Dates in SpreadsheetML

Detail

We agree that it is important for SpreadsheetML to support ISO 8601 dates, and that the current specification could be improved in several ways:

- The ability to store values as dates, rather than as numeric serial values formatted as dates
- Support for dates before 1900
- Proper treatment of 1900 as not being a leap year

We propose altering the specification to use both the lexical space and value space of ISO 8601 in its representation of dates.

To support the lexical space of ISO 8601 we must update the description of date and time representation within SpreadsheetML. These changes also add the necessary schema and provide examples of using ISO 8601 dates in SpreadsheetML.

In order to maintain compatibility with the existing corpus of binary documents, the specification continues to allow dates to be converted to and from numeric serial values. We propose extending this conversion to correctly support the value space of ISO 8601 from proleptic Gregorian 0000 to Gregorian 9999, and to correctly treat 1900 as not being a leap year. Because the previously specified serial conversion treated 1900 as a leap year (and therefore produced different serial values), we cannot simply extend the implementation and instead must add a “dateCompatibility” switch to enable the legacy behaviors.

We propose the following changes to the specification:

Part 4, §3.2.27, page 1,908, line 26:

[*Example*:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<workbook xmlns="http://schemas.openxmlformats.org/spreadsheetml/2006/5/main"
mlns:r="http://schemas.openxmlformats.org/officeDocument/2006/relationships">
  <fileVersion lastEdited="4" lowestEdited="4" rupBuild="4017"/>
  <workbookPr date1904="1" dateCompatibility="false" vbName="ThisWorkbook"
defaultThemeVersion="123820"/>
```

Part 4, §3.2.28, page 1,911, line 1:

[*Example*:

```
<workbookPr dateCompatibility="false" date1904="1"showObjects="none"
saveExternalLinkValues="0"
  defaultThemeVersion="123820"/>
```

end example]

Part 4, §3.2.28, page 1,912:

Parent Elements	
date1904 (Date 1904)	~~Specifies a boolean value that indicates whether the date systems used in the workbook starts in 1904.~~ ~~A value of on, 1, or true indicates the date system starts in 1904.~~ ~~A value of off, 0, or false indicates the workbook uses the 1900 date system, where 1/1/1900 is the first day in the system.~~ Value that indicates whether to use the 1900 or 1904 date base whenever serial values in the workbook should be converted to dates. A value of 1 or true indicates the workbook uses a date system based in 1904, either the 1904 date base or the 1904 backward compatibility date base, as specified by the value of the dateCompatibility attribute. A value of 0 or false indicates the workbook uses a date system based in 1900, either the 1900 date base or the 1900 backward compatibility *date base*, as specified by the value of the dateCompatibility attribute. (See §3.17.4.1 for the definition of the four valid date bases.) The default value for this attribute is false. The possible values for this attribute are defined by the XML Schema boolean datatype.
dateCompatibility (Date Compatibility)	Specifies whether the date base specified by the date1904 attribute should be treated as a compatibility date base or should support the full ISO 8601 date range. A value of 1 or true indicates that the date system in use is either the 1900 backward compatibility date base or the 1904 backward compatibility date base, as specified by the value of the date1904 attribute. A value of 0 or false indicates that the date system is either the 1900 date base or 1904 date-base system, based on the ISO 8601 date range, as specified by the value of the date1904 attribute.

Parent Elements	
	(See §3.17.4.1 for the definition of the four valid date bases.) The default value for this attribute is true. The possible values for this attribute are defined by the XML Schema boolean datatype.

Part 4, §3.2.28, page 1,915, line 3:

The following XML Schema fragment defines the contents of this element:

