Chapter 8
Working with Calculated Fields

Introduction

This chapter explains how to develop and use calculated fields in R&R. This information is presented in the following sections:

- ❑ Creating a Calculated Field
- ❑ Modifying a Calculated Field
- ❑ Deleting a Calculated Field
- ❑ Purging Unused Calculated Fields
- ❑ Calculated Field Expression Syntax

R&R enables you to create fields called calculated fields whose values are calculated as a report is generated, rather than being stored in the database. After you have defined a calculated field, you can:

- ❑ Insert it on a report layout;
- ❑ Use it as a linking field;
- ❑ Use it as a sort or group field;
- ❑ Include it in a report query;
- ❑ Use it in other calculated or total field expressions.

Here are just a few examples of how you can use calculated fields:

- ♦ In an invoice report, you can create a calculated field that multiplies the value in a PRICE field by the value in a QUANTITY field to provide a line item total.
- ♦ In a personnel report, you can create a calculated field that concatenates the values in the FIRSTNAME and LASTNAME fields. You can use this field as a linking field to establish a relation with a table that has a combined NAME field.
- ♦ In an accounts receivable report, you can create a calculated field that subtracts the value in a DUEDATE field from the value in a TODAY field. You can use this field in a query to print a report of all invoices more than 30 days outstanding.
- ♦ In a sales report, you can create a calculated field that computes monthly sales and use that field in creating various total fields to provide summary information for the year.

Creating a Calculated Field

To begin creating a calculated field, select Calculations ⇒ Calculated Field or the Calc button on the Standard Toolbar. If no calculated fields have yet been defined for the current report, the New Calculation dialog appears (see Figure 8.1). If one or more calculated fields have already been defined, select New in the Calculated Fields dialog to display the New Calculation dialog.

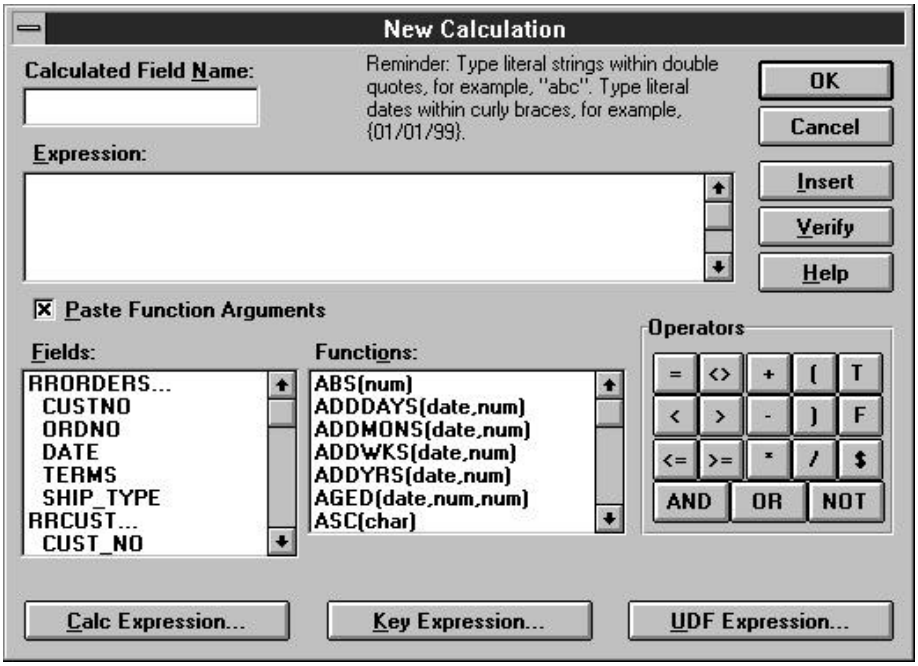

Figure 8.1 New Calculation Dialog Box

Follow these general procedures to create a calculated field:

1. In the Calculated Field Name box, enter a unique field name.
2. Either enter the expression in the Expression box or select fields, functions, operators, and/or expressions.
3. Optionally select Verify to determine whether the expression syntax is correct.
4. When the expression is correct, select OK.

Entering a Field Name

In the "Calculated Field Name" edit box, enter a field name of up to 10 characters. The name can contain letters, numbers, or the underscore character; it must start with a letter. The field name must be unique and cannot contain spaces.

