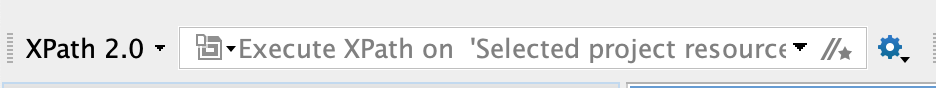
功能一、Xpath表达式

1、定义：Xpath是一种处理文档特定部分的语言，可用于在文档中导航和选择节点，Xpath定义了一个标准函数库，用于处理字符串、数字和布尔表达式。

2、Xpath工具栏： Oxygen XML编辑器提供了Xpath工具栏（如下图所示），使用户可以使用Xpath表达式快速方便地查询XML文档。



用户可以选择Xpath的版本：

图片包含 屏幕截图

描述已自动生成

（1）Xpath范围菜单：Oxygen XML编辑器允许用户选择执行Xpath操作的范围， Xpath表达式的可选位置包括当前文档、项目等。

（2）历史记录：Xpath组合框会保留最后使用的15个表达式的记录，从而用户可以进行快速选择。

（3）设置菜单（如下图所示）

图片包含 屏幕截图

描述已自动生成

①光标移动时更新：显示与当前光标位置处的XML节点对应的Xpath表达式。对于JSON文档，它显示当前属性的Xpath表达式。

②在键入时计算Xpath：实时计算正在编写的Xpath表达式。

③Xpath选项：打开当前选定处理引擎的首选项页（如下图所示）

图片包含 屏幕截图

描述已自动生成

④切换到Xpath生成器视图。

（4）停止使用： Xpath工具栏上有停止按钮，点击即可停止执行Xpath。

3、Xpath生成器视图：允许用户组合使用Xpath表达式。（视图界面如下图所示）

图片包含 屏幕截图

描述已自动生成

（1）执行表达式：点击执行。

（2）收藏设置：用于保存某些表达式以便重用。点击即可将表达式放入收藏夹，图形变为黄色时确认表达式已保存。展开星形按钮旁边的下拉菜单可查看所有收藏夹。Oxygen XML编辑器会自动将收藏夹分组到以执行方法命名的文件夹中。

图片包含 屏幕截图

描述已自动生成

（3）历史记录和设置菜单功能同上工具栏。

图片包含 屏幕截图

描述已自动生成

（4）辅助功能

①内容完成助手：基于用户当前的光标位置，提供与上下文相关的建议。此外，其提出的功能集也取决于引擎版本，用户可以在工具栏中的下拉菜单中选择引擎版本。

②语法突出显示：允许用户标识表达式的组件。如果用户要自定义表达式组件的颜色，点击“首选项”对话框（选项>首选项），再点击编辑器>语法突出显示。

③自动验证用户输入的表达式。

④其余的一些常见功能，比如复制、剪切、回退等操作，都在内容菜单中。

4、Xpath结果视图



包含属性如下：

（1）描述： Oxygen xml编辑器显示表达式的结果。

（2）Xpath位置：匹配节点的路径。

（3）资源：运行Xpath表达式的文档的名称。

（4）系统ID：文档本身的路径。

（5）位置：结果在文档中的位置。

具有功能如下：

（1）在编辑器中突出显示所有结果，单击“在编辑器中突出显示所有结果”即可关闭突出显示。如果用户想要自定义突出显示行为，点击“首选项”对话框（选项>首选项），再点击编辑器>突出显示类别（自定义设置包括设置高亮颜色、设置在给定时间中高亮显示的最大数量等）

图片包含 屏幕截图

描述已自动生成

（2）删除选定内容：从视图中删除当前选择。可用于删除已寻址或与用户任务无关的内容。

（3）删除全部：从视图中删除所有内容。

其余工具栏和菜单信息同上生成器视图。

功能二、插件

Oxygen XML编辑器有各种插件，提供了许多的附加功能。

下载插件：点击“帮助>安装新加载项”，打开加载项选择对话框，输入https://www.oxygenxml.com/InstData/Addons/default/updateSite.xml，选中所需附件，