```
<complexType name="CT_WorkbookPr">
    <attribute name="date1904" type="xsd:boolean" use="optional" default="false"/>
    <attribute name="dateCompatibility" type="xsd:boolean" use="optional" default="true"/>
    <attribute name="showObjects" type="ST_Objects" use="optional" default="all"/>
    <attribute name="showBorderUnselectedTables" type="xsd:boolean" use="optional" default="true"/>
    <attribute name="filterPrivacy" type="xsd:boolean" use="optional" default="false"/>
    <attribute name="promptedSolutions" type="xsd:boolean" use="optional" default="false"/>
    <attribute name="showInkAnnotation" type="xsd:boolean" use="optional" default="true"/>
    <attribute name="backupFile" type="xsd:boolean" use="optional" default="false"/>
    <attribute name="saveExternalLinkValues" type="xsd:boolean" use="optional" default="true"/>
    <attribute name="updateLinks" type="ST_UpdateLinks" use="optional" default="userSet"/>
    <attribute name="codeName" type="xsd:string" use="optional"/>
    <attribute name="hidePivotFieldList" type="xsd:boolean" use="optional" default="false"/>
    <attribute name="showPivotChartFilter" type="xsd:boolean" default="false"/>
    <attribute name="allowRefreshQuery" type="xsd:boolean" use="optional" default="false"/>
    <attribute name="publishItems" type="xsd:boolean" use="optional" default="false"/>
    <attribute name="checkCompatibility" type="xsd:boolean" use="optional" default="false"/>
    <attribute name="autoCompressPictures" type="xsd:boolean" use="optional" default="true"/>
    <attribute name="refreshAllConnections" type="xsd:boolean" use="optional" default="false"/>
    <attribute name="defaultThemeVersion" type="xsd:unsignedInt" use="optional"/>
</complexType>
```

Changes to Part 4, §3.3.1.3, page 2,038, lines 9–21:

3.3.1.93 v (Cell Value)

This element expresses the value contained in a cell. If the cell contains a string, then this value is an index into the shared string table, pointing to the actual string value. Otherwise, the value of the cell is expressed directly in this element. Cells containing formulas express the last calculated result of the formula in this element.

For applications not wanting to implement the shared string table, an 'inline string' may be expressed in an <is> element under <c> (instead of a <v> element under <c>),in the same way a string would be expressed in the shared string table. See <is> for an example.

[*Example*: In this example cell B4 contains the number "360" and C4 contains the UTC date 22 November 1976, 08:30.

```
<c r="B4">
  <v>360</v>
</c>
<c r="C4" t="d">
  <v>1976-11-22T08:30Z</v>
</c>
```

end example]

The possible values for this element are defined by the ST_Xstring simple type (**§¡Error! No se encuentra el origen de la referencia.**).

Parent Elements
c (**§¡Error! No se encuentra el origen de la referencia.**); cell (**§¡Error! No se encuentra el origen de la referencia.**); nc (**§¡Error! No se encuentra el origen de la referencia.**); oc (**§¡Error! No se encuentra el origen de la referencia.**); tp (**§¡Error! No se encuentra el origen de la referencia.**)

Changes to Part 4 §3.3.2.5, page 2,047, line 11 through page 2,048, line 6:

3.3.2.5 dynamicFilter (Dynamic Filter)

This collection specifies dynamic filter criteria. These criteria are considered dynamic because they can change, either with the data itself (e.g., "above average") or with the current system date (e.g., show values for "today"). For any cells whose values do not meet the specified criteria, the corresponding rows shall be hidden from view when the fitler is applied.

[*Example*:

