# 菜单和菜单命令

DW根据在Configuration/Menus文件夹的menus.xml文件的结构声明创建所有菜单。你可以通过编辑menus.xml文件重排，重命名，移除菜单命令。你也可以添加，修改，移除菜单命令的快捷键。不过，通过快捷方式编辑器去修改快捷键是一个较容易的方法。通过重启或重加载扩展可以使dw菜单生效。

# menus.xml文件

menus.xml文件包含菜单项，菜单，菜单命令，分隔符，快捷列表，和快捷键列表。XML标签描述以上内容，你可以用文件编辑器编辑它。

*注：小心编辑menus文件，dw忽略所有包含xml语法错误的菜单及菜单命令。*

在多用户操作系统，你可以用dw修改menus.xml.如修改快捷键使用快捷键编辑器。这种情况下，dw修建一个menus.xml文件在用户文件夹里。可以通过拷贝menus.xml到相应文件夹来完成多用户系统的同步。

如果用一个xml编辑器打开menus.xml文件，可能会看到错误信息因为在menus.xml有&符号。最好用文件编辑器来编辑menus.xml文件，不要用dw。关于XML的基本信息可以查看dw帮助。

*注：总会在你修改menus.xml文件时备份。如果你忘备份，可以在Configuration文件夹中，用menus.bak替换menus.xm。*

一个菜单栏是分开菜单或菜单集。例如，有一个主菜单栏，一个单独的网站面板菜单栏，和一个包含上下文菜单的菜单栏。一个菜单栏包括一个或多个菜单，菜单被一个菜单标签包含。

每个菜单包括一个或多个菜单命令，描述了菜单项标记和它属性。一个菜单也包括分隔符和子菜单。

除了主键盘快捷菜单命令，dw提供备用和上下文相关的快捷键。例如：control+Y是Redo的快捷键。同时control+shift+z也是Redo的快捷键。这些交替和上下文相关的快捷方式不能被菜单命令标签表示在menus.xml文件中的快捷方式列表。每个快捷列表包括一或多个快捷键，被快捷标签描述。

下一节介绍menus.xml文件中的标签语法。选项属性被标记在属性列表用{}包括，所有属性被标记在{}中提及。

# <menubar>

**描述**

在dw菜单结构中，提供关于菜单栏的信息

**属性**

name,{app},id,{platform}

·**name** 菜单栏名称。虽然name是一个必须项，可以设置为""。

·**app**应用程序在菜单栏的名称。现在未使用。

·**id** 菜单栏的菜单ID.menus.xml文件中menui ID必须是唯一的。

·**platform** 指明只会在给定的平台上显示的菜单栏。合法值是"win"和"mac"

**内容**

这个标签必须包含一或多个menu标签

**容器**

无

**例**

主菜单使用下面的菜单栏标签：

<menubar name="Main Window" id="DWMainWindow">

<!-- 菜单标签放在这 -->

</menubar>

# <menu>

**描述**

在dw菜单结构中，提供关于菜单或子菜单信息

**属性**

name,{app},id,{platform},{showIf}

　　·　**name** 在菜单栏显示的菜单名称。在windows中设置菜单的访问键在字符前使用“\_”。

·**app**应用程序在菜单栏的名称。现在未使用。

·**id**菜单ID。在文件中每个ID唯一的。

·**platform**指明只会在给定的平台上显示的菜单栏。合法值是"win"和"mac"

**·showIf** 只有给定dw判断值为true,菜单显示。可选项为：\_SERVERMODEL\_ASP,

\_SERVERMODEL\_ASPNET,\_SERVERMODEL\_JSP,\_SERVERMODEL\_CFML,

\_SERVERMODEL\_CFML\_UD4,\_SERVERMODEL\_PHP,\_FILE\_TEMPLATE,\_VIEW\_CODE,

\_VIEW\_DESIGN,\_VIEW\_LAYOUT,\_VIEW\_EXPANDED\_TABLES,\_VIEW\_STANDARD。你可以使用多个并以逗号分隔。你可以指定否用“！”。如，你想要你的菜单在asp页面的代码视图显示，

