How to Use Adhoc Parameters in Actuate Reports

By Chris Geiss
chris_geiss@yahoo.com
http://www.chrisgeiss.com

How to Use Adhoc Parameters in Actuate Reports

By Chris Geiss
Revised 3/31/2002

Table of Contents

About the Author

Chris Geiss is an independent consultant and software developer with over 16 years of software development and IT industry experience. He has been working with the Actuate reporting products for 4 years and specializes in consulting, development, training and mentoring services for the Actuate e.Reporting Suite. Chris can be contacted at chris_geiss@yahoo.com. See his web site at http://www.chrisgeiss.com.

Introduction

Adhoc report parameters are one of the many useful and powerful features of Actuate e.Reporting Suite 5.

An Adhoc parameter is an optional report parameter in which you can provide expressions to specify how the report should filter the data that appears in your report document. Each Adhoc parameter corresponds to a field (column) in the database. If you do not provide a value for an adhoc parameter, then Actuate will not filter the data in your generated document based on the corresponding database field. If you provide a single value in an adhoc parameter, then the report will only return data where the corresponding field in the database matches the value that you specified. However, if you need a broader set of data, you can also use expressions in adhoc parameters, in which you can specify a range or set of values that you are interested in. Thus adhoc parameters give you very flexible control over the data that is output to your report document. Please note that while adhoc parameters are usually optional, the report developer can specify that certain adhoc parameters are required.

Unfortunately, users occasionally run into problems when using reports that have adhoc parameters because they are not aware of all of the rules regarding how to use adhoc parameters. The rules for numeric and date parameters are straightforward, but the rules for string type parameters can be a little tricky and confusing to users because Actuate tries to make the user's life easier by providing automatic wildcard handling (pattern matching) for string type parameters. As a result, the user sometimes does not get the data that he or she expected.

This document summarizes the rules for entering adhoc parameter values and provides examples that illustrate these concepts. Please note that the adhoc parameter behavior discussed in this document is based on my work with Actuate Version 5.0 SP1.

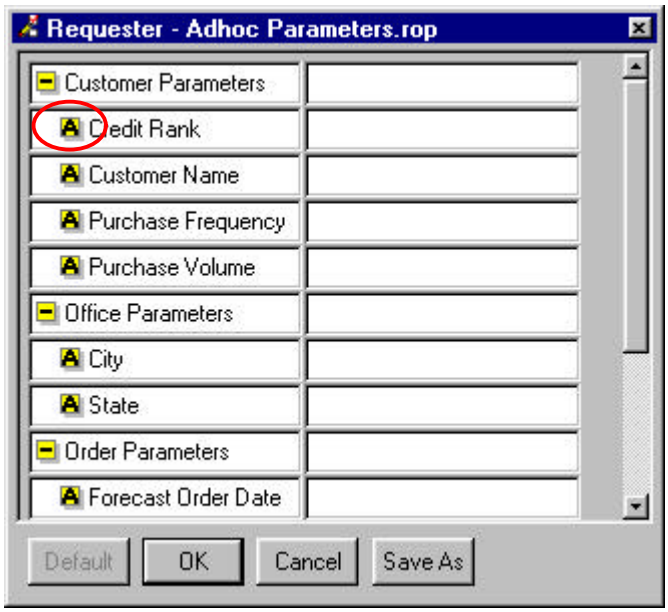
The Mailing List and How to Report Errors and Omissions

If you find any errors or omissions in this document, or have any suggestions, please email them to chris_geiss@yahoo.com so I can update this document accordingly. Also, if you email your contact information to me, I will include you on the mailing list for future updates to this document.