然后单击“下一步”，最后重新启动应用程序。

图片包含 屏幕截图

描述已自动生成

图片包含 屏幕截图

描述已自动生成

以下将以Oxygen中常见的四个插件为例进行介绍。

1、批处理转换器插件

该插件可用于转换以下格式：

• HTML to XHTML

• HTML to DITA

• Markdown to XHTML

• Markdown to DITA

• XML to JSON

• JSON to XML

• HTML to DocBook4

• HTML to DocBook5

• Markdown to DocBook4

• Markdown to DocBook5

• Excel to DITA

按照上述下载方式下载该插件，重启软件后在工具菜单和项目视图的山下文菜单中会出现该插件，其中包含各种转换类型的列表，选择某个转换类型，会弹出一个对话框，用于可以在其中设置转换选项。

2、内容融合插件

该插件是一个非常灵活且直观的协作平台，几乎能够适应所有可能被用到的类型的工作流。单击“显示内容融合任务管理器”工具栏按钮，即打开“内容融合任务管理器”视图，用户在其中可以创建和管理任务、附加文档，并将它们上传到内容融合服务器。

此外，用户还可以托管和配置自己的服务器并控制数据，以及上载自定义框架和插件，并配置各种设置，还可以在有限的时间内免费评估内容融合。

3、DITA Prolog更新插件

该插件提供了首选项页面（选项>首选项>Plugins>DITA Prolog Updater），可用于更新DITA主题或映射的prolog部分的各种选项。

首选项页包含以下选项：

（1）作者。指定作者的名字，默认为系统用户名。

（2）日期格式。指定添加到prolog中的日期的格式。如果输入的日期格式无效，则默认使用yyy/MM/dd格式。

（3）保存时自动更新prolog。在保存文档时更新prolog。

（4）设置创建者名称。保存文档时在prolog中将作者设置为文档的创建者，此选项仅适用于新文档。

（5）设置创建日期。保存文档时会将创建日期添加到prolog中。此选项仅适用于新文档。

（6）更新参与者名称。保存文档时作者将被设置为prolog中的参与者。此选项仅适用于已经存在的文档。

（7）更新修订日期。保存文档时将在prolog中添加修订日期。此选项仅适用于已经存在的文档。

4、PDF图像显示插件

从Oxygen XML 20.0版本开始，PDF图像以Author模式和PDF格式输出，所以对于以前的版本中的图像，默认不会显示。但是，该插件提供了Apache PDFBox库，来支持PDF图像的显示。

在插件安装完毕后，Apache PDFBox库会出现在Oxygen安装目录中，用户即可在Author模式下看到PDF图像。

功能三、实用工具

Oxygen XML编辑器涵盖了各种实用工具，帮助用户完成与XML相关的任务。点击“工具”菜单即可查看这些工具，其中部分工具还可以通过键盘快捷键或命令行脚本来启动。

图片包含 屏幕截图

描述已自动生成

以下将以Oxygen中常见的四个实用工具为例进行介绍。

1、生成示例XML文件工具

该工具包含一个对话框，允许用户配置用于生成XML文件的各种选项，比如设定用户名称空间和元素。

图片包含 屏幕截图

描述已自动生成2、生成/转换架构工具

该工具用于转换架构语言，允许用户将DTD或Relax NG（完整或紧凑语法）模式或一组XML文件转换为等效的XML模式、DTD或Relax NG（完整或紧凑语法）模式。由于目标语言限制，难以实现完全等价，只能生成近似值。

