

Table 1. Typecasting Functions

| Function | Description |
|---|---|
| <code><text>.TYPECAST_LIST_T(<separator>)</code> | <p>Treats the text in an HTTP request or response body as a list whose elements are delimited by the character in the <separator> argument. Index values in the list that is created start with zero (0).</p> <p>Text mode settings have no effect on the separator. For example, even if you set the text mode to IGNORECASE, and the separator is the letter “p,” an uppercase “P” is not treated as a separator.</p> <p>The following example creates a Rewrite action that constructs a list from an HTTP request body and extracts the fourth item in the list:</p> <pre>add rewrite action myreplace action REPLACE 'http.req.body(100) ' 'http.req.body(100).typecast_list_t('?').get(4) ' set rewrite policy myreplace_policy - action myreplace_action</pre> <p>This policy returns the string “fourth item” from the following request:</p> <pre>GET?first item?second item?third item?fourth item?</pre> <p>The following example extracts the fourth-from-last item from the list.</p> <pre>add rewrite action myreplace_action1 REPLACE 'http.req.body(100) ' 'http.req.body(100).typecast_list_t('?').get_reverse(4) ' set rewrite policy myreplace_policy1 - action myreplace_action1</pre> <p>This policy returns the string “first item” from the following request:</p> <pre>GET?first item?second item?third item?fourth item.</pre> |
| <code><text>.TYPECAST_NVLIST_T(<separator>, <delimiter>)</code> | <p>Treats the text as a name-value list. The <separator> argument identifies the character and separates the name and the value. The <delimiter> argument identifies</p> |

Table 1. Typecasting Functions

| Function | Description |
|--|---|
| <p>or</p> <pre>text.TYPESCAST_ NVLIST_T(<separator>, <delimiter>, <quote>)</pre> | <p>the character that separates each name-value pair. The <quote> character is required when typecasting text into a name-value list that supports quoted strings. Any delimiters that appear within the quoted string are ignored.</p> <p>The text mode has no effect on the delimiters. For example, if the current text mode is IGNORECASE and you specify “p” as the delimiter, an uppercase “P” is not treated as a delimiter.</p> <p>For example, the following policy counts the number of name-value pairs and inserts the result in a header named name-value-count:</p> <pre>add rewrite action mycount_action insert_http_header name-value-count 'http.req.header("Cookie").typecast_nvlist_ _t('=';';').count'</pre> <pre>set rewrite policy mycount_policy -action mycount_action</pre> <p>This policy can extract a count of arguments in Cookie headers and insert the count in a name-value-count header:</p> <pre>Cookie: name=name1; rank=rank1</pre> |
| <pre><text>.TYPESCAST_TIME_T</pre> | <p>Treats the designated text as a date string. The following formats are supported:</p> <ul style="list-style-type: none"> • RFC822: Sun, 06 Nov 1994 08:49:37 GMT • RFC850: Sunday, 06-Nov-94 08:49:37 GMT • ASCII TIME: Sun Nov 6 08:49:37 1994 • HTTP Set-Cookie Expiry date: Sun, 06-Nov-1994 08:49:37 GMT <p>For example, the following policy converts the string to a time value and then extracts the day. This policy matches all requests that have a day value lesser than or equal to 10.</p> <pre>Add rewrite policy mytime_policy "http.req.body(100) .typecast_time_t.day.le(10)" mytime_action bind rewrite global mytime_policy 100</pre> |

