

Características de migración de OOXML Transitional

Referencia técnica

Documento complementario a «Un estándar solo de nombre: lo que OOXML Transitional nos revela sobre la soberanía de los formatos».

Todos los nombres de elementos y las referencias a secciones de este documento se han verificado con conformidad con las normas ISO/IEC 29500-1:2016 e ISO/IEC 29500-4:2016.

Introducción

La norma ISO/IEC 29500 -el estándar internacional en el que se basan los formatos de archivo docx, xlsx y pptx de Microsoft- se publica en cuatro partes, con un total de varios miles de páginas. **La Parte 1, Fundamentos y referencia del lenguaje de marcado**, contiene la definición de conformidad Strict: el formato de documento limpio y moderno que cabría esperar razonablemente que admitiera un implementador independiente. **La Parte 4, Características de migración Transitional**, contiene el resto: un catálogo de elementos de compatibilidad, construcciones obsoletas, comportamientos específicos de la plataforma y referencias a peculiaridades no documentadas de las versiones de Microsoft Office de la década de 1990.

La clase Transitional existe para garantizar que los documentos convertidos desde los antiguos formatos binarios doc, xls y ppt -miles de millones de archivos acumulados a lo largo de dos décadas de dominio del mercado por parte de Microsoft Office- puedan representarse en XML sin pérdidas. En la práctica, Transitional es también lo que produce el propio Microsoft Office: Strict nunca ha sido el valor predeterminado en ninguna versión de Microsoft Office, y la opción de guardar en formato Strict no está disponible de manera uniforme en todas las ediciones de Microsoft 365 -está presente en las aplicaciones de escritorio instaladas, pero ausente en las versiones basadas en navegador-.

Este documento cataloga las principales categorías de características de Transitional y explica los problemas estructurales que plantean para cualquier parte que intente implementar, validar o confiar en OOXML como un estándar abierto.

Cómo describe el estándar sus propias características de compatibilidad

Antes de enumerar las características de Transitional, vale la pena señalar cómo las describe la Parte 4 en sus propias palabras. El texto introductorio de la sección sobre

ajustes de compatibilidad (§14.8.3) afirma que “varios de los ajustes hacen referencia a aplicaciones específicas y a versiones concretas de dichas aplicaciones” y que “esto se debe exclusivamente a motivos de compatibilidad con versiones anteriores”. La especificación es sincera: gran parte de la Parte 4 no son tanto especificaciones como referencias al comportamiento de productos comerciales identificados.

Lo más llamativo es que la norma califica repetidamente de incorrecto el comportamiento que ella misma exige. La sección 14.8.3.5 (`cachedColBalance`) comienza con la frase “las aplicaciones calcularán incorrectamente la altura de un párrafo” y añade la siguiente orientación editorial: “*Se recomienda que las aplicaciones no reproduzcan intencionalmente este comportamiento; se mantiene únicamente por motivos de compatibilidad con documentos existentes procedentes de una aplicación heredada*”. La sección 14.8.3.55 (`useWord97LineBreakRules`) precede su especificación con la admisión de que los cálculos que impone “normalmente no se considerarían correctos”. La sección 14.8.3.54 se titula `useWord2002TableStyleRules` con el subtítulo descriptivo “Mostrar incorrectamente el borde superior de las columnas condicionales”. La sección 14.8.3.4 (`autoSpaceLikeWord95`) se titula “Ajuste incorrecto del espaciado del texto para rangos Unicode específicos”.

Una norma que describe abiertamente su propio comportamiento obligatorio como incorrecto, y que aconseja a los implementadores que no reproduzcan partes de sí misma, se encuentra en una posición curiosa. Estas son las palabras exactas de la norma ISO/IEC 29500-4.

Categorías de características de Transitional

Emulación de aplicaciones heredadas

Existe un conjunto considerable de elementos de Transitional que indican a las aplicaciones modernas que emulen el comportamiento de versiones antiguas específicas de Microsoft Word, MacWord o WordPerfect. Estos elementos aparecen dentro del bloque `<w:compat>` en el archivo `settings.xml`. Entre los ejemplos verificados del apartado §14.8.3 se incluyen:

- `<w:autoSpaceLikeWord95>` (§14.8.3.4) - espacia la puntuación como lo hacía Word 95; el título de la sección caracteriza este comportamiento como un ajuste “incorrecto” del espaciado del texto
- `<w:footnoteLayoutLikeWW8>` (§14.8.3.20) - el nombre del elemento hace referencia al comportamiento de las notas al pie de Word 97/2000 («WW8»), aunque el título descriptivo revela que el comportamiento real exigido se refiere a los saltos de sección en presencia de notas al pie