```
<filterColumn colId="0">
  <dynamicFilter type="today"/>
</filterColumn>
```

end example]

Parent Elements
filterColumn (**§¡Error! No se encuentra el origen de la referencia.**)

Attributes	Description
maxVal (Max Value)	A maximum value for dynamic filter. ~~It~~ maxVal/maxValIso shall be required for today, yesterday, tomorrow, nextWeek, thisWeek, lastWeek, nextMonth, thisMonth, lastMonth, nextQuarter, thisQuarter, lastQuarter, nextYear, thisYear, lastYear, and yearToDate. The above criteria are based on a value range. For example, if today's date is September 22nd, then the range for thisWeek is the values greater than or equal to September 17

	and less than September 24. In the thisWeek range, the lower value is expressed using val or valIso. The higher value is expressed using maxVal or maxValIso. These dynamic filter shall not require val/valIso ~~/~~or maxVal/maxValIso: Q1, Q2, Q3, Q4, M1, M2, M3, M4, M5, M6, M7, M8, M9, M10, M11 and M12. The above criteria shall not specify the range using val/valIso and maxVal/maxValIso because Q1 always starts from M1 to M3, and M1 is always January. These types of dynamic filters shall use val/valIso and shall not use maxVal/maxValIso: aboveAverage and belowAverage If maxValIso and maxVal are both present, maxValIso shall take precedence. The possible values for this attribute are defined by the XML Schema double datatype.
maxValIso (Max ISO Value)	A maximum value for dynamic filter. maxVal/maxValIso shall be required for today, yesterday, tomorrow, nextWeek, thisWeek, lastWeek, nextMonth, thisMonth, lastMonth, nextQuarter, thisQuarter, lastQuarter, nextYear, thisYear, lastYear, and yearToDate. The above criteria are based on a value range; that is, if today's date is September 22nd, then the range for thisWeek is the values greater than or equal to September 17 and less than September 24. In the thisWeek range, the lower value is expressed using val or valIso. The higher value is expressed using maxVal of maxValIso. These dynamic filters shall not require val/valIso or maxVal/maxValIso: Q1, Q2, Q3, Q4, M1, M2, M3, M4, M5, M6, M7, M8, M9, M10, M11 and M12. The above criteria shall not specify the range using val/valIso and maxVal/maxValIso because Q1 always starts from M1 to M3, and M1 is always January. These types of dynamic filters shall use val/valIso and shall not use maxVal/maxValIso: aboveAverage and belowAverage If maxValIso and maxVal are both present, maxValIso shall take precedence. The possible values for this attribute are defined by the XML Schema dateTime datatype.
type (Dynamic filter type)	Dynamic filter type, e.g., “today” or “nextWeek”. The possible values for this attribute are defined by the ST_DynamicFilterType simple type (§**¡Error! No se encuentra el origen de la referencia.**).
val (Value)	A minimum value for dynamic filter. See description of maxVal/maxValIso to understand when val is required. If valIso and val are both present, valIso shall take precedence.

	The possible values for this attribute are defined by the XML Schema double datatype.
valIso (ISO Value)	A minimum value for dynamic filter. (See description of maxVal/maxValIso to understand when valIso is required.) If valIso and val are both present, valIso shall take precedence. The possible values for this attribute are defined by the XML Schema dateTime datatype.

The following XML Schema fragment defines the contents of this element:

```
<complexType name="CT_DynamicFilter">
  <attribute name="type" type="ST_DynamicFilterType" use="required"/>
  <attribute name="val" type="xsd:double" use="optional"/>
  <attribute name="valIso" type="xsd:dateTime" use="optional"/>
  <attribute name="maxVal" type="xsd:double" use="optional"/>
  <attribute name="maxValIso" type="xsd:dateTime" use="optional"/>

</complexType>
```

Part 4, §3.8.31, page 2,140, line 10:

To display	As	Use this code
Years	~~1900~~date-base minimum value–9999	Yyyy

Part 4, §3.8.31, page 2,140, line 12:

See §3.17.4.1 for detail on possible date bases~~special handling of certain days in the year 1900~~.

Changes to Part 4 §3.10.1.67, page 2,323, line 8 through 2,328, line 34:

3.10.1.67 pivotCacheDefinition (PivotCache Definition)

Represents the pivotCacheDefinition part. This part defines each field in the source data, including the name, the string resources of the instance data (for shared items), and information about the type of data that appears in the field.

[*Example*:

```
<pivotCacheDefinition xmlns="…" xmlns:r="…" r:id="rId1" refreshedBy="AnonUser"
  refreshedDate="2006-05-22T10:07:16Z" createdVersion="3" refreshedVersion="3"
  minRefreshableVersion="3" recordCount="182">
```