By using a combination of upper and lower case, you can enter calculated field names that are easily distinguishable from the names of table fields.

Entering a Field Expression

Enter the expression for a calculated field using either or both of the following methods:

- Type the expression directly in the Expression edit box.
- Use the list menus and buttons to select from available fields, functions, operators, calculated field expressions, key expressions, or user-defined function expressions.

The expression for a calculated field is a formula that determines how the field's value is calculated. For example, the value of a calculated field whose expression is QUANTITY * PRICE will be the value in the QUANTITY field multiplied by the value in the PRICE field.

The syntax of R&R calculated field expressions is similar to the syntax of expressions in Xbase languages. These expressions can contain functions, operators, constants, database fields, memo fields, total fields, and other calculated fields.

R&R provides standard Xbase functions useful for reports, as well as many additional functions that are available only in R&R. R&R also provides standard arithmetic, string, relational, logical, and date operators, as well as use of parentheses. In addition, you can define your own user-defined functions (UDFs) that can be saved and used in calculated fields in any report just like predefined functions.

See the **Calculated Field Expression Syntax** section of this chapter for information about the components and the structure of R&R expressions.

Typing the Expression

You can type a calculated field expression by entering the field names, constants, operators, and/or functions that make up the expression directly in the Expression edit box. If the expression contains a predefined or user-

defined function, you can type as few as the first four letters of the function (for example, cmon for CMONTH).

Using the List Menus and Buttons

You can easily create even a complex calculation by selecting your choices from the Fields and Functions list boxes, the Operators group box, and from the list boxes displayed when you select the Calc Expression, Key Expression, or UDF Expression button. Using this method reduces the amount of typing required; as a result, you can build expressions more quickly and accurately.

Selecting Fields

To insert a field into the expression, select from the Fields list box, which displays the fields from the currently related tables and currently attached text file (if any), as well as all calculated and total fields created in R&R. To place the selected field into the expression, select Insert (or double-click on the name).

Selecting Functions

To insert a function into the expression, select from the Functions list box, which displays a menu containing more than 100 predefined functions, as well as any user-defined functions you have created. Use the Help system if necessary to find the correct syntax for each function.

To place a function in the expression, select it and choose Insert (or double-click on the name). If the function requires one or more arguments, select or enter them as necessary.

Paste Function Arguments

You can use the "Paste Function Arguments" setting as an aid to providing function arguments. When "Paste Function Arguments" is On (the default), R&R inserts symbols representing the function's arguments into the expression. The symbol for the first argument is highlighted; any item you insert or type as an argument (a field or another function, for example) replaces the highlighted argument symbol.

For example, suppose that you have selected the SUBSTR function and inserted it into your calculated field expression. With "Paste Function Arguments" On, the function is inserted as follows:

```
SUBSTR(char,num,opt num)
```

In this example, the symbols in parentheses after the function name represent the arguments to SUBSTR: "char" represents a required character argument; "num" a required numeric argument, and "opt num" an optional numeric argument. If you select a character field as the first argument, the field name replaces "char" in the expression. You can then supply the other function arguments as needed.

When "Paste Function Arguments" is Off, R&R inserts function names without supplying argument symbols. Instead, the function name is inserted followed by open/close parentheses — for example, SUBSTR( ).

Selecting Operators

To insert an operator into the expression, click on the appropriate operator from the Operator group box. See Figures 8.3, 8.4, 8.5, 8.6, and 8.7 in this chapter for explanations of these operators and descriptions of their use in expressions.

Copying Other Calculated Expressions

You can easily copy other calculated field expressions; as a result, in some cases you can simply modify an existing expression rather than starting from scratch. To insert the expression of a calculated field that you have previously defined or the condition expression from a conditional total, select the Calc Expression button. R&R displays a list of calculated field names; as you highlight each field, its expression is also shown. Select the calculated field whose expression you want to copy and select OK (or double-click on the field name).

Selecting Key Expressions

To insert the key expression of an index file, select Key Expression. Then select an index file from those listed or enter the name of an index file in the edit box. If necessary, select a different drive and/or directory to select from a list of index files in a different location. If you have selected or entered a multiple index file (such as MDX or CDX), you must also select or enter the name of an index tag. Select OK to insert the key expression.