- `<w:lineWrapLikeWord6>` (§14.8.3.25) - ajusta las líneas utilizando el algoritmo de Word 6.0
- `<w:mwSmallCaps>` (§14.8.3.26) - utilizar un algoritmo específico para las versalitas (la sección define una secuencia numérica opaca cuya única justificación es que coincida con el resultado de una aplicación heredada concreta)
- `<w:shapeLayoutLikeW8>` (§14.8.3.35) - Ajustar la forma de los elementos utilizando la lógica de Word 2000
- `<w:splitPgBreakAndParaMark>` (§14.8.3.38) - gestionar los saltos de página de una forma tradicional específica
- `<w:suppressTopSpacingWP>` (§14.8.3.43) - aplicar un cálculo estático de la distancia entre líneas de referencia heredado de WordPerfect
- `<w:truncateFontHeightsLikeWP6>` (§14.8.3.46) - utilizar la división entera truncada para la conversión de la altura de la fuente, imitando a WordPerfect 6.0
- `<w:useWord97LineBreakRules>` (§14.8.3.55) - aplicar reglas de espaciado entre caracteres cuyo propio texto de especificación admite que «normalmente no se considerarían correctas»
- `<w:useWord2002TableStyleRules>` (§14.8.3.54) - muestran incorrectamente el borde superior de las columnas condicionales
- `<w:wpJustification>` (§14.8.3.56) - Justificar el texto utilizando un cálculo de ancho específico derivado de WordPerfect que implica el número mágico 281/7200

La especificación de estos elementos se basa en gran medida en referencias: «emular el comportamiento de la aplicación X, versión Y». El comportamiento de esas aplicaciones nunca se publicó. Por lo tanto, un desarrollador independiente debe realizar ingeniería inversa en el software de la década de 1990 -incluidos sus errores y casos extremos- para poder representar correctamente los documentos conformes.

Vector Markup Language (VML)

Microsoft presentó VML al W3C en mayo de 1998 como propuesta de estándar para gráficos vectoriales en la web. El W3C lo evaluó junto con la propuesta competidora PGML, presentada por Adobe y Sun, y finalmente no eligió ninguna de las dos: en su lugar, se desarrolló SVG, que alcanzó el estatus de Recomendación en 2001. Por lo tanto, VML no fue un formato «preestándar» en ningún sentido neutral: fue una propuesta que el organismo de normalización pertinente rechazó.

A pesar de ello, VML persiste dentro de OOXML Transitional porque los documentos convertidos desde archivos .doc contienen marcado VML, y Microsoft Office sigue

generándolo en ciertos contextos. Las implementaciones deben ser compatibles tanto con VML como con su sustituto moderno, DrawingML, para manejar archivos del mundo real. Entre los elementos VML representativos se incluyen `<v:shape>`, `<v:rect>`, `<v:oval>`, `<v:line>`, `<v:textbox>` y `<v:imagedata>`. La sección 14.8.4.1 de la Parte 4 (relyOnVML) especifica con mayor detalle cómo se debe conservar la salida VML cuando el contenido de WordprocessingML se guarda como una página web.

Elementos específicos de la plataforma

Varias características de transición dan por sentadas tecnologías específicas de la plataforma de Microsoft. El ejemplo más claro es `<w:usePrinterMetrics>` (§14.8.3.52), que indica a las aplicaciones que utilicen métricas específicas de la impresora para maquetar los documentos. La nota no normativa que acompaña a esta sección ilustra la implementación haciendo referencia directa a la API de Windows: *“En la plataforma Windows, puede utilizar la función GetDeviceCaps para recuperar información específica del dispositivo para la impresora especificada”*. El estándar no afirma que GetDeviceCaps sea obligatorio, pero no ofrece ninguna guía de implementación para ninguna otra plataforma, y el algoritmo que describe asume un contexto de dispositivo GDI de Windows.

Otras características vinculadas a la plataforma incluyen la incrustación de controles ActiveX (diseñada exclusivamente para Windows), la incrustación de objetos OLE a través del Modelo de Objetos de Componentes y el procesamiento XSLT que históricamente asumía el intérprete XML de Microsoft (MSXML).

