

[Example: Consider a hypothetical application, ChrisOffice 2011, which allows sound effects to be applied to SpreadsheetML conditional formatting. The implementers of ChrisOffice 2011 decide to store the data inside application-defined extension elements because they would like other Office applications to be able to round-trip that data, and in SpreadsheetML an application-defined extension element (**extLst**) is already defined under the **conditionalFormattingElements** element. An example of this element with the additional sound effect added might look like this:

```
<conditionalFormattingElements>
  <extLst>
    <ext uri="myurl" xmlns:col1="http://chrisoffice/2011">
      <col1:soundeffect>
        <col1:sourceFile>moo.mp3</col1:sourceFile>
      </col1:soundeffect>
    </ext>
  </extLst>
</conditionalFormattingElements>
```

Understood by  
ChrisOffice 2011 but  
round-tripped by  
other IS 29500  
compliant  
applications

Now we add the ability to use video as well as MP3 in ChrisOffice 2013. We want to allow ChrisOffice 2011 to understand everything except the video, but we'll have to use a different namespace to avoid ChrisOffice discovering unknown content in understood namespaces. Because ChrisOffice 2011 doesn't have its own extension elements inside its extension elements, our only option is to make the extra content ignorable, which will mean that ChrisOffice 2011 discards it upon load.

Inside the file, this might look like:

```
<conditionalFormattingElements>
  <extLst>
    <ext uri="myurl" xmlns:col1="http://chrisoffice/2011">
      <col1:soundeffect mc:Ignorable="col3"
xmlns:col3="http://chrisoffice/2013">
        <col1:sourceFile>moo.mp3</col1:sourceFile>
        <col3:sourceVideo>cow.mpg</col3:sourceFile>
      </col1:soundeffect>
    </ext>
  </extLst>
</conditionalFormattingElements>
```

Understood by  
ChrisOffice 2013,  
discarded by  
ChrisOffice 2011 but  
round-tripped by  
other IS 29500  
compliant  
applications

ChrisOffice 2011 will discard the video segment when it reads the file, which is our desired behaviour. Because processing of MCE constructs is not permitted inside application-defined extension elements (*insert relevant reference*), applications which do not understand the original sound effect construct will not needlessly throw away the new video content. *end example]*