Selecting a UDF Expression

To insert the expression of a user-defined function (UDF), select User Function Expression. R&R displays a list box of UDF names. As you move the cursor from name to name, each UDF's expression displays. Select the UDF expression you want to insert and select OK (or double-click). For more information on UDFs, see Chapter 10, "Using Functions."

Verifying Expression Syntax

When you have completed your expression, select Verify. If you have entered an expression incorrectly (for example, if you have used an unrecognized field or operator), R&R will notify you with an error message. When possible, the cursor will also be positioned at the place in the expression where the error occurred. Edit the existing expression. When the expression is complete and correct, select OK.

Modifying a Calculated Field

You can use either of the following procedures to open the Edit Calculation dialog box to edit a calculated field:

- Either right-click on the field and select Properties or select the field and press F4 to open the Field Properties dialog box; then select Expression.
- Select Calculations ⇒ Calculated Field and select the field to edit from the list box. Then select Edit.

The Edit Calculation dialog is the same as the New Calculation dialog, except that it shows the selected field's name and displays its expression in the Expression edit box.

Modifying the Calculated Field Expression

To modify the existing expression, position the edit cursor as necessary to insert or delete elements. To delete a portion of the expression, drag the mouse over that portion and press Delete. Following the procedures described in the section on creating a calculated field, revise the expression as necessary. When you are done, select OK.

If the calculated field's data type changes as the result of editing and the field is used in another calculated field expression, R&R displays a list of calculated fields that will be affected by your change. The fields on this list will either change their data types or become impossible for R&R to evaluate as a result of your change. At this point, you can select Cancel to cancel the change or OK to make the change. Selecting OK will cause the affected fields that R&R cannot evaluate to be flagged with question marks in the Field Menu. If the flagged fields are used in your report, you will have to edit their expressions before you can print or view the report.

If the field you are editing appears in a query, is totaled, or is a calculated linking field, you will not be allowed to change the data type of its result.

Because R&R estimates the width of a calculated field, and because editing a calculation does not automatically change this estimated width, you may need to adjust the width of your calculated fields using Format ⇒ Field. For instructions on changing field width, see Chapter 5, "Working with Fields."

Adding a Field Comment

You can enter a brief comment (up to 99 characters) to serve as a "plain English" explanation of a calculated field. To do so, select Calculations ⇒ Calculated Field, highlight the field name, and select Comment to display the Field Comment dialog. Enter the explanation in the Comment box and then select OK. When a calculated field has a comment attached to it, right-clicking in the Status Bar toggles between field name/expression and comment.

You can also enter or edit a field comment using the Field Properties dialog box.

Deleting a Calculated Field

To delete a calculated field, open the Calculated Fields dialog box. Select the field you want to remove; then select Delete.

If the field you are trying to delete is totaled or used in another calculation, R&R lists all calculated and/or total fields that will be affected by the deletion. At this point, you can select Cancel to retain the field or OK to delete it. Selecting OK will cause the affected fields to be deleted. You cannot delete calculated fields used as linking fields without first deleting the table relations that use them.

Purging Unused Calculated Fields

To remove all unused calculated fields from your report definition:

1. Select Calculations ⇒ Purge Calculations.

   The Purge Calculations dialog box appears (see Figure 8.2).

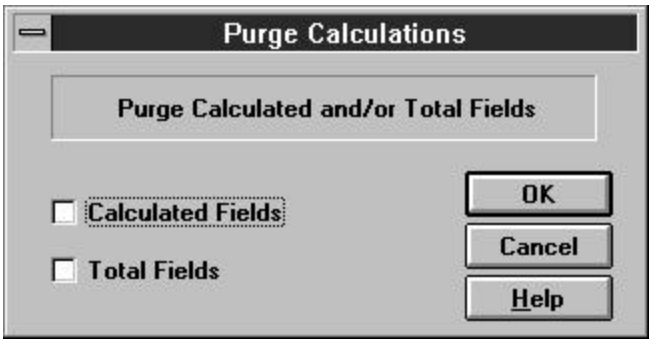

Figure 8.2 Purge Calculations Dialog

2. Click the Calculated Fields check box (an X appears in the box).
3. Select OK.

   R&R lists the fields that will be removed from the report definition and prompts you to confirm.
4. Select OK to remove the listed fields or Cancel to return to the Purge Calculations dialog without removing the fields.