用户如果想要使用此工具，点击“工具”菜单进行选择，或使用键盘快捷键（WIN：Alt+Shift+C或MAC：Command+Alt+C）

图片包含 屏幕截图

描述已自动生成

如上图所示，对话框包含：

（1）输入。选择源架构的语言。如果转换基于一组而不是一个XML文件，需要选择XML文档选项，并使用文件选择器添加转换中涉及的XML文件。

（2）输出。选择目标架构的语言。用户可以选择编码、最大线宽和缩进、一级缩进的大小，以及指定输出文件的路径。

（3）完成后关闭对话框。如果取消选择该选项，则转换后对话框将保持打开状态，方便用户继续转换其他文件。

（4）高级选项。如果选择XML 1.0 DTD作为输入，则可以单击此按钮访问更多高级选项，以进一步微调转换。

图片包含 屏幕截图

描述已自动生成

高级选项如下：

①XML 1.0 DTD输入部分

•属性定义列表：指定如何根据元素的名称，来构造代表属性列表声明定义的名称。

•冒号替换：用于表示DTD中的元素声明和属性列表声明定义的名称时指定的字符。

•任意名称：指定为DTD中声明元素内容具有任意模型而生成的定义的名称。

•元素定义：指定如何根据元素的名称构造表示元素声明的定义的名称。

•注释前缀：使用注释表示默认值。默认情况下，转换引擎将使用for前缀，除非该前缀与DTD中使用的前缀冲突。

•串联式属性列表：指示应用程序不生成属性列表声明的定义，而是将属性列表声明中声明的属性移动到为元素声明生成的定义中。这是输出语言为XSD时的默认操作。

•全部严格：通过使用对所有声明元素的显式引用选择，保留所有内容模型的精确语义。默认情况下，转换引擎会采用允许所有元素被使用的通配符。

•生成start元素：指定转换引擎是否应生成start元素。

•xmlns映射表：每行指定输入架构中的命名空间。

②W3CXML模式输出部分

•禁用抽象元素。禁止在生成的XML模式中使用抽象元素和替换组，也可以通过使用注释属性进行控制。

•任何过程内容：指定任何元素的processContents属性的值。

•任何属性进程内容：指定anyAttribute元素的processContents属性，默认值是

skip。

3、格式化和缩进

Oxygen xml编辑器支持同时格式化和缩进多个文件,此操作可用于XML格式的任何文档，也可用于XQuery、CSS、JavaScript和JSON文档。

图片包含 屏幕截图

描述已自动生成

（1）作用域部分包括以下内容：

•所有打开的文件：美观打印。

•当前文件的目录：当前编辑文件文件夹中的所有文件。

•项目文件：当前项目中的所有文件。

•选定项目文件：当前项目中选定的文件。

•指定路径：在指定的路径。

（2）选项部分包括以下选项：

•文件筛选：允许用户从选定范围筛选文件。

•递归子目录：对指定范围递归执行美观打印，但是如果作用域设置为All，此选项会被忽略。

•包括隐藏文件：如果选中，隐藏文件也会被打印出来。

•生成带有扩展名的备份文件：Oxygen XML编辑器将生成修改过的文件，默认扩展名为.bak，但用户可以根据需要更改扩展名。

4、签名验证

该工具可以验证文件的签名是否有效。点击“工具”菜单选择“签名验证”，弹出的“验证签名”对话框允许用户指定其签名已验证的文件的位置。如果签名有效，则会出现一个对话框，显示签名者的姓名；否则将提出错误，并显示有关该问题的详细信息。

图片包含 屏幕截图

描述已自动生成

专题一、自定义框架和作者模式

情景设置：某个团队采用Oxygen进行文档开发，Oxygen含有用于DocBook、DITA、TEI、XHTML和jat的内置配置框架，但是该团队想要创建专属的自定义架构来处理其他类型的文档，即自定义接口以适应创作团队的需要。

完全配置框架通常需要自定义CSS样式表、XML模式、GUI组件（菜单操作、工具栏、内联组件、内容完成建议等），再配置其他更通用的设置，然后将框架捆绑到团队中共享。CSS和GUI组件用于自定义界面，而其他常规设置可以配置为适应自定义文档模板、XML目录、转换方案等。  
 可以通过扩展捆绑包和JAVA接口这两种方式来自定义框架和作者模式。

1、配置扩展捆绑包

团队可以将扩展添加到自定义框架，使用“文档类型配置”对话框（第95页）中的“扩展名”选项卡。

配置扩展捆绑包的流程如下：

（1）在IDE中创建一个新的Java项目。在Java项目文件夹中创建一个lib文件夹并在其中复制[oxygen\_INSTALL\_DIR]/lib文件夹中的oxygen.jar文件。