Comportamientos considerados incorrectos

Existe una categoría específica de elementos de transición que no emula una aplicación heredada concreta, sino que impone cálculos o comportamientos de representación que la propia norma describe como incorrectos. Los ejemplos más claros son:

- `<w:cachedCollBAlance>` (§14.8.3.5) - establece que las aplicaciones “deberán calcular incorrectamente la altura de un párrafo” para equilibrar las columnas
- `<w:useWord97LineBreakRules>` (§14.8.3.55) - impone cálculos de espaciado entre caracteres que, según reconoce la especificación, “normalmente no se considerarían correctos”
- `<w:useWord2002TableStyleRules>` (§14.8.3.54) - titulado “Visualización incorrecta del borde superior de las columnas condicionales”
- `<w:convMailMergeEsc>` (§14.8.3.6) - establece que los delimitadores de comillas invertidas se conviertan automáticamente en comillas dobles durante la

combinación de correspondencia, una transformación que modifica los datos que se están combinando

No se trata de características de compatibilidad con formatos heredados en un sentido neutral. Son cálculos que el estándar exige que las implementaciones realicen, al tiempo que señala que dichos cálculos son erróneos.

Propiedades de documento y características de plataforma obsoletas

Más allá de la configuración de compatibilidad, OOXML Transitional conserva propiedades heredadas de formatos anteriores; por ejemplo, la configuración de documentos relacionada con las etiquetas inteligentes (una función de Office que ya no se utiliza), la incrustación de fuentes específica de Windows, los supuestos de procesamiento XSLT y el sistema de fechas con el año bisiesto de 1900 que se analiza a continuación.

El año bisiesto de 1900 y la arquitectura de un error impuesto

El error del año bisiesto de 1900 es el defecto más mencionado en OOXML Transitional, pero su tratamiento en el estándar resulta más interesante de lo que suele indicarse. La Parte 1, en §18.17.4.1, define dos sistemas de fechas para SpreadsheetML:

“En el sistema de fechas 1900, el límite inferior es el 1 de enero del año 0001 a las 00:00:00, lo que corresponde a una fecha y hora serial de -693593. El límite superior es el 31 de diciembre de 9999 a las 23:59:59.999, que tiene una fecha y hora serial de 2,958,465.9999884. La fecha base de este sistema es las 00:00:00 del 30 de diciembre de 1899, que tiene una fecha y hora serial de 0.”

La fecha de referencia del sistema de fechas de 1900 es el **30 de diciembre de 1899**, y no, como podría sugerir el nombre del sistema, el 1 de enero de 1900. La razón es el error del año bisiesto: para mantener la compatibilidad con Lotus 1-2-3, que había asumido que 1900 era un año bisiesto, la aritmética de la fecha serial debe incluir un 29 de febrero de 1900 fantasma. Retrasar la fecha base dos días da cabida a este día fantasma, al tiempo que mantiene la fecha y hora serial de, por ejemplo, el 1 de enero de 1900 en el valor entero esperado.

Lo que llama la atención es que la norma no establece en ningún momento de manera explícita que 1900 deba considerarse un año bisiesto. El error está codificado estructuralmente, a través de la elección de la fecha base y los ejemplos prácticos que la siguen. Un lector del apartado §18.17.4.1 que no esté familiarizado con la historia encontrará un conjunto de definiciones del sistema de fechas que parece coherente, sin ninguna anomalía señalada. El error está presente en la aritmética sin que se mencione

expresamente.

El sistema de fechas de 1904, definido junto con el sistema de 1900 en la misma sección, tiene como fecha base el 1 de enero de 1904 -tal y como su nombre indica- porque a ese sistema no se le aplicaba ninguna restricción de compatibilidad con Lotus 1-2-3. La asimetría es prueba de ello.

La consecuencia práctica es que cada archivo .xlsx creado con la configuración predeterminada codifica el error aritmético de Lotus 1-2-3 de la década de 1980. Los cálculos de fechas que abarcan febrero de 1900 producen resultados incorrectos. La norma exige este comportamiento y nunca lo reconoce como un defecto.

Problemas estructurales

Contaminación de espacios de nombres

Las características de la versión Transitional comparten espacios de nombres XML con las características que cumplen con el modo Strict. No existe una separación sintáctica clara. Para validar si un documento determinado cumple realmente con el modo Strict -en contraposición a la versión Transitional en la que, por casualidad, no hay elementos Transitional presentes- es necesario recorrer todo el árbol del documento y comparar cada elemento con el subconjunto Strict.