Calculated Field Expression Syntax

An expression is a formula that is used to compute the value of a calculated field using fields, constants, operators, and functions. For example, the expression SUBTOTAL*DISCNT calculates discount by multiplying the value in the subtotal field by that in the discount rate field.

To take another example, the following expression calculates the number of months between the dates in the PAYDATE and INVDATE fields:

```
MONSBTWN(PAYDATE,INVDATE)
```

This section explains calculated field expressions by:

- Describing the types of expressions;
- Explaining the parts of an expression: fields, constants, operators, and functions;
- Explaining how R&R handles errors in evaluating expressions.

Types of Expressions

Expressions are classified by the data type of the results they produce. Based on this categorization, you can develop the character, numeric, date, logical, and memo expressions.

Character Expressions

A character expression produces characters or character strings. For example, "Dear "+FIRSTNAME–"," is a character expression. The character string it produces will be a salutation such as “Dear John,”. Note that character strings can also include numbers and punctuation marks.

Numeric Expressions

A numeric expression produces numbers. For example, 3%TOTAL is a numeric expression. The number it produces will be 3 percent of the value of the TOTAL field. Numeric expressions can also be used as logical expressions; a numeric expression is considered to be true when it has a non-zero value and false when it has a zero value.

Date Expressions

A date expression produces dates. For example, ORDERDATE+30 is a date expression. The date it produces will be 30 days later than the date in the ORDERDATE field.

Logical Expressions

A logical expression produces a logical value of true or false. For example, BALANCE>100 is a logical expression. The expression produces a true value if the value in the balance field is greater than 100, a false value if the balance is less than or equal to 100. Logical expressions can also be used as numeric expressions. In this case, the true value is one and the false value is zero.

Memo Expressions

A memo expression produces a memo field. Only expressions that contain memo fields can produce memo fields. For example, you can use the IIF( ) function to return one of two memo fields based on a specified condition.

The following expression means that if the balance equals zero, return the THANKS memo field; otherwise, return the SENDCASH memo field:

```
IIF(BALANCE=0,THANKS,SENDCASH)
```

Parts of an Expression

Expressions are made up of one or more of the following:

- Fields
- Constants
- Operators
- Functions
- Wildcards

For example, in the expression SUBTOTAL*DISCNT, SUBTOTAL and DISCNT are fields and * is the multiplication operator. In the expression ADDDAYS(INVDATE,30), ADDDAYS is a function that adds a given number of days to a date, INVDATE is a field, and 30 is a constant. Each of these components is explained in more detail in the following sections.

Fields

Expressions can include data fields of any type (character, numeric, logical, date, memo), total fields, and other calculated fields. For example, in the expression SUBTOTAL*DISCNT, the SUBTOTAL field might be a total field that sums line item totals.

The expression for a calculated field can even contain the calculated field itself. For example, you can create a calculated field called RUNTOTAL that keeps a running total of the AMOUNT field by using the expression RUNTOTAL+AMOUNT. The name of the calculated field is used in the expression that defines it.

Field Names with Table Aliases

If several tables contain fields with the same name, the field name used in an expression must be preceded by a table alias. For example, if your report uses a customer table and a product table that both have a field called NAME, the NAME field must be preceded by a table alias that identifies the table it belongs to. If RRCUST and RRPROD are the table aliases, you would enter either RRCUST->NAME or RRPROD->NAME in the expression.

Field Names with Special Characters

If an expression contains the name of a field that includes one of the special characters listed below (possible in the case of memo fields from text files), you must precede the character with a backslash to indicate that the character is part of the field name. For example, to include the EMP# field in an expression, you must type EMP\#.

Include a backslash before each of the following characters if it appears in a field name:

+ - / * ^ % ( ) , = # < > $! " ' [(space)

If you insert a field whose name includes any special characters that could be interpreted as operators in the expression (for example % or >), R&R will automatically insert a backslash before the special character to indicate that the character is part of the field name.

Constants

A constant is a value you specify as part of an expression. For example, if the expression for a field that calculates interest is balance times interest rate, the figure that represents the interest rate can be a constant. In the expression BALANCE*.08, .08 is the interest rate constant.

