Charge records.

## Calculating a Running Balance

The last field on the second record line is a total field containing the running balance. This total field is defined as follows. Note that the report is grouped by the Patient field in order to calculate a total for each patient. Placing this group total in the body area allows you to see the running total.

```
RunningBal = Sum(AMOUNT, PATIENT, Every, Running)
```

The logical expression used to control printing of the item line (the second record line) tests whether the item date is between the starting and ending dates. Define the following calculated field and then assign it to the second record line using the Format, Band Line Properties, Logical command, and select Print when True.

```
Print_Item = INRANGE(DATE,START_DATE,END_DATE)
```

When this field evaluates to true, the associated line will print.

## The Starting Balance

The starting balance line is positioned before the line containing charges and payments. This ordering of lines is required so that the starting balance line is the first line to print. The date field is simply the calculated field Start\_date. The text "Starting Balance" is typed directly on the report.

The starting balance field is calculated by subtracting the amount in the first item record from the running balance. In this example, \$206 minus \$60 leaves \$146. Define this calculated field as follows.

```
StartBal = RunningBal - AMOUNT
```

The trickiest part of the report is developing a line logical expression to print the starting balance line only once. This expression needs to test whether we are on the first item and whether the balance isn't zero. If both conditions are true, the expression should return true.

The best way to test for the first item is to create a total field that counts items between the starting and ending dates. We already have a field we can total to give us the desired count: Print\_Item. Define the following total field.

Item\_Count = Sum(Print\_Item, Patient, Every, Running)

Recall that Print\_Item is a logical field that will be false (zero) for items outside the desired date range and true (one) for items within the desired date range.

Now you can create a calculated field that controls the printing of the starting balance line using the following definition. Apply this expression to the first body line using the Format, Band Line Properties, Logical command.

Print\_SBal = Item\_Count=1 .AND. StartBal <> 0

The following table of field values for each record may help illustrate the process by which the line logical expression controls when each record line prints. If Print\_SBal is true, the starting balance line will print.

DATE	AMOUNT	Balance	Print_Item	Item_Count	Print_SBal
04/05/94	555.00	555.00	false (0)	0	false
05/03/94	-409.00	146.00	false (0)	0	false
06/06/94	60.00	206.00	true (1)	1	true
06/06/94	15.00	221.00	true (1)	2	false
06/15/94	-100.00	121.00	true (1)	3	false
06/15/94	55.00	176.00	true (1)	4	false
06/15/94	55.00	231.00	true (1)	5	false

---

All trademarks are the property of their respective owners. The information contained in this technical bulletin is subject to change without notice. Liveware Publishing Inc. provides this information "as is" without warranty of any kind, either expressed or implied, but not limited to the implied warranty of merchantability and fitness for a particular purpose. Liveware Publishing may improve or change the product at any time without further notice; this document does not represent a commitment on the part of Liveware Publishing. The software described in this document is furnished under a license agreement or nondisclosure agreement. The software may be used or copied only in accordance with the terms of the licensing agreement.