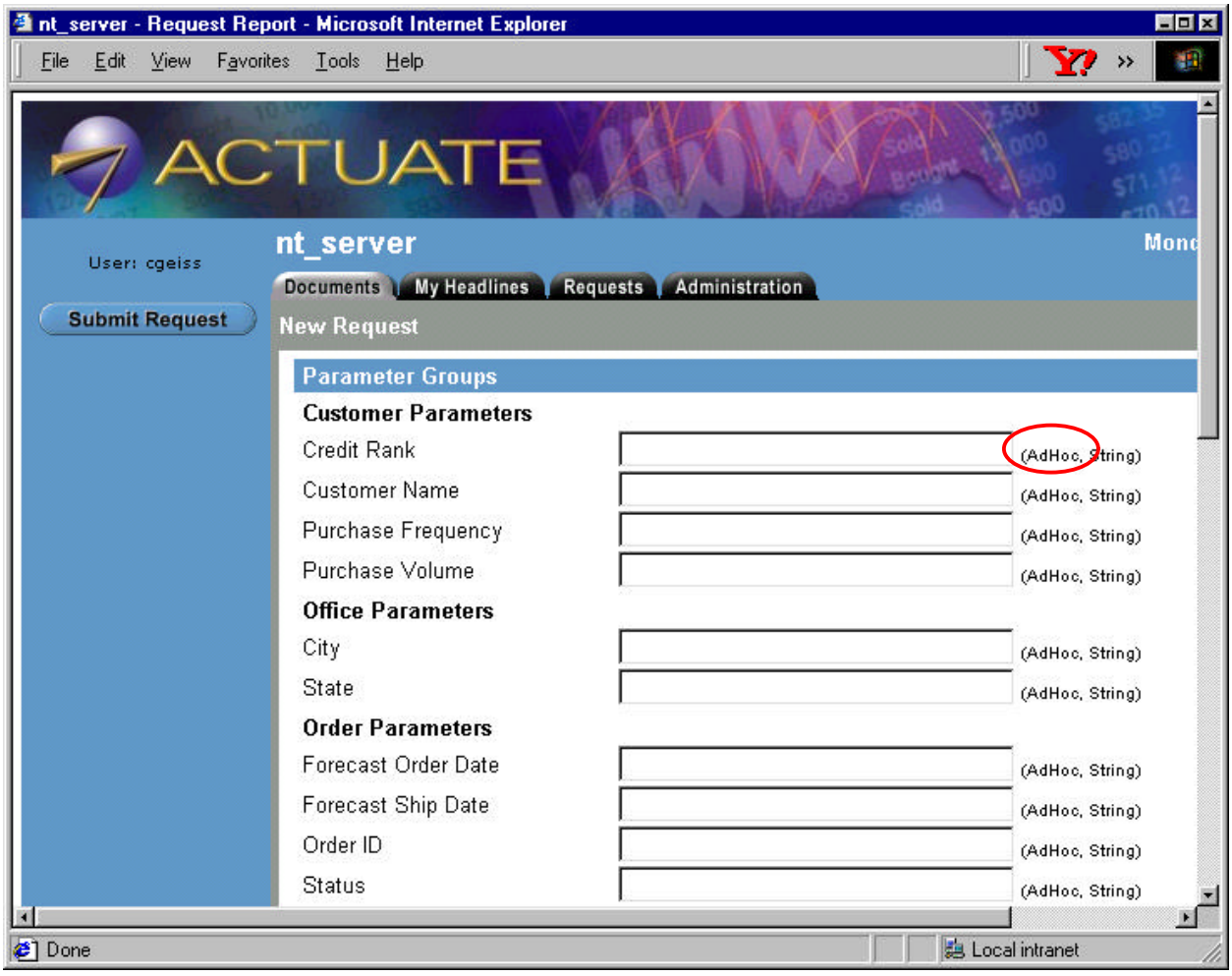
How do You Know that a Parameter is an Adhoc Parameter?

Actuate uses a variety of mechanisms to alert the user that a parameter is an adhoc parameter.

When you are using a client side tool (such as End User Desktop, Administrator Desktop, e.Report Designer, or e.Report Designer Professional) you will see a special "A" icon in black on a yellow background next to each adhoc parameter when you run the report. The screenshot below shows examples of this icon. In this case, all of the parameters are adhoc parameters.



If you are running reports via a web browser, then Actuate will use the text "AdHoc" to mark the report parameters that are adhocs. This is illustrated in the screenshot below.



Please note, that the screenshots above are from the default screen designs that are provided by Actuate. The Actuate environment you use to run reports may have been customized, and the screens you see may not look like the ones illustrated above. If this is the case in your environment, please contact your system administrator or software development department to find out how you can tell if a report parameter is an adhoc parameter.

