

The `UTL_URL` package has two functions: `ESCAPE` and `UNESCAPE`.

This chapter contains the following topics:

- [Overview](#)
- [Exceptions](#)
- [Examples](#)
- [Summary of UTL_URL Subprograms](#)



See Also:

[UTL_HTTP](#)

UTL_URL Overview

A Uniform Resource Locator (URL) is a string that identifies a Web resource, such as a page or a picture. Use a URL to access such resources by way of the HyperText Transfer Protocol (HTTP).

For example, the URL for Oracle's Web site is:

```
http://www.oracle.com
```

Normally, a URL contains English alphabetic characters, digits, and punctuation symbols. These characters are known as the *unreserved characters*. Any other characters in URLs, including multibyte characters or binary octet codes, must be escaped to be accurately processed by Web browsers or Web servers. Some punctuation characters, such as dollar sign (\$), question mark (?), colon (:), and equals sign (=), are reserved as delimiters in a URL. They are known as the *reserved characters*. To literally process these characters, instead of treating them as delimiters, they must be escaped.

The unreserved characters are:

- A through Z, a through z, and 0 through 9
- Hyphen (-), underscore (_), period (.), exclamation point (!), tilde (~), asterisk (*), accent ('), left parenthesis ((), right parenthesis ())

The reserved characters are:

- Semi-colon (;) slash (/), question mark (?), colon (:), at sign (@), ampersand (&), equals sign (=), plus sign (+), dollar sign (\$), percentage sign (%), and comma (,)

The `UTL_URL` package has two functions that provide escape and unescape mechanisms for URL characters. Use the escape function to escape a URL before the URL is used to fetch a Web page by way of the `UTL_HTTP` package. Use the unescape function to unescape an escaped URL before information is extracted from the URL.

For more information, refer to the Request For Comments (RFC) document RFC2396. Note that this URL escape and unescape mechanism is different from the `x-www-form-urlencoded` encoding mechanism described in the HTML specification:

`http://www.w3.org/TR/html`

UTL_URL Exceptions

UTL_URL raises an exception when it encounter a processing issue.

The following table lists the exceptions that can be raised when the UTL_URL package API is invoked.

Table 304-1 UTL_URL Exceptions

Exception	Error Code	Reason
BAD_URL	29262	The URL contains badly formed escape code sequences
BAD_FIXED_WIDTH_CHARSET	29274	Fixed-width multibyte character set is not allowed as a URL character set.

UTL_URL Examples

UTL_URL can be used for encoding and decoding.

You can implement the `x-www-form-urlencoded` encoding using the `UTL_URL.ESCAPE` function as follows:

```
CREATE OR REPLACE FUNCTION form_url_encode (
    data    IN VARCHAR2,
    charset IN VARCHAR2) RETURN VARCHAR2 AS
BEGIN
    RETURN utl_url.escape(data, TRUE, charset); -- note use of TRUE
END;
```

For decoding data encoded with the `form-URL-encode` scheme, the following function implements the decoding scheme:

```
CREATE OR REPLACE FUNCTION form_url_decode(
    data    IN VARCHAR2,
    charset IN VARCHAR2) RETURN VARCHAR2 AS
BEGIN
    RETURN utl_url.unescape(
        replace(data, '+', ' '),
        charset);
END;
```

Summary of UTL_URL Subprograms

This table lists and briefly describes the UTL_URL subprograms.

Table 304-2 UTL_URL Package Subprograms

Subprogram	Description
ESCAPE Function	Returns a URL with illegal characters (and optionally reserved characters) escaped using the %2-digit-hex-code format
UNESCAPE Function	Unescapes the escape character sequences to their original forms in a URL. Convert the %XX escape character sequences to the original characters

ESCAPE Function

This function returns a URL with illegal characters (and optionally reserved characters) escaped using the %2-digit-hex-code format.

Syntax

```
UTL_URL.ESCAPE (  
    url                      IN VARCHAR2 CHARACTER SET ANY_CS,  
    escape_reserved_chars    IN BOOLEAN DEFAULT FALSE,  
    url_charset              IN VARCHAR2 DEFAULT utl_http.body_charset)  
RETURN VARCHAR2;
```

Parameters

Table 304-3 ESCAPE Function Parameters

Parameter	Description
url	The original URL
escape_reserved_chars	Indicates whether the URL reserved characters should be escaped. If set to TRUE, both the reserved and illegal URL characters are escaped. Otherwise, only the illegal URL characters are escaped. The default value is FALSE.
url_charset	When escaping a character (single-byte or multibyte), determine the target character set that character should be converted to before the character is escaped in %hex-code format. If url_charset is NULL, the database charset is assumed and no character set conversion will occur. The default value is the current default body character set of the UTL_HTTP package, whose default value is ISO-8859-1. The character set can be named in Internet Assigned Numbers Authority (IANA) or in the Oracle naming convention.