（2）创建类（例如simple.documentation.framework.SDFExtensionsBundle）以扩展抽象类ro.sync.ecss.extensions.api.ExtensionsBundle。

举例如下：

public class SDFExtensionsBundle extends ExtensionsBundle {

（3）首先实现方法getDocumentTypeID和getDescription。文档类型ID用于唯一标识当前框架，且是必须的，特别是需要在会话间长久存储和检索时。

举例如下：

public String getDocumentTypeID() {

return "Simple.Document.Framework.document.type";

}

public String getDescription() {

return "A custom extensions bundle used for the Simple Document" +

"Framework document type";

}

（4）自定义作者扩展状态侦听器，通过createAuthorExtensionStateListener方法来实现。

举例如下：

public AuthorExtensionStateListener createAuthorExtensionStateListener() {

return new SDFAuthorExtensionStateListener();

}

（5）声明用于内容完成建议筛选的方法的接口是ro.sync.content completion.xml.schemaManager筛选器。createSchemaManager筛选器方法负责创建内容完成筛选器。

举例如下：

public SchemaManagerFilter createSchemaManagerFilter() {

return new SDFSchemaManagerFilter();

}

（6）作者模式支持文档和文档节之间基于链接的导航，如果文档包含定义为指向其他元素（例如，基于id属性的链接），扩展会提供查找引用内容的方法。所以，执行ro.sync.ecss.extensions.api.link.ElementLocatorProvider接口应该使用createElementLocatorProvider方法。每次需要定位由链接指向的元素时，都会调用该方法。

举例如下：

public ElementLocatorProvider createElementLocatorProvider() {

return new DefaultElementLocatorProvider();

}

（7）拖放功能可以通过调用ro.sync.exml.editor.xmleeditor.pageauthor.AuthorDnDListener接口来实现。当鼠标拖动、移动或退出作者编辑模式，以及当跌落动作发生改变或当发生跌落时，调用来自监听器的相关方法。每个方法都接收包含有关拖放操作信息的DropTargetEvent。拖放扩展在作者模式下可用于oxyxml编辑器Eclipse插件和独立应用程序。文本模式对应的侦听器仅适用于Oxygen XML编辑器Eclipse插件。每个实现对应的方法是：createAuthorAWTDndListener、createTextSWTDndListener和createAuthorSWTDndListener。

举例如下：

public AuthorDnDListener createAuthorAWTDndListener() {

return new SDFAuthorDndListener();

}

（8）包中可以包含的另一个扩展是引用解析器。ref元素表示的引用和指示

引用的资源是位置。如果要获得引用资源的内容，团队必须实现一个Java扩展类：ro.sync.ecss.extensions.api.AuthorReferenceResolver。负责创建自定义引用解析器的方法是createAuthorReferenceResolver。实例化的引用解析器对象将保留并使用，直到检测过程激活与另一文档类型对应的另一个扩展捆绑包。  
 举例如下：

public AuthorReferenceResolver createAuthorReferenceResolver() {

return new ReferencesResolver();

}

（9）动态定制CSS样式：ro.sync.ecss.extensions.api.node.AuthorNode可以提供ro.sync.ecss.extensions.api.StylesFilter。负责创建StylesFilter的扩展包方法是createAuthorStylesFilter。在检测过程中激活与另一文档类型对应的另一个扩展包之前，将保留并使用实例化的筛选器对象。

举例如下：

public StylesFilter createAuthorStylesFilter() {

return new SDFStylesFilter();

}

（10）要自定义表格格式编辑数据，需要执行ro.sync.ecss.extensions.api.AuthorTableCellSpanProvider和调用ro.sync.ecss.extensions.api.AuthorTableColumnWidthProvider接口。ExtensionsBundle中指定这两个扩展点的两个方法是createAuthorTableCellSpanProvider和createAuthorTableColumnWidthProvider。

举例如下：

public AuthorTableCellSpanProvider createAuthorTableCellSpanProvider() {

return new TableCellSpanProvider();

}

public AuthorTableColumnWidthProvider

createAuthorTableColumnWidthProvider() {

return new TableColumnWidthProvider();