Table 1. Typecasting Functions

| Function | Description |
|---|---|
| <code><numeric string>.TYPECAST_IP_ADDRESS_T</code> | <p>Treats a numeric string as an IP address.</p> <p>For example, the following policy matches HTTP requests that contains Cookie headers with a value of: 12.34.56.78\r\n.</p> <pre>set rewrite policy ip_check_policy -rule 'http.req.cookie .value("ip").typecast_ip_address_t.eq(12.34.56.78)'</pre> <pre>bind rewrite global ip_check_policy 200 -type req_default</pre> |
| <code><numeric string>.TYPECAST_IPV6_ADDRESS_T</code> | <p>Treats a string as an IPv6 address in the following format:</p> <p>0000:0000:CD00:0000:0000:00AB:0000:CDEF</p> |
| <code><text>.TYPECAST_HTTP_URL_T</code> | <p>Treats the designated text as the URL in the first line of an HTTP request header. The supported format is [<code><protocol>://<hostname><path>?<query></code>], and the text mode is set to URLENCODED by default.</p> <p>For example, the following policy replaces a URL-encoded part of a string in an HTTP header named Test.</p> <pre>add rewrite action replace_header_string replace "http.req.header("Test").typecast_http_url_t.path .before_str("123").after_str("ABC") "\"string\""</pre> <pre>add rewrite policy rewrite_test_header_policy true replace_header_string bind rewrite global rewrite_test_header_policy 1 END -type res_override</pre> <p>Consider the following header:</p> <pre>Test: ABC%12123\r\n</pre> <p>This policy would replace the preceding header with the value <code>ABC%string123\r\n</code>.</p> |
| <code><text>.TYPECAST_HTTP_HOSTNAME_T</code> | <p>Provides operations for parsing an HTTP host name as it appears in HTTP data. The format for a host name is <code>abc.def.com:8080</code>.</p> |

Table 1. Typecasting Functions

| Function | Description |
|---|---|
| <code><text>.TYPECAST_HTTP_METHOD_T</code> | <p>Converts text to an HTTP method.</p> <p>For example, the following policy matches any HTTP request that contains a Host header with a value equal to POST:</p> <pre>Add rewrite policy method_policy "http.req.header(\"Host\") .typecast http method t.eq(POST)" act1</pre> |
| <code><text>.TYPECAST_DNS_DOMAIN_T</code> | <p>Enables the designated text to be parsed like a DNS domain name in the format ab.def.com.</p> |
| <code><text>.TYPECAST_HTTP_HEADER_T ("<name>")</code> | <p>Converts the designated text to a multi-line HTTP header that you specify in a <name> argument.</p> <p>For example, the following expression converts "MyHeader" to "InHeader":</p> <pre>http.req.header("MyHeader").typecast_http_header_t("InHeader")</pre> <p>Typically, text operations that you specify in this type of expression apply to only the last line of this header, with some exceptions. For example, the CONTAINS operation operates on values in all the lines in instances of this header type.</p> |
| <code><text>.TYPECAST_COOKIE_T</code> | <p>Treats the designated text as an HTTP cookie as it appears in a Set-Cookie or Set-Cookie2 header. You can apply name-value list operations as well as text operations to the designated text. For example, you can designate equals (=) as the name-value delimiter and the semicolon (;) as the list element delimiter.</p> <p>If you apply name-value list operations, the list is parsed as if IGNORE_EMPTY_ELEMENTS were in effect.</p> <p>Each cookie begins with a <code>cookie-name=cookie-value</code> pair, optionally followed by attribute-value pairs that are separated by a semicolon, as follows:</p> <pre>cookie1=value1;version=n.n;value;domain=value;path=value</pre> <p>If the same attribute appears more than once in a cookie, the value for the first instance of the attribute is returned.</p> |
| <code><number>.TYPECAST_DOUBLE_AT</code> | <p>Transforms the number to a value of data type double.</p> |
| <code><number>.TYPECAST_IP_ADDRESS_AT</code> | <p>Converts the number to an IP address.</p> |
| <code><number>.TYPECAST_TIME_AT</code> | <p>Converts the number to time format.</p> |