指定属性为 showIf="\_VIEW\_CODE,\_SERVERMODEL\_ASP"。

**内容**

这个标签可以包括一或多个menuitem标签，和一或多个separator标签。它也可以包含menu标签和标准的html注释标签

**容器**

这个标签必须放在menubar标签中。

**示例**

<menu name="\_File" id="DWMenu\_File">

<!-- 可包含 menuitem,separator,menu,和注释标签 -->

</menu>

# <menuitem>

**描述**

为dw菜单定义一个菜单命令。

**属性**

name,id,{app},{key},{platform},{enabled},{arguments},{command},{file},{checked},{dynamic}

{isdomrequired},{showIf}

**·name** 菜单上显示菜单命令的名称。以下划线开头后跟命令关键字。只有windows系统。

**·id** 用于dw标识项。这个ID在菜单结构中必须是唯一的。如果你在menus.xml文件中添加一个新的菜单命令，确保菜单命令的ID的唯一性，使用公司名或其它唯一字符串做为前缀.

·**app**应用程序在菜单栏的名称。现在未使用。

·**key** 命令快捷键，使用下列字符串做为修饰键值

·**Cmd** 指定为control 键

·**Alt**和Opt　可交换指定Alt键(windows) 或Option键(Mac)

·**Shift** 指定Shift键

·**Ctrl**　指定Control 键

·用加号分隔修饰键，如果使用多个字符，如Cmd+Opt+5　键值属性意思着当用 户按Control+Alt+5时菜单命令运行

·特殊键值用特定的名称指定：F1到F12,PgDn,PgUp,Home,End,Ins,Del,Tab,Esc,BkSp 和Space。修饰键值可以运用到特殊键。

·**platform**指明只会在给定的平台上显示的菜单栏。合法值是"win"和"mac"

·**enabled** 运用javascript代码来决定当前菜单命令是否可用。如果函数返回false,菜单命令暗灰色不可用。默认为“true”，但最好是清晰指定一个值即使值为“true”

**·arguments** 为dw的javascript文件中的代码提供参数，参数是文件属性中指定。 用单引号括着参数，用双引号去界定属性值

**·command** 指定一个javascript表达式当用户从菜单上选择这个项时运行。对于复杂的javascript代码，使用javascript文件（在file属性上指示）代替。你必须为每个命令指定file或command。

**·file** 一个包含javascirpt的html文件，它控制菜单命令。指定对于Configuration文件夹的一个相对路径。（如，Help>欢迎菜单命令指定file="Command/Welcom.htm".）file属性重载了command,enabled,checked属性。你必须为每个菜单命令指定file或command。

有关使用历史面板创建一个命令文件的信息，请参见Dreamweaver帮助。如果如何写自定义javascript命令参阅Commands。

**·checked** 一个JavaScript表达式，指示菜单命令是否旁边有一个复选标记菜单;如果表达式结果为true，项边显示一个复选标识。

**·dynamic** 如果存在，指示菜单命令由HTML文件运行过程中动态确定。文件包含javascript代码设置菜单命令的文本和状态。如果指定一个标签为dynamic，你必须指定一个file属性。

**·isdomrequired** 指明是否在运行菜单命令前同步设计视图和代码视图。有效值为“true” 和“false”。如果你设置属性值为“false”，表示更改菜单命令该文件不使用dw的DOM对象。更多DOM信息，请参阅dw Document Object Model

**·showIf** 只有给定dw判断值为true,菜单显示。可选项为：\_SERVERMODEL\_ASP,

\_SERVERMODEL\_ASPNET,\_SERVERMODEL\_JSP,\_SERVERMODEL\_CFML,

\_SERVERMODEL\_CFML\_UD4,\_SERVERMODEL\_PHP,\_FILE\_TEMPLATE,\_VIEW\_CODE,

\_VIEW\_DESIGN,\_VIEW\_LAYOUT,\_VIEW\_EXPANDED\_TABLES,\_VIEW\_STANDARD。你可以使用多个并以逗号分隔。你可以指定否用“！”。如，你想要你的菜单在asp页面的代码视图显示，