R&R recognizes four types of constants:

- Numeric constants
- Character constants
- Logical constants
- Date constants

Numeric Constants

A numeric constant is a number that may contain a decimal point and may be preceded by a plus or minus sign. For example, 3.1415927 can be used as a constant in expressions that require the value of pi to seven decimal places.

Character Constants

A character constant is any character or string of characters in the ANSI character set, enclosed in double quotes, single quotes, or square brackets. For example, the expression "Dear "+FIRSTNAME–"," will return a value like “Dear John,”.

Note that if you use a backslash, question mark, or asterisk in a character value that is the right-hand element in a comparison, you must precede the special character with a backslash to tell R&R to treat the character literally, For example, use FIELD = "*" to test for records in which FIELD contains only an asterisk. Without the backslash, R&R will treat the asterisk as a wildcard character.

Logical Constants

A logical constant is a true or false value; for example, .T. for true and .F. for false. Upper or lower case may be used. Valid logical constants are .T., .F., True, False, On, Off, Yes, No. For example, the following expression will return a true value if the value in the CHILDREN field is greater than 0, and a false value if it is less than or equal to 0:

```
IIF(CHILDREN>0, .T. , .F. )
```

Date Constants

You can produce date constants using the following format:

```
{<date-value>}
```

For example, the following expression adds 30 days to the literal date enclosed in curly braces:

```
ADDDAYS({06/17/1995},30)
```

You can also use the CTOD( ) function, which turns character strings into dates, to produce date constants. For example, in the expression CTOD("3/17/95")–CTOD("1/1/95"), the CTOD( ) function turns the two strings in quotation marks into date constants, which are then subtracted to calculate the number of days between the two dates. For more information on CTOD, refer to Chapter 10, “Using Functions.”

Operators

Operators are symbols that perform operations within an expression. For example, in the expression SUBTOTAL*DISCNT, the asterisk is a symbol that performs the operation of multiplication.

R&R provides five types of operators:

- Arithmetic
- Date
- Character (string)
- Relational

- Logical

Arithmetic Operators

R&R provides the arithmetic operators shown in Figure 8.3.

Operator	*Description*
+	Addition
-	Subtraction
/	Division
*	Multiplication
%	Percentage
^ *or* **	Exponentiation
(*and*)	Open/close parenthesis for grouping

Figure 8.3 Arithmetic Operators

The addition, subtraction, division, and multiplication operators are used in R&R expressions just as they are used in algebraic expressions. The percentage operator is used as if it were "percent of," as in the expression 10 % 25 = 2.5. The numbers on either side of the percent sign are multiplied together and divided by 100.

The exponentiation operator, which raises a quantity to a power, is placed after the quantity and before the power, as in the following expression for calculating the balance of a compound interest account:

```
BALANCE*(1+INTEREST)^PERIODS
```

The parenthesis operators are used to group operations, indicating the order in which they will be performed or evaluated.

Without parentheses, arithmetic expressions are evaluated by R&R in the following order:

1. Unary plus and minus (positive and negative signs)
2. Exponentiation
3. Multiplication, Division, and Percentage
4. Addition and Subtraction

For example, the expression 1+4/2 results in 3 rather than 2.5 because division is performed before addition. If an expression contains several operators that have the same precedence, the operations are performed left

to right. For example, in the expression 1+4/2*6/3, the division and multiplication operations are performed left to right before the addition operation is performed. The first operation is 4/2, resulting in 2; the second multiplies 2 by 6, resulting in 12; the third operation divides 12 by 3, resulting in 4; the last operation adds 1 to 4, resulting in 5.

By using parentheses, you can deviate from the standard order of evaluation. R&R performs operations within parentheses first. For example, the result of 1+4/2 is 3, but the result of (1+4)/2 is 2.5 because R&R performs the operation in parentheses first.

If parentheses are nested or embedded, as in the expression 3*((1+4)/2), the operation in the most deeply embedded parentheses is performed first. In this case, the first operation is 1+4, the second is 5/2, and the third is 3*2.5.

Date Operators

R&R provides the date operators shown in Figure 8.4.

These date operators are supplemented by a number of predefined date functions such as ADDDAYS( ), ADDMONS( ), and ADDWKS( ) that are explained in Chapter 10, "Using Functions."