```
    <cacheSource type="worksheet">
      <worksheetSource name="Table1"/>
    </cacheSource>
    <cacheFields count="28">
      <cacheField name="Customer Name" numFmtId="0">
      <cacheField name="Postal Code" numFmtId="0">
        <sharedItems/>
      </cacheField>
      <cacheField name="Product Category" numFmtId="0">
        <sharedItems count="1">
          <s v="Bikes"/>
        </sharedItems>
      </cacheField>
      <cacheField name="Year" numFmtId="0">
        <sharedItems count="1">
          <s v="2001"/>
        </sharedItems>
      </cacheField>
      <cacheField name="Quarter" numFmtId="0">
        <sharedItems containsSemiMixedTypes="0" containsString="0"
          containsNumber="1" containsInteger="1" minValue="3" maxValue="3"
          count="1">
          <n v="3"/>
        </sharedItems>
      </cacheField>
    </cacheFields>
  </pivotCacheDefinition>
```

end example]

Parent Elements
Root element of SpreadsheetML Pivot Table Cache Definition part

Child Elements	Subclause
cacheFields (PivotCache Fields)	§**¡Error! No se encuentra el origen de la referencia.**
cacheHierarchies (PivotCache Hierarchies)	§**¡Error! No se encuentra**

	el origen de la referencia.
cacheSource (PivotCache Source Description)	§**¡Error! No se encuentra el origen de la referencia.**
calculatedItems (Calculated Items)	§**¡Error! No se encuentra el origen de la referencia.**
calculatedMembers (Calculated Members)	§**¡Error! No se encuentra el origen de la referencia.**
dimensions (OLAP Dimensions)	§**¡Error! No se encuentra el origen de la referencia.**
extLst (Future Feature Data Storage Area)	§**¡Error! No se encuentra el origen de la referencia.**
kpis (OLAP KPIs)	§**¡Error! No se encuentra el origen de la referencia.**
maps (OLAP Measure Group)	§**¡Error! No se encuentra el origen de la referencia.**

measureGroups (OLAP Measure Groups)	§**¡Error! No se encuentra el origen de la referencia.**
tupleCache (Tuple Cache)	§**¡Error! No se encuentra el origen de la referencia.**

1

Attributes	Description
backgroundQuery (Background Query)	Specifies a boolean value that indicates whether the application should query and retrieve records asynchronously from the cache. A value of on, 1, or true indicates the application will retrieve records asynchronously from the cache. A value of off, 0, or false indicates the application will retrieve records synchronously. The possible values for this attribute are defined by the XML Schema boolean datatype.
createdVersion (PivotCache Created Version)	Specifies the version of the application that created the cache. This attribute is application-dependent. The possible values for this attribute are defined by the XML Schema unsignedByte datatype.
enableRefresh (Enable PivotCache Refresh)	Specifies a boolean value that indicates whether the end-user can refresh the cache. This attribute depends on whether the application exposes a method for allowing end-users control over refreshing the cache via the user interface. A value of on, 1, or true indicates the end-user can refresh the cache. A value of off, 0, or false indicates the end-user cannot refresh the cache. The possible values for this attribute are defined by the XML Schema boolean datatype.
id (Relationship Identifier)	Specifies the unique identifier that corresponds to the related `pivotCacheRecords` part. See (§**¡Error! No se encuentra el origen de**

Namespace: .../officeDocument/2006/relationships	**la referencia.**) for more information. The possible values for this attribute are defined by the ST_RelationshipId simple type (§**¡Error! No se encuentra el origen de la referencia.**).
invalid (Invalid Cache)	Specifies a boolean value that indicates whether the cache is invalid and needs to be refreshed. A value of on, 1, or true indicates the cache is invalid and needs to be refreshed. A value of off, 0, or false indicates the cache does not need to be refreshed. The possible values for this attribute are defined by the XML Schema boolean datatype.
minRefreshableVersion (Minimum Version Required for Refresh)	Specifies the earliest version of the application that is required to refresh the cache. This attribute is application-dependent. The possible values for this attribute are defined by the XML Schema unsignedByte datatype.
missingItemsLimit (Missing Items Limit)	Specifies the number of unused items to allow before discarding unused items. This attribute is application-dependent. The application must specify a threshold for unused items. The possible values for this attribute are defined by the XML Schema unsignedInt datatype.