Table 1. Typecasting Functions

| Function | Description |
|--|---|
| <code><number>.TYPECAST_TIME_AT.BETWEEN(<time1>, <time2>)</code> | <p>Returns a Boolean value (TRUE or FALSE) that indicates whether the time value designated by <number> is between the lower and upper time value arguments <time1> and <time2>.</p> <p>The following are prerequisites for this function:</p> <ul style="list-style-type: none"> • Both the lower and upper time arguments must be fully specified. For example, GMT 1995 Jan is fully specified. But GMT Jan, GMT 1995 20 and GMT Jan Mon_2 are not fully specified. • Both arguments must be either GMT or Local. • The day of the week must not be present in either argument. However, the day of the month can be specified as the first, second, third, or fourth weekday of the month (example Wed_3 is the third Wednesday of the month). • The upper time argument, <time2>, must be bigger than the lower time argument, <time1>. <p>The following examples assume that the current time value is GMT 2005 May 1 10h 15m 30s and that the day is the first Sunday of the month of May in 2005. The result of the evaluation is given after each example.</p> <pre>BETWEEN(GMT 2004, GMT 2006): TRUE BETWEEN(GMT 2004 Jan, GMT 2006 Nov): TRUE BETWEEN(GMT 2004 Jan, GMT 2006): TRUE BETWEEN(GMT 2005 May Sun_1, GMT 2005 May Sun 3): TRUE BETWEEN(GMT 2005 May 1, GMT May 2005 1): TRUE BETWEEN(LOCAL 2005 May 1, LOCAL May 2005 1): The result depends on the NetScaler system's timezone.</pre> <p>Parameters:</p> <p><time1> - Lower time value</p> <p><time2> - Upper time value</p> |
| <code><number>.TYPECAST_TIME_AT.DAY</code> | <p>Extracts the day of the month from the current system time and returns the value as a number that corresponds to the day of the month. The returned value ranges from 1 to 31.</p> |
| <code><number>.TYPECAST_TIME_AT.EQ(<t>)</code> | <p>Returns a Boolean value (TRUE or FALSE) that indicates whether the time value designated by <number> is equal to the time value argument <t>.</p> |

Table 1. Typecasting Functions

| Function | Description |
|--|---|
| | <p>The following examples assume that the current time value is GMT 2005 May 1 10h 15m 30s and that the day is the 1st Sunday of the month of May in 2005. The result of the evaluation is given after each example.</p> <pre>EQ(GMT 2005): TRUE EQ(GMT 2005 Dec): FALSE EQ(Local 2005 May): TRUE or FALSE, depending on the time zone. EQ(GMT 10h): TRUE EQ(GMT 10h 30s): TRUE EQ(GMT May 10h): TRUE EQ(GMT Sun): TRUE EQ(GMT May Sun_1): TRUE</pre> <p>Parameters:</p> <p><t> - Time</p> |
| <pre><number>.TYPECAST_TIME_AT.GE(<t>)</pre> | <p>Returns a Boolean value (TRUE or FALSE) that indicates whether the time value designated by <number> is greater than or equal to the time value argument <t>.</p> <p>The following examples assume that the current time value is GMT 2005 May 1 10h 15m 30s and that the day is the 1st Sunday of the month of May in 2005. The result of the evaluation is given after each example.</p> <pre>GE(GMT 2004): TRUE GE(GMT 2005 Jan): TRUE GE(Local 2005 May): TRUE or FALSE, depending on the time zone. GE(GMT 8h): TRUE GE(GMT 30m): FALSE GE(GMT May 10h): TRUE GE(GMT May 10h 0m): TRUE GE(GMT Sun): TRUE GE(GMT May Sun_1): TRUE</pre> <p>Parameters:</p> <p><t> - Time</p> |
| <pre><number>.TYPECAST_TIME_AT.GT(<t>)</pre> | <p>Returns a Boolean value (TRUE or FALSE) that indicates whether the time value designated by <number> is greater than the time value argument <t>.</p> <p>The following examples assume that the current time value is GMT 2005 May 1 10h 15m 30s and that the day</p> |