Operator	*Description*
+	Addition of days to a date. For example, ORDERDATE+30
–	Subtraction of days from a date or subtraction of one date from another. For example, DUEDATE–CURDATE
(*and*)	Open/close parenthesis for grouping

Figure 8.4 Date Operators

Character Operators

R&R provides the character operators shown in Figure 8.5.

Operator	*Description*
+	Concatenation, the joining of two character strings into one. For example, FIRSTNAME+LASTNAME
–	Concatenation after removing trailing spaces from the string before the operator. For example, FIRSTNAME–" "+LASTNAME
(*and*)	Open/close parenthesis for grouping

Figure 8.5 Character (String) Operators

Note that both + and – "add" strings together. The only difference between the two operators is that – removes the trailing spaces from the string that precedes it. For example, if FIRSTNAME is John and LASTNAME is Jones (assuming both fields are 10 characters):

FIRSTNAME+LASTNAME	=	John Jones
FIRSTNAME–LASTNAME	=	JohnJones
FIRSTNAME–" "+LASTNAME	=	John Jones

The last expression adds a string consisting of one space to the first name field, after the – operator removes trailing spaces.

Relational Operators

R&R provides the relational operators shown in Figure 8.6. Use these operators to compare numeric, date, or character data.

Operator	*Description*
=	Equal to
# or <>	Not equal to
<	Less than
<=	Less than or equal to
>	Greater than
>=	Greater than or equal to
$	Substring comparison; returns True if term to the left of the $ is identical to or contained in term to the right of the $ or False if not
(*and*)	Open/close parenthesis for grouping

Figure 8.6 Relational Operators

Logical Operators

The logical operators shown in Figure 8.7 produce a true or false value.

Operator	*Description*
.NOT., NOT	Logical not
.AND., AND	Logical and
.OR., OR	Logical or
(*and*)	Open/close parenthesis for grouping

Figure 8.7 Logical Operators

R&R evaluates the .NOT. logical operator before it evaluates .AND. and .OR., which have equal precedence. In expressions containing both .AND. and .OR., R&R evaluates the operators from left to right, according to their order in the expression.

For example, in the following expression the .OR. operator is evaluated before the .AND. operator:

DEPT = "Sales" .OR. DEPT = "Mkting" .AND. JOBCODE = "Mgr"

In evaluating this expression, R&R first checks to see whether employees meet either of the first two conditions. If an employee meets either condition, R&R then applies the last logical condition to determine whether the employee's job code is "Mgr". As a result, the expression is true for all employees who are managers in either the Sales or Marketing department.

Mixed Operator Types

When you insert more than one type of operator in a calculation, R&R performs operations in the following order:

1. Arithmetic, Date, and Character operations
2. Relational operations
3. Logical operations

For example, R&R evaluates the expression 3*4<8.OR.4>3+2 as false, because the arithmetic operators (* and +) are performed first, the relational operators (< and >) are performed second, and the logical comparison (.OR.) is performed last. The expression is evaluated as if it were ((12<8).OR.(4>5)).

In the expression BALANCE>100.AND.DAYS>30, R&R evaluates the relational operators (>) first. Both conditions must be true for the expression to be true; that is, the balance must be greater than 100 and the number of days must be greater than 30.

In the expression CTOD("7/1/95")<DATE+30, R&R adds DATE+30 first and then compares the result to the date returned by the CTOD( ) (character to date) function. Therefore, the expression is true only when DATE+30 results in a date that is later than 7/1/95.

Functions

Functions perform special operations such as converting data from one data type to another, calculating the elapsed time between dates, changing number formats, and conditionally returning values.

A single function can serve as a calculated field expression, or several functions can be included within a single expression. For example, to include the system date in a report, you can create a calculated field called SYSDATE with the expression DATE( ). When included in a report, this field will provide the system date when the report is generated.

An expression can also consist of a number of functions that can be nested or embedded within each other. For example, the expression CMONTH(DATE( ))+", "+STR(YEAR(DATE( )),4) contains four

functions. The DATE( ) function returns the system date in the format 11/04/94. The CMONTH( ) function returns the name of the month of the date supplied by DATE( ). The YEAR( ) function returns the four-digit year of the date supplied by DATE( ). The STR( ) function converts this numeric year value to a four-character string, so that it can be concatenated with the name of the month string and the character constant ", ". On 11/04/95, the value returned by the expression is November, 1995.