How do You Know the Data Type Associated with an Adhoc Parameter?

When you enter values and expressions for adhoc parameters, it is important to know if you are entering an adhoc expression for a string, date or number type adhoc parameter. The reason this is important, is the parameter type determines the rules for entering values for the parameter. Be sure to refer to your user documentation, or contact your system administrator or software development department to find out the data types for the adhoc parameters in each of the reports that you use.

Rules for Entering Adhoc Parameter Values and Expressions

This section summarizes the rules for entering adhoc parameter values and expressions.

String Type Parameters

1) If you enter a string value that is not enclosed in single quote marks, Actuate assumes you want a wildcard character (%) at the end of the string so that the query matches any strings that start with the text you entered. This helps deal with database columns that are defined as char instead of varchar and are thus right padded with spaces. The user does not have to enter the trailing spaces to get a match.
2) You can explicitly include wildcard characters (%) within the text you provide and Actuate will use the wildcards you provide and will not automatically place one at the end of the text you entered.
3) If you want to search for an exact text value, and do not want wildcard handling, then enclose the text you are searching for in single quote marks, for example ‘ABC’. If you specify a value that is not contained within single quotes, then Actuate will make it into a wildcard, unless the text you enter starts with =.
4) The asterisk is not treated as a wild card, but instead as a literal character for which you are searching. For example, A*B only matches the text “A*B”, and not “ACB” or “ACDB”.
5) Double quotation marks are treated as literal characters, and do not have any special meaning; thus entering a double quote mark searches for the character “.
6) If you want to search for more than one item, then you can separate items with commas. This works for wildcard and literal searches. For example: A,B,C or A%,B%,C% are wildcard searches and search for any string that starts with A or B or C. ‘A’,’B’,’C’ does a literal search for the exact strings A or B or C. ‘A’, B, C% is also acceptable syntax.
7) If you want to search for a range of items, then you can separate them with a dash (-). For example, ABC-DEF matches all values that are alphabetically between the strings “ABC” and “DEF”, inclusive. When you use the dash syntax, wildcard handling is disabled for the strings you specify.
8) Backslash acts as an escape character that removes the special meaning for some characters. For example A\-P searches for all items starting with “A-P” instead of looking for items between “A” and “P”.
9) The escape character does not work with %. Therefore, you cannot search for strings containing the % character by specifying \%.
10) The = character can be used to search for strings containing literal % characters. For example, =A%B searches for strings that equal the text “A%B”, and does not treat the % as a wildcard. Basically, if the text you enter starts with =, then Actuate treats it as a literal string to search for and not as a wildcard expression. However, unless you place any text after the = between single quotes, or you place a % somewhere in the text, Actuate will place a % at the end of the string.
11) The comma (,) takes precedence over the single quote mark (‘). A comma basically becomes “OR” unless it is escaped with the \ character.
12) The various syntaxes can be combined. For example, a list of ranges can be specified as: A-C, G-I, X-Z.
13) The inequality operators (<, <=, >, >=) work as expected so long as you place the string values you are looking for within single quotes. For example, <’ABC’,>’XYZ’.

Numeric and Date Type Parameters

1) These adhoc parameters are straightforward and behave the way you would expect since there is no default wildcard behavior as there is with strings.
2) These parameters support the comma (,) to specify a list of values or ranges.
3) These parameters support the dash (-) to specify a range of values.
4) These parameters support the equality and inequality operators.
5) With numeric parameters, negative signs can be confused as range dashes, so use the inequality operators to avoid problems. For example, instead of -300--200, use >=-300,<=-200.
6) The only date formats permitted for date type adhoc parameters are mm/dd/yy and mm/dd/yyyy. However, note that 3/1/02 can be used instead of 03/01/02. Please note that these comments apply to US locales only, I have not tested this with non-US installations or localized versions of Actuate products.

Tables of Adhoc Parameter Value and Expression Examples

The sections that follow provide tables of examples for each of the major adhoc parameter types: string, numeric and date. Each table provides examples of parameter values and expressions, the data that will be returned by each example expression, and any applicable comments.