optimizeMemory (Optimize Cache for Memory)	Specifies a boolean value that indicates whether the application will apply optimizations to the cache to reduce memory usage. This attribute is application-dependent. This application must define its own cache optimization methods. The application must also decide whether to expose cache optimization status via the user interface or an object model. A value of on, 1, or true indicates the application will apply optimizations to the cache. A value of off, 0, or false indicates the application will not apply optimizations to the cache. The possible values for this attribute are defined by the XML Schema boolean datatype.
recordCount (PivotCache Record Count)	Specifies the number of records in the cache. The possible values for this attribute are defined by the XML Schema unsignedInt datatype.

refreshedBy (Last Refreshed By)	Specifies the name of the end-user who last refreshed the cache. This attribute is application-dependent and is specified by applications that track and store the identity of the current user. This attribute also depends on whether the application exposes mechanisms via the user interface whereby the end-user can refresh the cache. The possible values for this attribute are defined by the ST_Xstring simple type (§**¡Error! No se encuentra el origen de la referencia.**).
refreshedDate (Legacy PivotCache Last Refreshed Date)	Specifies the date when the cache was last refreshed as serial date value. This attribute depends on whether the application exposes mechanisms via the user interface whereby the end-user can refresh the cache. If refreshedDateIso and refreshedDate are both present, refreshedDateIso shall take precedence. The possible values for this attribute are defined by the XML Schema double datatype.
refreshedDateIso (PivotCache Last Refreshed Date ISO)	Specifies the date when the cache was last refreshed. This attribute depends on whether the application exposes mechanisms via the user interface whereby the end-user can refresh the cache. If refreshedDateIso and refreshedDate are both present, refreshedDateIso shall take precedence. The possible values for this attribute are defined by the XML Schema dateTime datatype.
refreshedVersion (PivotCache Last Refreshed Version)	Specifies the version of the application that last refreshed the cache. This attribute depends on whether the application exposes mechanisms via the user interface whereby the end-user can refresh the cache. The possible values for this attribute are defined by the XML Schema unsignedByte datatype.
refreshOnLoad (Refresh On Load)	Specifies a boolean value that indicates whether the application will refresh the cache when the workbook has been opened. A value of on, 1, or true indicates that application will refresh the cache when the workbook is loaded. A value of off, 0, or false indicates the application will not automatically refresh cached data. The end user must trigger refresh of the cache manually via the application user interface. The possible values for this attribute are defined by the XML Schema boolean datatype.

saveData (Save Pivot Records)	Specifies a boolean value that indicates whether the pivot records are saved with the cache. A value of on, 1, or true indicates pivot records are saved in the cache. A value of off, 0, or false indicates are not saved in the cache. The possible values for this attribute are defined by the XML Schema boolean datatype.
supportAdvancedDrill (Supports Attribute Drilldown)	Specifies whether the cache's data source supports attribute drilldown. The possible values for this attribute are defined by the XML Schema boolean datatype.
supportSubquery (Supports Subqueries)	Specifies whether the cache's data source supports subqueries. The possible values for this attribute are defined by the XML Schema boolean datatype.
tupleCache (Stores Cache for OLAP Functions)	Specifies a boolean value that indicates whether the PivotCache is used store information for OLAP sheet data functions. A value of on, 1, or true indicates information about OLAP sheet data functions are stored in the cache. A value of off, 0, or false indicates the PivotCache does not contain information about OLAP sheet data functions. The possible values for this attribute are defined by the XML Schema boolean datatype.
upgradeOnRefresh (Upgrade PivotCache on Refresh)	Specifies a boolean value that indicates whether the cache is scheduled for version upgrade. This attribute depends on whether the application exposes mechanisms via the user interface whereby the cache may be upgraded. A value of on, 1, or true indicates the cache is scheduled for upgrade. A value of off, 0, or false indicates the cache is not scheduled for upgrade. The possible values for this attribute are defined by the XML Schema boolean datatype.

The following XML Schema fragment defines the contents of this element:

```
<complexType name="CT_PivotCacheDefinition">
    <sequence>
        <element name="cacheSource" type="CT_CacheSource" minOccurs="1" maxOccurs="1"/>
        <element name="cacheFields" type="CT_CacheFields" minOccurs="1" maxOccurs="1"/>
        <element name="cacheHierarchies" minOccurs="0" type="CT_CacheHierarchies"/>
        <element name="kpis" minOccurs="0" type="CT_PCDKPIs"/>
        <element name="tupleCache" minOccurs="0" type="CT_TupleCache"/>
        <element name="calculatedItems" minOccurs="0" type="CT_CalculatedItems"/>
        <element name="calculatedMembers" type="CT_CalculatedMembers" minOccurs="0"/>
        <element name="dimensions" type="CT_Dimensions" minOccurs="0"/>
        <element name="measureGroups" type="CT_MeasureGroups" minOccurs="0"/>
        <element name="maps" type="CT_MeasureDimensionMaps" minOccurs="0"/>
        <element name="extLst" minOccurs="0" type="CT_ExtensionList"/>
    </sequence>
    <attribute ref="r:id" use="optional"/>
    <attribute name="invalid" type="xsd:boolean" use="optional" default="false"/>
    <attribute name="saveData" type="xsd:boolean" use="optional" default="true"/>
    <attribute name="refreshOnLoad" type="xsd:boolean" use="optional" default="false"/>
    <attribute name="optimizeMemory" type="xsd:boolean" use="optional" default="false"/>
    <attribute name="enableRefresh" type="xsd:boolean" use="optional" default="true"/>
    <attribute name="refreshedBy" type="ST_Xstring" use="optional"/>
    <attribute name="refreshedDate" type="xsd:double" use="optional"/>
    <attribute name="refreshedDateIso" type="xsd:dateTime" use="optional"/>
    <attribute name="backgroundQuery" type="xsd:boolean" default="false"/>
    <attribute name="missingItemsLimit" type="xsd:unsignedInt" use="optional"/>
    <attribute name="createdVersion" type="xsd:unsignedByte" use="optional" default="0"/>
    <attribute name="refreshedVersion" type="xsd:unsignedByte" use="optional" default="0"/>
    <attribute name="minRefreshableVersion" type="xsd:unsignedByte" use="optional" default="0"/>
    <attribute name="recordCount" type="xsd:unsignedInt" use="optional"/>
    <attribute name="upgradeOnRefresh" type="xsd:boolean" use="optional" default="false"/>
    <attribute name="tupleCache" type="xsd:boolean" use="optional" default="false"/>
    <attribute name="supportSubquery" type="xsd:boolean" use="optional" default="false"/>
    <attribute name="supportAdvancedDrill" type="xsd:boolean" use="optional" default="false"/>
</complexType>
```

Changes to Part 4, §3.17.4, page 2,522, lines 5–11:

3.17.4 Dates and Times

Each unique instant in SpreadsheetML time is represented as an ISO 8601 formatted string, which is made up of a date component, time component, and timezone component. ~~a distinct non-negative numeric *serial value*, which is made up of an integer date component and a fractional time component. As dates and times are numeric values, they can take part in arithmetic operations.~~

Numerous functions take dates and/or times as arguments ~~one or more serial values or strings representing dates and/or times~~. Functions that care only about the date shall ignore any time information that is provided. Functions that care only about the time shall ignore any date information that is provided.

[*Example*: The date 22 November 1976 at exactly 08:30 Pacific Standard Time (+08:00 UTC) could be represented in the following ways within SpreadsheetML. Note that this is not an exhaustive list of the representations:

1976-11-22T08:30:00,000+08:00
1976-11-22T16:30Z
end example]

For compatibility with existing spreadsheet applications, a consuming application should allow certain numeric serial values to be converted into dates and times for display or for use in calculations. These values should behave as defined in §3.17.4.1, §3.17.4.2, and §3.17.4.3.

Changes to Part 4, §3.17.4.1, page 2,522, line 12:

3.17.4.1 Date ~~Representation~~Conversion for Serial Values

A date which has been converted to a numeric value is a *serial value*. This is made up of an integer date component and a fractional time component. Going forward in time, the date component of a serial value increases by 1 each day. A serial value represents a UTC date and time, and as such has no timezone information.

