| Without century (yy) (1) | With century (yyyy) | Standard | Input/Output (3) |
| --- | --- | --- | --- |
| - | **0** or **100**(1,2) | Default for datetime and smalldatetime | mon dd yyyy hh:miAM (or PM) |
| **1** | **101** | U.S. | 1 = mm/dd/yy 101 = mm/dd/yyyy |
| **2** | **102** | ANSI | 2 = yy.mm.dd 102 = yyyy.mm.dd |
| **3** | **103** | British/French | 3 = dd/mm/yy 103 = dd/mm/yyyy |
| **4** | **104** | German | 4 = dd.mm.yy 104 = dd.mm.yyyy |
| **5** | **105** | Italian | 5 = dd-mm-yy 105 = dd-mm-yyyy |
| **6** | **106** (1) | - | 6 = dd mon yy 106 = dd mon yyyy |
| **7** | **107** (1) | - | 7 = Mon dd, yy 107 = Mon dd, yyyy |
| **8** | **108** | - | hh:mi:ss |
| - | **9** or **109**(1,2) | Default + milliseconds | mon dd yyyy hh:mi:ss:mmmAM (or PM) |
| **10** | **110** | USA | 10 = mm-dd-yy 110 = mm-dd-yyyy |
| **11** | **111** | JAPAN | 11 = yy/mm/dd 111 = yyyy/mm/dd |
| **12** | **112** | ISO | 12 = yymmdd 112 = yyyymmdd |
| - | **13** or **113**(1,2) | Europe default + milliseconds | dd mon yyyy hh:mi:ss:mmm(24h) |
| **14** | **114** | - | hh:mi:ss:mmm(24h) |
| - | **20** or **120** (2) | ODBC canonical | yyyy-mm-dd hh:mi:ss(24h) |
| - | **21** or **121** (2) | ODBC canonical (with milliseconds) default for time, date, datetime2, and datetimeoffset | yyyy-mm-dd hh:mi:ss.mmm(24h) |
| - | **126** (4) | ISO8601 | yyyy-mm-ddThh:mi:ss.mmm (no spaces) Note: When the value for milliseconds (mmm) is 0, the millisecond value is not displayed. For example, the value '2012-11-07T18:26:20.000 is displayed as '2012-11-07T18:26:20'. |
| - | **127**(6, 7) | ISO8601 with time zone Z. | yyyy-mm-ddThh:mi:ss.mmmZ (no spaces) Note: When the value for milliseconds (mmm) is 0, the milliseconds value is not displayed. For example, the value '2012-11-07T18:26:20.000 is displayed as '2012-11-07T18:26:20'. |
| - | **130**(1,2) | Hijri (5) | dd mon yyyy hh:mi:ss:mmmAM In this style, mon represents a multi-token Hijri unicode representation of the full month's name. This value does not render correctly on a default US installation of SSMS. |
| - | **131** (2) | Hijri (5) | dd/mm/yyyy hh:mi:ss:mmmAM |

1 These style values return nondeterministic results. Includes all (yy) (without century) styles and a subset of (yyyy) (with century) styles.+

2 The default values (style\* 0\* or 100, 9 or 109, 13 or 113, 20 or 120, and 21 or 121) always return the century (yyyy).

3 Input when you convert to datetime; output when you convert to character data.

4 Designed for XML use. For conversion from datetime or smalldatetime to character data, the output format is as described in the previous table.

5 Hijri is a calendar system with several variations. SQL Server uses the Kuwaiti algorithm.

Important

By default, SQL Server interprets two-digit years based on a cutoff year of 2049. That is, the two-digit year 49 is interpreted as 2049 and the two-digit year 50 is interpreted as 1950. Many client applications, such as those based on Automation objects, use a cutoff year of 2030. SQL Server provides the two digit year cutoff configuration option that changes the cutoff year used by SQL Server and allows for the consistent treatment of dates. We recommend specifying four-digit years.

6 Only supported when casting from character data to datetime or smalldatetime. When character data that represents only date or only time components is cast to the datetime or smalldatetime data types, the unspecified time component is set to 00:00:00.000, and the unspecified date component is set to 1900-01-01.

7The optional time zone indicator, Z, is used to make it easier to map XML datetime values that have time zone information to SQL Server datetime values that have no time zone. Z is the indicator for time zone UTC-0. Other time zones are indicated with HH:MM offset in the + or - direction. For example: 2006-12-12T23:45:12-08:00.