Storing .Net Timestamps in an Oracle Database

October 4, 2008

The precision of a MS .NET DateTime value is more than an Oracle Date can handle.

MS .NET expresses DateTime up to milliseconds (‘d-MM-yyyy HH:mm:ss.fff’) whereas Oracle only uses seconds as smallest time unit (‘dd-mm-yyyy HH24:MI:SS’). So if you want to store .NET DateTime in an Oracle DB you’ll lose precision. The milliseconds will be truncated.

How to overcome this!

The .NET framework comes to the rescue by allowing every DateTime to be converted into a number. DateTime.Ticks property returns the number of elapsed 100 nanosecond intervals since January 1st 1601 for the DateTime in question. So we just store this in a Number field of Oracle.

A drawback is these ticks are difficult to read. For example: December 14, 2007, at 15:23 becomes 2193385800000000 in the DB. If I want to write a select query to do some lookups I need to convert the dates into ticks and vice versa. This becomes a pain very fast.

Well, it became my pain very fast so I wrote a Oracle package. Remember this is not a full solution since Oracle will truncate the milliseconds. But it is good enough to find, for example, all records between this and that day without requiring to do the conversion in code upfront.

The Oracle Package: (example below)

```
CREATE OR REPLACE PACKAGE NET_DATE_UTIL
AS
  FUNCTION DATE_TO_TICKS(P_DATE IN DATE) RETURN NUMBER;
  FUNCTION TICKS_TO_DATE(P_TICKS IN NUMBER) RETURN DATE;
  FUNCTION TICKS_TO_VARCHAR_FRMSTR(P_TICKS IN NUMBER,P_FORMAT IN VARCHAR) RETURN VARCHAR;
  FUNCTION TICKS_TO_VARCHAR(P_TICKS IN NUMBER) RETURN VARCHAR;
END;

CREATE OR REPLACE PACKAGE BODY NET_DATE_UTIL AS

-- NET Ticks expresses a timestamp in nanoseconds since 01-01-1601 01:00:00
DOTNET_BASE_DATE DATE := TO_DATE('01-01-1601 01:00:00','MM-DD-YY HH24:MI:SS');
CONVERSION_FACTOR NUMBER := 24*60*60*1000*10000;

FUNCTION DATE_TO_TICKS (P_DATE IN DATE) RETURN NUMBER IS
BEGIN
  RETURN (P_DATE-DOTNET_BASE_DATE)*(CONVERSION_FACTOR);
END;

FUNCTION TICKS_TO_DATE (P_TICKS IN NUMBER) RETURN DATE IS
```

```
BEGIN
  RETURN DOTNET_BASE_DATE + (P_TICKS/CONVERSION_FACTOR);
END;

FUNCTION TICKS_TO_VARCHAR_FRMSTR (P_TICKS IN NUMBER,P_FORMAT IN VARCHAR) RETURN VARCHAR IS
BEGIN
  RETURN TO_CHAR(TICKS_TO_DATE(P_TICKS),P_FORMAT);
END;

FUNCTION TICKS_TO_VARCHAR (P_TICKS IN NUMBER) RETURN VARCHAR IS
BEGIN
  RETURN TICKS_TO_VARCHAR_FRMSTR(P_TICKS,'DD-MM-YYYY');
END;

END;
/
```

Example:

```
SELECT NET_DATE_UTIL.TICKS_TO_DATE(TimestampField)
FROM MyTable
WHERE TimestampField < NET_DATE_UTIL.DATE_TO_TICKS(SYSDATE);
```

The two last functions convert the dates into strings. The last function will return the date according to a fixed format ‘D-MM-YYYY’ The first allows you to pass the format, in which you want to date to be returned. Any valid Oracle format is allowed.

```
SELECT NET_DATE_UTIL.TICKS_TO_VARCHAR_FRMSTR(TimestampField,'M-DD-YY HH24:MI')
FROM MyTable;
```