}

（11）XML词汇表可以包含指向文档各个区域的链接，如果文档包含定义为链接的元素，则可以为每个链接提供更相关的文本说明。为此，ro.sync.ecss.extensions.api.link.LinkTextResolver的实现接口应该由createLinkTextResolver方法返回。

举例如下：

public LinkTextResolver createLinkTextResolver() {

return new DitaLinkTextResolver();

}

（12）将编译后的类打包到JAR文件中。

（13）将JAR文件复制到自定义框架目录中（例如：frameworks/sdf）。

（14）将JAR文件添加到类路径：点击“首选项”对话框（选项>首选项），再点击“文档类型关联”，选择文档类型，点击Edit，选择Classpath选项卡，然后按Add按钮。在显示的对话框中，输入JAR文件相对于Oxygen XML frameworks文件夹的位置。

（15）点击Extensions选项卡，注册Java类。按“选择”按钮并选择类的名称（例如：SDFExtensionsBundle）。

2、通过JAVA接口

通过Oxygen xml编辑器作者模式扩展功能，有一组内置操作，包括插入文本和XML片段，以及在作者模式下编辑的当前文档上执行XPath表达式。但是在某些情况下，用户需要对其进行扩展，比如下列情况：

•用户需要输入一个元素，该元素的属性将由用户通过图形用户界面编辑。

•用户必须将选定的元素内容（或整个文档）发送到服务器进行处理。

•作者需要从服务器中提取信息，并将其直接插入已编辑的XML文档。

•需要在当前文档上应用XPath表达式，并处理结果节点集的节点。

举例如下：使用作者模式下的对话框，添加了用于在自定义框架中插入图像的功能，图像由图像元素表示，图像文件的位置由ref属性的值表示。在Java实现中，将显示一个带有文本字段的对话框，用户可以在其中输入完整的URL或浏览本地文件。

具体步骤如下所示：

（1）设置一个示例项目。

（2）修改ro.sync.ecss.extensions.api.AuthorOperation接口。此接口定义了三种方法：doOperation、getArguments和getDescription。

（3）将编译后的类打包到JAR文件中。

（4）将sdf.jar文件复制到自定义框架目录（[OXYGEN\_INSTALL\_DIR]\frameworks\[custom\_framework\_DIR]）中。

（5）将sdf.jar添加到类路径。点击“首选项”对话框（选项>首选项），转到“文档类型关联”，选择“SDF”，然后按“编辑”按钮。

（6）选择“文档类型配置”对话框中的“类路径”选项卡，然后按“添加”按钮，在显示的对话框中，输入JAR文件相对于Oxygen xml框架文件夹的位置。

（7）点击操作子选项卡，复制自定义框架目录中菜单项和工具栏的图标文件。

（8）定义操作的属性。

（9）进行设置。仅当当前元素是节时，才添加图像、数目或文章。

•将XPath表达式的值设置为

local-name()='section' or local-name()='book'

or local-name()='article'

•将调用操作字段设置为simple.documentation.framework.InsertImageOperation。

（10）使用工具栏面板将操作添加到工具栏。打开或创建xml文件，并将光标放在有效位置，然后按工具栏上与操作关联的按钮，在对话框中选择图像URL并按OK按钮，图像将插入到文档中。

专题二、常见问题的起因与解决方案

用户在使用Oxygen xml时可能会遇到许多问题，本专题就其中常见的一些问题进行分析，包括问题产生的原因、具体表现和解决方案等，从而为用户提供参考和帮助。

1、外部进程内存不足

（1）问题描述：Oxygen xml编辑器在尝试使用内置的ApacheFop处理器生成PDF输出时抛出内存不足错误。

（2）产生原因：分配的内存量可能不足。

（3）解决方案：

①打开“首选项”对话框（选项>首选项），转到XML>XSLT-FO-XQuery>FO处理器，并增加内置FOP可用的内存值。

②对于外部XSL-FO处理器、XSLT处理器和外部工具，使用Java虚拟机的-XMX参数集，在命令行中设置所分配内存的最大值。

2、Linux或Solaris上的显示问题

（1）问题描述：我在Linux或Solaris上遇到了显示问题（例如屏幕冻结）。

（2）产生原因：可能是屏幕外pixmap支持的渲染问题。

（3）解决方案：添加启动参数-Dsun.java2d.pmoffscreen=false

3、运行转换时嵌套的应用模板调用过多

（1）问题描述：当使用默认的Oxygen XML编辑器DocBook to HTML转换方案将DocBook文件转换为HTML时，收到错误消息提示嵌套的应用模板调用太多。