Especificación basada en comportamientos no documentados

Muchos elementos de transición se especifican en términos como “diseño similar al de Word 95” o “cálculo del espaciado como lo hacía WordPerfect”. No se trata de especificaciones formales, sino de referencias al comportamiento observable de productos comerciales cuyo código fuente, motores de diseño y algoritmos nunca se han publicado. Cualquier desarrollador que intente implementar estos elementos debe basarse en ingeniería inversa, documentación filtrada o en un proceso de prueba y error comparando con las aplicaciones originales.

El resultado es que las partes del estándar con más probabilidades de aparecer en documentos del mundo real son también las menos susceptibles de ser implementadas de forma independiente.

Obsolescencia sin eliminación

Los elementos marcados como obsoletos o heredados permanecen en la especificación de forma indefinida. No existe un calendario previsto para su obsolescencia, ni una versión del estándar en la que se vaya a retirar la clase Transitional, ni ningún mecanismo mediante el cual el estándar pueda converger eventualmente en el subconjunto Strict. Dado que Microsoft Office sigue generando resultados en formato Transitional, la eliminación de estos elementos del estándar invalidaría los documentos

que el principal productor de archivos OOXML sigue generando.

Autorreconocido como incorrecto

Hay un subconjunto de elementos de transición que son obligatorios, aunque la propia norma describe su comportamiento como incorrecto. En el caso de `<w:cachedCollBalance>` (§14.8.3.5), la especificación recomienda explícitamente que las implementaciones no reproduzcan ese comportamiento: *“Se recomienda que las aplicaciones no reproduzcan intencionalmente este comportamiento; se mantiene únicamente por motivos de compatibilidad con documentos existentes de una aplicación heredada”*. Un estándar internacional que desaconseja seguir partes de sí mismo se encuentra en una situación inusual.

El problema de la clase de compatibilidad

La norma define dos clases de compatibilidad con estatus prácticos muy diferentes.

La **compatibilidad Strict** excluye las características de transición enumeradas anteriormente. Se trata de la versión “limpia” prevista de OOXML. En la práctica, Strict nunca ha sido el formato de guardado predeterminado en ninguna versión de Microsoft Office, el soporte de lectura/escritura llegó solo con Office 2013, y la opción de guardar en formato Strict no está disponible de manera consistente en todas las ediciones de Microsoft 365: está presente en las aplicaciones de escritorio instaladas, pero ausente en las versiones basadas en navegador. La inconsistencia entre ediciones es un patrón de larga data en la historia de Microsoft Office: la versión para macOS ha ofrecido históricamente un conjunto de opciones diferente al de la versión para Windows. Por lo tanto, la compatibilidad Strict existe principalmente como un artefacto de especificación: una definición que ningún fabricante importante genera de forma predeterminada y que la propia Microsoft admite de manera desigual en los productos que vende bajo el mismo nombre.

La **compatibilidad Transitional** incluye todos los elementos heredados. Es lo que Microsoft Office produce realmente, lo que los usuarios reciben realmente y lo que cualquier implementación de la competencia debe admitir para ser útil en la práctica. Desde el punto de vista de un implementador, la compatibilidad Transitional es el estándar.

La consecuencia práctica es que las partes de OOXML que existen sobre el papel como un “estándar moderno y limpio” no son las que se utilizan en los documentos reales, y las partes que se utilizan en los documentos reales son precisamente las que están peor especificadas, las que más dependen de la plataforma y las que más se basan en el conocimiento del historial de implementación de un único proveedor.

Ejemplos ilustrativos

<w:footnoteLayoutLikeWW8> (Word)

```
<w:compat>
  <w:footnoteLayoutLikeWW8/>
</w:compat>
```

El nombre del elemento hace referencia al comportamiento de diseño de Word 97/2000 (siendo «WW8» el identificador interno de la versión de Word 97). Los algoritmos reales utilizados por esas versiones de Word -incluidos los casos extremos, el comportamiento de los saltos de línea y las interacciones con la geometría de la página- nunca se documentaron formalmente. Para cumplir con los requisitos, el desarrollador debe realizar ingeniería inversa en Word 97.

El sistema de fecha 1900 (Excel)

```
<workbookPr date1904="false"/>
```

Cuando `date1904="false"` (el valor predeterminado), las implementaciones deben tratar las fechas de acuerdo con el sistema de fechas de 1900 definido en la Parte 1, §18.17.4.1. Ese sistema tiene como fecha base el 30 de diciembre de 1899, una codificación estructural del 29 de febrero de 1900 “fantasma” heredada de Lotus 1-2-3. Las hojas de cálculo que realicen operaciones aritméticas con fechas que abarquen febrero de 1900 producirán resultados incorrectos, y la norma exige este comportamiento incorrecto sin reconocerlo explícitamente como un defecto.