Table 1. Typecasting Functions

| Function | Description |
|--|---|
| | <p>is the 1st Sunday of the month of May in 2005. The result of the evaluation is given after each example.</p> <pre>GT(GMT 2004): TRUE GT(GMT 2005 Jan): TRUE GT(Local 2005 May): TRUE or FALSE, depending on the time zone. GT(GMT 8h): TRUE GT(GMT 30m): FALSE GT(GMT May 10h): FALSE GT(GMT May 10h 0m): TRUE GT(GMT Sun): FALSE GT(GMT May Sun_1): FALSE</pre> <p>Parameters:</p> <p><t> - Time</p> |
| <pre><number>.TYPECAST_TIME_AT.HOURS</pre> | <p>Extracts the hour from the current system time and returns the corresponding value as an integer that can range from 0 to 23.</p> |
| <pre><number>.TYPECAST_TIME_AT.LE(<t>)</pre> | <p>Returns a Boolean value (TRUE or FALSE) that indicates whether the time value designated by <number> is lesser than or equal to the time value argument <t>.</p> <p>The following examples assume that the current time value is GMT 2005 May 1 10h 15m 30s and that the day is the 1st Sunday of the month of May in 2005. The result of the evaluation is given after each example.</p> <pre>LE(GMT 2006): TRUE LE(GMT 2005 Dec): TRUE LE(Local 2005 May): TRUE or FALSE, depending on the time zone. LE(GMT 8h): FALSE LE(GMT 30m): TRUE LE(GMT May 10h): TRUE LE(GMT Jun 11h): TRUE LE(GMT Wed): TRUE LE(GMT May Sun_1): TRUE</pre> <p>Parameters:</p> <p><t> - Time</p> |
| <pre><number>.TYPECAST_TIME_AT.LT(<t>)</pre> | <p>Returns a Boolean value (TRUE or FALSE) that indicates whether the time value designated by <number> is lesser than the time value argument <t>.</p> <p>The following examples assume that the current time value is GMT 2005 May 1 10h 15m 30s and that the day</p> |

Table 1. Typecasting Functions

| Function | Description |
|--|--|
| | <p>is the 1st Sunday of the month of May in 2005. The result of the evaluation is given after each example.</p> <pre>LT(GMT 2006): TRUE LT(GMT 2005 Dec): TRUE LT(Local 2005 May): TRUE or FALSE, depending on the time zone. LT(GMT 8h): FALSE LT(GMT 30m): TRUE LT(GMT May 10h): FALSE LT(GMT Jun 11h): TRUE LT(GMT Wed): TRUE LT(GMT May Sun_1): FALSE</pre> <p>Parameters:</p> <p><t> - Time</p> |
| <number>.TYPECAST_TIME_AT.MINUTES | Extracts the minute from the current system time and returns the value as an integer that can range from 0 to 59. |
| <number>.TYPECAST_TIME_AT.MONTH | Extracts the month from the current system time and returns the value as an integer that can range from 1 (January) to 12 (December). |
| <number>.TYPECAST_TIME_AT.RELATIVE_BOOT | Calculates the number of seconds that have elapsed after the most recent reboot or the number of seconds to the next scheduled reboot, depending on which is closer to the current time, and returns an integer. If the closest boot time is in the past, the integer is negative. If the closest boot time is in the future (scheduled reboot time), the integer is positive. |
| <number>.TYPECAST_TIME_AT.RELATIVE_NOW | Calculates the number of seconds between the current system time and the specified time, and returns the value as an integer. If the designated time is in the past, the integer is negative. If it is in the future, the integer is positive. |
| <number>.TYPECAST_TIME_AT.SECONDS | Extracts the seconds from the current system time and returns the value as an integer that can range from 0 to 59. |
| <number>.TYPECAST_TIME_AT.WEEKDAY | Returns an integer that corresponds to the day of the week; 0 for Sunday and 6 for Saturday. |
| <number>.TYPECAST_TIME_AT.WITHIN(<time1>, <time2>) | <p>Returns a Boolean value (TRUE or FALSE) that indicates whether the time value designated by <number> lies within all the ranges defined by lower and upper time value arguments <time1> and <time2>.</p> <p>If an element of time such as the day or the hour is left unspecified in the lower argument, <time1>, then it is assumed to have the lowest value possible for its range.</p> |

Table 1. Typecasting Functions

| Function | Description |
|--------------------------------|--|
| | <p>If an element is left unspecified in the upper argument , <time2>, then it is assumed to have the highest value possible for its range.</p> <p>If the year is specified in one of the arguments, then it must be specified in the other argument as well.</p> <p>Following are the ranges for different elements of time:</p> <ul style="list-style-type: none"> • month: 1-12 • day: 1-31 • weekday: 0-6 • hour: 0-23 • minutes: 0-59 • seconds: 0-59. <p>Each element of time in the lower time value argument defines a range in combination with the corresponding element in the upper time value argument. For the result to be TRUE, each element of time in the time value designated by <number> must lie in the corresponding range specified by the lower and upper arguments.</p> <p>The following examples assume that the current time value is GMT 2005 May 10 10h 15m 30s.and that the day is the second Tuesday of the month. The result of the evaluation is given after each example.</p> <p>WITHIN(GMT 2004, GMT 2006): TRUE WITHIN(GMT 2004 Jan, GMT 2006 Mar): FALSE (May doesn't fall in the Jan-Mar range.) WITHIN(GMT Feb, GMT): TRUE (May falls in the Feb-Dec range.) WITHIN(GMT Sun_1, GMT Sun_3): TRUE (2nd Tuesday lies within 1st Sunday and the 3rd Sunday.)</p> <p>WITHIN(GMT 2005 May 1 10h, GMT May 2005 1 17h): TRUE WITHIN(LOCAL 2005 May 1, LOCAL May 2005 1): The result depends on the NetScaler system's timezone.</p> <p>Parameters:</p> <p><time1> - Lower time value</p> <p><time2> - Upper time value</p> |
| <number>.TYPECAST_TIME_AT.YEAR | Extracts the year from the current system time and returns the value as a four-digit integer. |