There are ~~two~~four different bases which may be used for converting dates into serial values:

- In the *1900 date-base system*, the lower limit is January 1, ~~1900~~0000, which has serial value ~~1~~-693959. The upper-limit is December 31, 9999, which has serial value 2,958,465.
- In the *1904 date-base system*, the lower limit is January 1, ~~1904~~0000, which has serial value ~~0~~-695421. The upper-limit is December 31, 9999, which has serial value 2,957,003.
- In the *1900 backward compatibility date-base system*, the lower limit is January 1, 1900, which has serial value 1. The upper limit is December 31, 9999, which has serial value 2,958,465.
- In the *1904 backward compatibility date-base system*, the lower limit is January 1, 1904, which has serial value 0. The upper limit is December 31, 9999, which has serial value 2,957,003.

A serial value outside of the range for its date base system is ill-formed.

The ~~As to which~~ date base system an implementation uses by default and ~~or~~ whether it allows its users to switch between date base systems, is unspecified. ~~See §3.17.6.7 for XML-related details.~~ [*Note*: As the XML allows either date base system to be used, an implementation must be able to deal with ~~both~~ all systems. *end note*]

[*Note:* The 1900 date-base system is the preferred system to be used by applications when converting serial values to dates. The use of the 1900 backward compatibility or 1904 backward compatibility date-base system should be avoided. *end note*]

The date-base system is recorded in the Workbook part's XML by the presence or absence of the date1904 attribute of the workbookPr element. A value of 1 for this attribute indicates 1904.

[*Example*:

1900 date-base system: <workbookPr showObjects="all"/>
1904 date-base system: <workbookPr date1904="1" showObjects="all"/>

end example]

For legacy reasons, an implementation using the 1900 date base system shall treat 1900 as though it was a leap year. [*Note*: That is, serial value 59 corresponds to February 28, and serial value 61 corresponds to March 1, the next day, allowing the (non-existent) date February 29 to have the serial value 60. *end note*] A consequence of this is that for dates between January 1 and February 28, WEEKDAY shall return a value for the day immediately prior to the correct day, so that the (non-existent) date February 29, 1900 has a day-of-the-week that immediately follows that of February 28, and immediately precedes that of March 1, 1900.

[*Example*: For the 1900 date base system:

05-Aug-1893 is represented by the serial value -2338.0000000…
~~DATEVALUE("~~01-Jan-1900~~") results in~~ is represented by the serial value ~~1~~2.0000000…
~~DATEVALUE("~~03-Feb-1910~~") results in~~ is represented by the serial value 3687.0000000…
~~DATEVALUE("~~01-Feb-2006~~") results in~~ is represented by the serial value 38749.0000000…
~~DATEVALUE("~~31-Dec-9999~~") results in~~ is represented by the serial value 2958465.0000000…

For the 1904 date base system:

05-Aug-1893 is represented by the serial value -3800.0000000…
~~DATEVALUE("~~01-Jan-1904~~") results in~~ is represented by the serial value 0.0000000…
~~DATEVALUE("~~03-Feb-1910~~") results in~~ is represented by the serial value 2225.0000000…
~~DATEVALUE("~~01-Feb-2006~~") results in~~ is represented by the serial value 37287.0000000…
~~DATEVALUE("~~31-Dec-9999~~") results in~~ is represented by the serial value 2957003.0000000…

end example]

Changes to Part 4, §3.17.4.2, page 2,523, line 21:

3.17.4.2 Time ~~Representation~~Conversion for Serial Values

The time component of a serial value ranges in value from 0–0.99999999, and represents times from 0:00:00 (12:00:00 AM) to 23:59:59 (11:59:59 P.M.), respectively.

Going forward in time, the time component of a serial value increases by 1/86,400 each second. [*Note*: As such, the time 12:00 has a serial value time component of 0.5. *end note*]

[*Example*:

~~TIMEVALUE("~~00:00:00~~") results in~~ the serial value 0.0000000…

~~TIMEVALUE("~~00:00:01~~") results in~~ is represented by the serial value 0.0000115…
~~TIMEVALUE("~~10:05:54~~") results in~~ is represented by the serial value 0.4207639…