VML en documentos convertidos

```
<w:pict>
  <v:shapetype id="_x0000_t75" coordsize="21600,21600"
    o:spt="75" o:preferrelative="t"
    path="m@4@5l@4@11@9@11@9@5xe" filled="f" stroked="f">
    <v:stroke jointstyle="miter"/>
    <v:path gradientshapeok="t" o:connecttype="rect"/>
  </v:shapetype>
</w:pict>
```

El marcado VML de este tipo aparece en documentos convertidos desde el formato .doc. Las implementaciones que solo sean compatibles con el sistema gráfico moderno DrawingML no podrán mostrar dicho contenido.

<w:cachedColBalance> - un error de cálculo impuesto

```
<w:compat>
  <w:cachedColBalance/>
</w:compat>
```

La primera frase del apartado 14.8.3.5 establece que este elemento “especifica si las aplicaciones deben calcular incorrectamente la altura de un párrafo a efectos del equilibrio de columnas”. Las directrices que lo acompañan recomiendan que las aplicaciones no reproduzcan intencionalmente ese comportamiento. Por lo tanto, se indica al desarrollador que implemente un cálculo incorrecto, a sabiendas de que es incorrecto, con el fin de mantener la compatibilidad con los documentos generados por una aplicación heredada.

Implicaciones para la interoperabilidad

Las características de Transitional plantean cuatro problemas de implementación distintos.

Complejidad de implementación. Cualquier tercero que desee leer o escribir archivos OOXML en su forma real debe implementar no solo el subconjunto Strict moderno, sino también el ámbito de aplicación Transitional, mucho más amplio, que incluye elementos especificados por referencia a comportamientos heredados no documentados y elementos cuyo comportamiento requerido se describe como incorrecto.

Complejidad para pruebas. La validación de la corrección requiere comparar la salida con el comportamiento de Microsoft Office, incluido el comportamiento de versiones de Office descatalogadas hace mucho tiempo a las que se hace referencia en la norma. Ningún conjunto de pruebas formal puede capturar la totalidad de “lo que hacía Word 97”.

Fragmentación del estándar. Los documentos producidos por Microsoft Office no cumplen con la compatibilidad Strict. Por lo tanto, el valor de la compatibilidad Strict para los responsables de adquisiciones, los archivistas o los responsables políticos que especifican “ISO/IEC 29500” es limitado: la clase de compatibilidad que pueden tener en mente no es la que utilizan los documentos reales.

Legado indefinido. Las características de Transitional no tienen un plazo de obsolescencia. Permanecen en el estándar mientras Microsoft Office siga emitiéndolas, es decir, indefinidamente.

Conclusión

Las características de migración transitoria reflejan la decisión de diseño fundamental de OOXML como norma internacional: en lugar de especificar un formato de documento limpio y neutral desde el punto de vista de la implementación, la norma formaliza el comportamiento existente de la línea de productos de una empresa, incluyendo treinta años de decisiones de compatibilidad acumuladas, supuestos sobre plataformas y errores heredados -algunos de los cuales la propia norma reconoce como incorrectos y aconseja a los implementadores que no los reproduzcan.

Esta es la base de la crítica que desde hace tiempo sostiene que OOXML es un “estándar” solo de nombre. La certificación ISO es real. La estructura de conformidad de dos partes es real. Pero el estándar con el que se encuentran realmente la mayoría de los usuarios -y el que Microsoft Office produce de manera predeterminada y soporta de manera más consistente en toda su línea de productos- es el que perpetúa comportamientos heredados, depende de propiedades no documentadas de aplicaciones de la década de 1990, codifica estructuralmente errores aritméticos heredados de la década de 1980 y no puede ser implementado de manera limpia por ninguna parte sin acceso privilegiado al conocimiento institucional de Microsoft.

Para las administraciones públicas, los archivistas y los responsables de adquisiciones que evalúan la política de formatos, la implicación es directa: especificar OOXML en una licitación no ofrece la interoperabilidad independiente del proveedor que un estándar abierto debe proporcionar. El formato OpenDocument (ODF, ISO/IEC 26300) se diseñó desde el principio para evitar estos problemas y sigue siendo el único estándar de documentos ampliamente implementado que cumple con la definición sustantiva -y no meramente formal- de apertura.