Table 1. Typecasting Functions

| Function | Description |
|---|--|
| <code><prefix>.TYPECAST_NUM_T(<type>)</code> | <p>Casts numeric string data to a signed 32-bit number. The argument <code><type></code> can be one of the following:</p> <ul style="list-style-type: none"> • <code>DECIMAL</code>. Treat the string as a decimal number and cast to a signed 32-bit number. • <code>HEX</code>. Treat the string as a hexadecimal number and cast to a signed 32-bit number. • <code>DECIMAL_PREFIX</code>. Consider the part of the string up to the first occurrence of a character that is not a valid decimal character and cast to a signed 32-bit number. • <code>HEX_PREFIX</code>. Consider the part of the string up to the first occurrence of a character that is not a valid hexadecimal character and cast to a signed 32-bit number. <p>For example, the following policy extracts a numeric portion of a query string, adds 4 to the number, and inserts an HTTP header named <code>Company</code> with the resulting decimal value.</p> <pre>add rewrite action myadd_action insert_http_header Company "http.req.url.query.typecast_num_t(decimal) .add(4) "</pre> <pre>add rewrite policy myadd_policy true myadd_action bind rewrite global myadd_policy 300 END - type RES_DEFAULT</pre> <p>For example, this policy would extract "4444" from the following URL stub:</p> <pre>/test/file.html?4444</pre> <p>The action that is associated with the policy would insert the following HTTP response header:</p> <pre>Company: 4448\r\n</pre> |
| <code><prefix>.TYPECAST_NUM_AT</code> | <p>Casts a number of any data type to a number of data type integer.</p> |
| <code><prefix>.TYPECAST_DOUBLE_AT</code> | <p>Casts a number of any data type to a number of data type double.</p> |
| <code><prefix>.TYPECAST_UNSIGNED_LONG_AT</code> | <p>Casts a number of any data type to a number of data type unsigned long.</p> |
| <code><prefix>.TYPECAST_NUM_T(<type>, <default>)</code> | <p>Casts string data to a signed 32-bit number. If the typecasting operation raises an undefined (<code>UNDEF</code>)</p> |

Table 1. Typecasting Functions

| Function | Description |
|---|---|
| | condition, the function returns the value specified for default. The type argument takes the values specified for <code>TYPECAST NUM T(<type>)</code> . |
| <code><prefix>.TYPECAST_UNSIGNED_LONG_T(<type>)</code> | Casts string data to data of type unsigned long. The argument can be one of the following: <ul style="list-style-type: none"> • <code>DECIMAL</code>. Treat the string as a decimal number and cast to unsigned long. • <code>HEX</code>. Treat the string as a hexadecimal number and cast to unsigned long. • <code>DECIMAL_PREFIX</code>. Consider the part of the string up to the first occurrence of a character that is not a valid decimal character and cast to unsigned long. • <code>HEX_PREFIX</code>. Consider the part of the string up to the first occurrence of a character that is not a valid hexadecimal character and cast to unsigned long. |
| <code><prefix>.TYPECAST_UNSIGNED_LONG_T(<type>, <default>)</code> | Casts string data to data of type unsigned long. If the typecasting operation raises an undefined (<code>UNDEF</code>) condition, the function returns the value specified for default. The type argument takes the values specified for <code>TYPECAST UNSIGNED LONG T(<type>)</code> . |