Usage Notes

Use this function to escape URLs that contain illegal characters as defined in the URL specification RFC 2396. The legal characters in URLs are:

- A through Z, a through z, and 0 through 9
- Hyphen (-), underscore (_), period (.), exclamation point (!), tilde (~), asterisk (*), accent ('), left parenthesis ((), right parenthesis ())

The reserved characters consist of:

- Semi-colon (;) slash (/), question mark (?), colon (:), at sign (@), ampersand (&), equals sign (=), plus sign (+), dollar sign (\$), and comma (,)

Many of the reserved characters are used as delimiters in the URL. You should escape characters beyond those listed here by using `escape_url`. Also, to use the reserved characters in the name-value pairs of the query string of a URL, those characters must be escaped separately. An `escape_url` cannot recognize the need to escape those characters because once inside a URL, those characters become indistinguishable from the actual delimiters. For example, to pass a name-value pair `$logon=HR/<password>` into the query string of a URL, escape the `$` and `/` separately as `%24logon=HR%2F<password>` and use it in the URL.

Normally, you will escape the entire URL, which contains the reserved characters (delimiters) that should not be escaped. For example:

```
utl_url.escape('http://www.acme.com/a url with space.html')
```

Returns:

```
http://www.acme.com/a%20url%20with%20space.html
```

In other situations, you may want to send a query string with a value that contains reserved characters. In that case, escape only the value fully (with `escape_reserved_chars` set to `TRUE`) and then concatenate it with the rest of the URL. For example:

```
url := 'http://www.acme.com/search?check=' || utl_url.escape  
( 'Is the use of the "$" sign okay?', TRUE );
```

This expression escapes the question mark (?), dollar sign (\$), and space characters in 'Is the use of the "\$" sign okay?' but not the ? after search in the URL that denotes the use of a query string.

The Web server that you intend to fetch Web pages from may use a character set that is different from that of your database. In that case, specify the `url_charset` as the Web server character set so that the characters that need to be escaped are escaped in the target character set. For example, a user of an EBCDIC database who wants to access an ASCII Web server should escape the URL using `US7ASCII` so that a space is escaped as `%20` (hex code of a space in ASCII) instead of `%40` (hex code of a space in EBCDIC).

This function does not validate a URL for the proper URL format.

UNESCAPE Function

This function unescapes the escape character sequences to its original form in a URL, to convert the `%XX` escape character sequences to the original characters.

Syntax

```
UTL_URL.UNESCAPE (  
    url           IN VARCHAR2 CHARACTER SET ANY_CS,  
    url_charset   IN VARCHAR2 DEFAULT utl_http.body_charset)  
    RETURN VARCHAR2;
```

Parameters

Table 304-4 UNESCAPE Function Parameters

Parameter	Description
url	The URL to unescape

Table 304-4 (Cont.) UNESCAPE Function Parameters

Parameter	Description
<code>url_charset</code>	After a character is unescaped, the character is assumed to be in the <code>source_charset</code> character set and it will be converted from the <code>source_charset</code> to the database character set before the URL is returned. If <code>source_charset</code> is <code>NULL</code> , the database charset is assumed and no character set conversion occurred. The default value is the current default body character set of the <code>UTL_HTTP</code> package, whose default value is "ISO-8859-1". The character set can be named in Internet Assigned Numbers Authority (IANA) or Oracle naming convention.

Usage Notes

The Web server that you receive the URL from may use a character set that is different from that of your database. In that case, specify the `url_charset` as the Web server character set so that the characters that need to be unescaped are unescaped in the source character set. For example, a user of an EBCDIC database who receives a URL from an ASCII Web server should unescape the URL using `US7ASCII` so that `%20` is unescaped as a space (0x20 is the hex code of a space in ASCII) instead of a ? (because 0x20 is not a valid character in EBCDIC).

This function does not validate a URL for the proper URL format.