~~TIMEVALUE("~~12:00:00~~") results in~~ is represented by the serial value 0.5000000…
~~TIMEVALUE("~~23:59:59~~") results in~~ is represented by the serial value 0.9999884…

end example]

Changes to Part 4, §3.17.4.3, page 2,523, line 35:

3.17.4.3 Combined Date and Time ~~Representation~~Conversion for Serial Values

Any date component can be added to any time component to produce a serial value for that date/time combination.

[*Example*: For the 1900 date base system:

DATE(1893,8,5)+TIME(0,0,1) results in the serial value -2337.999989…
DATE(1910,2,3)+TIME(10,5,54) results in the serial value 3687.4207639…
DATE(1900,1,1)+TIME(12,0,0) results in the serial value 1.5000000…
DATE(9999,12,31)+TIME(23,59,59) results in the serial value 2958465.9999884…

For the 1904 date base system:

DATE(1893,8,5)+TIME(0,0,1) results in the serial value -3799.999989…
DATE(1910,2,3)+TIME(10,5,54) results in the serial value 2225.4207639…
DATE(1904,1,1)+TIME(12,0,0) results in the serial value 0.5000000…
DATE(9999,12,31)+TIME(23,59,59) results in the serial value 2957003.9999884…

end example]

Changes to Part 4, §3.17.6.7, page 2,529, line 27 through page 2,530, line 5:

3.17.6.7 Dates and Times

~~As a~~A date and/or time is represented by an ISO 8601 string. ~~number, a date/time serial value shall be stored in XML as the unformatted text form of that number, as accurately as possible.~~

The date-base system is recorded in the Workbook part's XML by the dateCompatibility and ~~presence or absence of the~~ date1904 attributes of the workbookPr element. ~~A value of 1 for this attribute indicates 1904.~~
[*Example*:

1900 date-base: <workbookPr dateCompatibility="false" showObjects="all"/>
1904 backwards compatibility date-base: <workbookPr dateCompatibility="true" date1904="true"~~"1"~~ showObjects="all"/>

end example]

Part 4, §3.17.7.74, page 2,600, line 11:

3.17.7.74 DATE

...

Name	Type	Description
Year	Number	A ~~year,~~positive number, truncated to an integer representing the year, that together with *month* and *day* specifies the date whose serial value is to be computed. For the 1900 date base and 1904 date base systems: • If *year* is in the range 0–~~18~~99, inclusive, the year shall be interpreted as *year* + 1900. • If *year* is in the range ~~1900~~100–9999, inclusive, the year shall be interpreted as *year*. For the-1900 backward compatibility date-base and 1904 backward compatibility date-base systems: • If *year* is in the range ~~4~~0–1899, inclusive, the year shall be interpreted as *year* + 1900. • If *year* is in the range 1900~~4~~–9999, inclusive, the year shall be interpreted as *year*.

Part 4, §3.17.7.74, page 2,601, lines 10–14:

However, if *year* is outside the acceptable range for the date base currently in use, #NUM! is returned.

- ~~*year* is less than 0 or is greater than or equal to 10000, and the 1900 date base system is being used, #NUM! is returned.~~
- ~~*year* is less than 4, is greater than or equal to 10000, is in the range 1900–1903, inclusive, and the 1904 date base system is being used, #NUM! is returned.~~

Part 4, §3.17.7.347, page 2,825, lines 17–19:

Return Type and Value: number – The Gregorian year for the date and/or time having the given *date-value*. ~~For the 1900 date base system, the returned value shall be in the range 1900–9999. For the 1904 date base system, the returned value shall be in the range 1904–9999.~~The range of return values is determined by the date base currently in use (§3.17.4).

Changes to Part 4, §3.18.12, page 2,840, line 19 through page 2,841, line 0:

3.18.12 ST_CellType (Cell Type)

Indicates the cell's data type.

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

Enumeration Value	Description
b (Boolean)	Cell containing a boolean.
d (Date)	Cell contains a date in the ISO 8601 format
e (Error)	Cell containing an error.
inlineStr (Inline String)	Cell containing an (inline) rich string, i.e., one not in the shared string table. If this cell type is used, then the cell value is in the is element rather than the v element in the cell (c element).
n (Number)	Cell containing a number.
s (Shared String)	Cell containing a shared string.
str (String)	Cell containing a formula string.