In the following tables, the single quote and double quote characters are used in the example parameter values and expressions. To avoid confusion, when a reference is made to a specific character string that is matched by a parameter value or expression, the string is delimited with the { and } bracket characters. The { and } characters should not be interpreted as being a part of the string that is matched.

Examples for a String Type Adhoc Parameter

Parameter Expression	Data Returned in the Report Document	Comments
A	Wildcard. Matches all values that start with {A}.	A single string value is automatically treated like a wildcard expression.
A%	Same as above.	You can provide the % explicitly.
A\%	Same as above.	The backslash character does not escape the meaning of the %.
'A%'	Same as above.	Using % forces wildcard handling.
'A\%'	Same as above.	Same as above.
%A	Wildcard. Matches all values that end with {A}.	You can change the default location of the % in a wildcard expression.
A%B	Wildcard. Matches all values that start with {A} and end with {B}.	Same as above.
%A%B%	Wildcard. Matches all values that have an {A} and {B} within them.	You can include more than one % wildcard character.
A*	Wildcard. Matches all values that start with {A*}.	The asterisk does not act like a wildcard character. Only use '*' if you are searching for the asterisk character.
'A'	Equality. Matches all values that equal {A}.	Placing parameter values in single quotes forces them to be treated literally instead of as wildcard expressions.
'A*'	Equality. Matches all values that equal {A*}.	Only use '*' if you are searching for the asterisk character.
"A"	Matches all values that start with {"A"}.	The double quote character does not have any special meaning. Double quote marks are treated as characters you are trying to match.
'"A"'	Equality. Matches all values that equal {"A"}.	Enclosing the parameter value above in single quotes takes away the wildcard behavior.
A,B,C	List of wildcards. Matches any value that starts with {A} or {B} or {C}.	
A\,B,C	Matches all values that start with {A,B} or {C}.	The backslash character escapes the meaning of the comma.
'A,B','C'	Results in error.	Do not use. Comma has precedence over the single quote. Use \ to escape the meaning of the comma.
'A\,B','C'	Matches all values that equal {A,B} or {C}.	This provides the result desired above. Use \ to escape the meaning of the comma.
%A,%B,%C	Matches all values that end in {A} or {B} or {C}.	
'A','B','C'	Matches all values that equal {A} or {B} or {C}.	List of exact values to match.
"A","B","C"	Matches all values starting with {"A"} or {"B"} or {"C"}	The comparison includes the quote marks.
A-P	Match all values that are alphabetically between {A} and {P}.	Dash is used to specify a range of values and removes the default wildcard behavior.
A\-P	Matches all values that start with {A-P}.	The backslash escapes the range meaning of the dash.
'A\-P'	Matches all values that equal {A-P}.	The single quotes take away the wildcard behavior.
'A-P'	SQL syntax error.	Do not use. Must use a backslash to escape the meaning of the dash.

Examples for a String Type Adhoc Parameter (cont.)