R&R provides over 100 predefined functions, as well as the ability to create and save user-defined functions. For information on creating your own functions, see Chapter 10, "Using Functions."

Wildcard Characters

You can use wildcards for pattern matching in calculated field expressions that compare character strings and dates. For example, the expression NAME="HEN*" will match any string starting with HEN, including names such as Henrietta, Henry, and Henderson. The expression DATE={12/*/95} will match any date in December of 1995.

Figure 8.8 lists the wildcard characters used to define patterns.

Character	*Meaning*
?	In a character expression, matches any single character in the same position in the field.
*	In a character expression, matches any group of characters (including no characters). In a date expression, matches any value in that part of the date (e.g. 01/*/95).
@	In a date expression, matches any value that corresponds to that part of the system date (e.g. @/@/95).

Figure 8.8 Wildcards in Expressions

You can use wildcards in calculated fields that use equality or inequality operators (for example, NAME="HEN*") or functions (for example, CTOD("@/*/95")). In relational expressions, use wildcard characters for pattern matching in the right side of the expression; characters in the left side of the expression are treated as literals.

For example, STATE="C?" finds a match when STATE contains CA, because R&R treats the question mark on the right side of the relational operator as a wildcard. However, "C?"=STATE will evaluate to false and no match will be found, because R&R treats the question mark on the left side of the relational operator as a literal.

To use a wildcard character as a literal in the right side of the expression, precede it with a backslash. For example, use FIELD="*" to test for records in which FIELD contains only an asterisk.

In calculated field expressions that use the functions CASE( ) and INLIST( ), you can use wildcard characters with the items that you want to test against; wildcard characters used as test values will be treated as literals. In the following examples, the function INLIST( ) uses the first argument as the test value, and compares it with each value in the list. When a match is found, INLIST( ) returns the number corresponding to the position of the matching value in the list. When no match is found, INLIST( ) returns 0.

INLIST("AB","XX","A?") returns 2
INLIST("A?","XX","AB") returns 0

You can also compare fields that contain wildcard characters as data values. For example, if FIELD1 contains the string “AB” and FIELD2 contains the string “A?”, R&R will find the match when you use the calculated field expression FIELD1=FIELD2.

Error Conditions in Evaluating Expressions

When you print or preview a report, if R&R encounters an error in evaluating a calculated field expression, it displays asterisks as the error value in the field and in any fields that total the calculated field. For example, if R&R tries to evaluate a date expression that adds a specified number of days to an empty date field, it will display **/**/** as the result, since it cannot produce a proper date.

To take another example, if R&R tries to evaluate a numeric expression that requires it to divide by zero, it will display asterisks as the result, since division by zero is an undefined operation. If this calculated field were totaled, the total value would also display as asterisks.

The standard error value sorts as the last value in an ascending sort or the first value in a descending sort.

Customizing Error Values

You can create your own error values for specific expressions by using the IIF and ERROR functions. If you know that an expression may result in an error, you can specify the error value that will display. For example, if the expression for an average price field divides total invoice amount by quantity ordered (AVERAGE=INVAMT/QUANTITY), the expression will result in an error when the quantity ordered is 0. To print 0 instead of the standard error value of asterisks, you can use the following expression.

```
IIF(ERROR(AVERAGE),0,AVERAGE)
```

If evaluation of the AVERAGE field results in an error, then ERROR(AVERAGE) will be true, leading the IIF function to return 0. If evaluation of the AVERAGE field does not result in an error, ERROR(AVERAGE) will be false, leading the IIF function to return the result of the AVERAGE expression.

To print a string instead of a number as an error value, you might use an expression like:

```
IIF(ERROR(AVERAGE),"N/A",STR(AVERAGE))
```

Field Width Errors

Asterisks in a calculated numeric field or total field may also indicate the field width is not large enough to accommodate the result. R&R estimates the number of digits required for each calculated and total field based on the fields and functions used in the expression. You may need to adjust this width using Format ⇒ Field, especially if you have edited the expression since the initial field width estimate was established. See Chapter 5, "Working with Fields," for information on changing field width.