（2）产生原因：可能是屏蔽堆栈溢出错误的结果。

（3）解决方案：尝试使用-Xss4m值设置一个新的VM选项，也可以尝试将其慢慢增加到更大的值（例如-Xss5m或-Xss6m）。

4、大型文档的性能问题

（1）问题描述：当编辑大型文档时，应用程序的性能会随着时间的推移而大大降低。

（2）产生原因：应用程序需要更多的内存才能正常运行。

（3）解决方案：在运行应用程序的平台的配置文件中设置XMX参数，增加XML编辑器可用的最大内存量。

5、在远程桌面上使用Oxygen xml编辑器时的性能问题

（1）问题描述：尝试在远程桌面协议（RDP）环境中运行Oxygen xml编辑器时，性能缓慢且不稳定。

（2）产生原因：在缓慢的RDP连接上运行独立版本的Oxygen XML编辑器可能会导致性能下降。

（3）解决方案：可以尝试将Oxygen XML编辑器作为Eclipse插件运行。

6、启动应用程序等待时间过长

（1）问题描述：Oxygen xml编辑器需要非常长的时间才能启动。

（2）产生原因：一些防病毒软件需要扫描压缩档案（如所有Java应用程序使用的JAR库），所以会导致Java应用程序（如Oxygen xml编辑器）启动非常缓慢。

（3）解决方案：将Oxygen xml编辑器文件夹添加到防病毒软件设置中的异常列表中。

7、无法将Oxygen xml编辑器与Windows上的文件类型相关联

（1）问题描述：当用户在Windows文件浏览器对话框中选择文件oxygen21.1.exe时，Oxygen xml编辑器应用程序不会添加到“打开方式”对话框中的应用程序列表中。

（2）产生原因：此问题通常是由于以前版本的Oxygen xml编辑器中保留的旧Windows注册表项造成的。

（3）解决方案：卸载所有已安装版本的Oxygen xml编辑器，并重新安装当前版本的Oxygen xml编辑器。如果只安装了当前版本，则可以直接重新安装（无需先卸载）。

8、无法在Internet Explorer中打开XML文件

（1）问题描述：当用户试图在IE中打开一个XML文件时，它会在OxygenXML编辑器中打开该文件，而不是在IE中。

（2）产生原因：因为IE使用Windows系统文件关联来打开文件，并且用户在名为“文件关联”的安装程序面板中将XML文件与Oxygen xml编辑器关联，所以XML文件是在编辑器中打开的。

（3）解决方案：要在IE中打开XML文件，必须将IE设置为XML文件的默认系统应用程序。具体步骤如下：

①右键单击Windows资源管理器中的XML文件，然后选择“打开方式>选择默认程序”。

②在应用程序列表中选择IE。

③选择“始终使用选定程序”选项。

9、无法在Windows中卸载Oxygen xml编辑器

（1）问题描述：当我尝试在Windows中卸载Oxygen xml编辑器时，消息提示无法找到files.log文件。

（2）产生原因：Oxygen xml编辑器使用的install4j安装程序在安装过程中创建files.log文件。如果无法卸载产品，则很可能在安装过程中此文件出错。

（3）解决方案：只需将软件重新安装到当前安装所在的目录中，旧版本会自动卸载。

10、在Windows上启动时崩溃，出现有关nvoglv32.dll文件的错误

（1）问题描述：用户试图在Windows上启动Oxygen xml编辑器，但它崩溃了，并弹出了“Fault Module Name: nvoglv32.dll”。

（2）产生原因：可能是OpenGL驱动程序的问题。

（3）解决方案：在Oxygen xml编辑器安装中创建一个名为opengl32.dll的空文件即可避免这种情况。