Parameter Expression	Data Returned in the Report Document	Comments
“A-P”	Match all values that are alphabetically between {“A} and {P”}.	
‘A’-‘P’	Matches the same values as: A-P	The single quotes have no effect.
“A”-“P”	Match all values that are alphabetically between {“A”} and {“P”}.	
=A	Matches all values that equal {A%}. Does not perform a wildcard match.	Using the equal sign avoids the conversion to a wildcard, but includes a %.
=A%	Same as above.	
=%A	Matches all values that equal {%A}. Does not perform a like.	
=%A%B%C%	Matches all values that equal {%A%B%C%}.	
=’A’	Same as: ‘A’	Using the equal sign here does not make a difference.
=”A”	Matches all values that equal {“A”%}.	
=A,B,C	Matches all values that equal {A%} or that start with {B} or {C}.	Using the equal sign results in the % character being automatically inserted, but the % does not act like a wildcard.
‘A’,B, C	Matches all values that equal {A} or that start with {B} or {C}.	
=A,=B,=C	Matches all values that equal {A%} or {B%} or {C%}.	Use = if you want to search for strings containing %.
=’A’,=’B’,=’C’	Matches all values that equal {A} or {B} or {C}.	
=A-P	Same as: A-P	Use A-P instead.
ABC-DEF	Matches all values that are alphabetically between {ABC} and {DEF}.	The range is inclusive.
‘A’,B,C-F	Matches all values that are equal to {A} or start with {B} or are alphabetically between {C} and {F}.	The list, equality and range features can be combined.
A-C,X-Z	Matches all values that are alphabetically between {A} and {C} or {X} and {Z}.	
3/1/02	Matches all values that start with {3/1/02}.	Date values used with string parameters are handled differently than when used with date parameters.
‘3/1/02’	Matches all values that equal {3/1/02}.	Same as above.
<ABC	Matches all values alphabetically less than {ABC%}.	The added % may cause matching problems.
<’ABC’	Matches all values alphabetically less than {ABC}.	Placing the text in single quotes solves the added % problem.
<=’ABC’	Matches all values alphabetically less than or equal to {ABC}.	
>’ABC’	Matches all values alphabetically greater than {ABC}.	
>=’ABC’	Matches all values alphabetically greater than or equal to {ABC}.	
<’ABC’,>’XYZ’	Matches all values alphabetically less than {ABC} or greater than {XYZ}.	
<=’ABC’,DEF-XYZ	Matches all values alphabetically less than or equal to {ABC} or alphabetically between {DEF} and {XYZ}.	

Examples for a Numeric Type Adhoc Parameter

Parameter Expression	Data Returned in the Report Document	Comments
10	Matches all values equal to 10.	Single value.
=10	Same as above.	Equal sign is not required.
10,20,30	Matches all values equal to 10, 20 and 30.	List of values.
10-30	Matches all values between 10 and 30 (inclusive).	Range of values, inclusive.
10-30,100-120	Matches all values between 10 and 30 and all values between 100 and 120.	List of value ranges.
<10	Matches all values less than 10.	Less than.
<=10	Matches all values less than or equal to 10.	Less than or equal to.
>10	Matches all values greater than 10.	Greater than.
>=10	Matches all values greater than or equal to 10.	Greater than or equal to.
<10,>=100	Matches all values less than 10 and all values greater than or equal to 100.	List of comparisons.
<10,100-500	Matches all values less than 10 and all values between 100 and 500.	Comparison combined with a range.
<10,100-500,1000	Matches all values less than 10 and all values between 100 and 500 and all values equal to 1000.	Compound list.
-300,-200	Matches the values -300 and -200.	List using negative values.
\-300,\-200	Same as above.	\ does not have any effect here.
\-300-\-200	n/a	Does not work.
>=-300,<=-200	Matches all values greater than or equal to -300 and all values less than or equal to -200.	Negative values can be problematic when using the dash notation for ranges. Use inequality expressions instead.
3/1/02	Matches all values equal to 3.	Everything after first / is ignored.

Examples for a Date Type Adhoc Parameter

Parameter Expression	Data Returned in the Report Document	Comments
3/1/02	Matches the date 3/1/02.	Single value; do not have to include leading zeroes.
03/01/02	Same as above.	Leading zeroes are OK.
3/1/02,3/10/02	Matches the dates 3/1/02 and 3/10/02.	List of dates.
3/1/02-3/10/02	Matches dates between 3/1/02 and 3/10/02 (inclusive).	Range of dates, inclusive.
3/1/02,3/10/02-3/20/02	Matches the date 3/1/02 and dates between 3/10/02 and 3/20/02.	Single value and a list.
=3/1/02	Matches the date 3/1/02.	Equal sign is unnecessary.
<3/1/02	Matches dates less than 3/1/02.	Less than.
<=3/1/02	Matches dates less than or equal to 3/1/02.	Less than or equal to.
>3/10/02	Matches dates greater than 3/10/02.	Greater than.
>=3/10/02	Matches dates greater than or equal to 3/10/02.	Greater than or equal to.
<3/1/02,>=3/30/02,3/20/02-3/24/02	Matches dates less than 3/1/02, greater than or equal to 3/30/02 and dates between 3/20/02 and 3/24/02 (inclusive).	Ranges can be combined with comparison expressions.
3-1-02	n/a	Invalid date format.
2002-3-1	n/a	Invalid date format.