指定属性为 showIf="\_VIEW\_CODE,\_SERVERMODEL\_ASP"。

**内容**

无（空标签）

**容器**

此标签必须包含在menu标签中

**示例**

<menuitem name="\_New" key="Cmd+N" enabled="true" ommand="

dw.createDocument()" id="DWMenu\_File\_New" />

# <separator>

**描述**

指示分隔符将要出现在菜单的相应位置

**属性**

{app}

**·app**应用程序在菜单栏的名称。现在未使用。

**内容**

无（空标签）

**容器**

标签必须放在menu标签内

**示例**

<separator />

# <shortcutlist>

**描述**

在menus.xml文件中指定快捷方式列表

**属性**

{app},id,{platform}

**·app**应用程序在菜单栏的名称。现在未使用。

**·id** 快捷方式列表ID。它应该和dw中菜单栏中菜单ID(或上下文菜单)的快捷方式相关联。有效值为“DMMainWindow”,“DWMainSite”,“DWTimelineContext”,“DWHTMLContext”。

·**platform**指明只会在给定的平台上显示的菜单栏。合法值是"win"和"mac"

**内容**

此标签包含一或多个shortcut标签。也可以包含一或多个注释标签。

**容器**

无

**示例**

<shortcutlist id="DWMainWindow">

<!-- 快捷方式标签或注释标签 -->

</shortcutlist>

# <shortcut>

**描述**

在menus.xml文件中，指定键盘快捷方式

**属性**

key,{app},{platform},{file},{arguments},{command},id,{name}

**·key** 激活键盘快捷盘的组合键。具体语法参阅　menuitem

**·app**应用程序在菜单栏的名称。现在未使用。

·**platform**指明只会在给定的平台上显示的菜单栏。合法值是"win"和"mac"

**·file** 当用户使用键盘快捷键时dw根据文件地址运行其中的javascript代码。file属性会覆盖command属性。必须为每个快捷方式指定file或command属性。

**·arguments** 为dw的javascript文件中的代码提供参数，参数是文件属性中指定。 用单引号括着参数，用双引号去界定属性值

**·command** 指定一个javascript表达式当用户使用键盘快捷键时运行。你必须为每个命令指定file或command。

**·id**快捷方式唯一标识符

**·name** 通过键盘快捷键运行命令的名称，菜单命令的名称。如，为F12快捷方式的name 属性设置为'在主浏览器浏览'。

**内容**

无（空标签）

**容器**

此标签必须包含在shortcutlist标签内

**示例**

<shortcut key="Cmd+Shift+Z" file="Menus/MM/Edit\_Clipboard.html" arguments="'redo'"

id="DWShortcuts\_Edit\_Redo" />

# <tool>

**描述**

代表一个工具，它包含此工具所有的快捷方式，子标签，在menus.xml文件中

**属性**

{name},id

**·name** 本地化版本工具的名称

**·id** 内部工具标识，标识该工具的快捷键。

**内容**

此标签可以包含一或多个activate,override,action标签

**容器**

此标签必须放到menu标签中

**示例**

<tool name="Hand tool" id="com.Macromedia.tools.hand">

<!-- 工具标签 -->

</tool>

# <action>

**描述**

当工具激活，组合键按下时运行的组合键值和javascript代码

**属性**

{name},key,command,id

**·name** 本地化版本行为。

**·key**组合键用于执行行为。具体语法，参见menuitem.

**·command** 可运行的javascript语句块。此属性格式和shortcut的command属性相同。

**·id** 唯一的ID用于参考的行为。

**内容**

无（空标签）

**容器**

此标签包含在tool标签

**示例**

<action name="Set magnification to 50%" key="5" command="dw.activeViewScale=0.50" id="DWTools\_Zoom\_50" />

# <activate>

**描述**

包含激活工具的组合键

**属性**

{name},key,id

**·name** 本地化版本行为。

**·key**组合键激活工具。具体语法，参见menuitem.

**·id** 唯一的ID用于参考的行为。

**内容**

无（空标签）

**容器**

此标签包含在tool标签

**示例**

<activate name="Switch to Hand toll" key="H" id="DWTools\_Hand\_Active1" />

# <override>

**描述**

包含组合键暂时激活工具。而在另一个模态的工具,用户可以按住这个键切换到这个工具

**属性**

{name},key,id

**·name** 本地化版本行为。

**·key**快速激活工具使用的组合键。具体语法，参见menuitem.

**·id** 唯一的ID用于参考的行为。

**内容**

无（空标签）

**容器**

此标签包含在tool标签

**示例**

<override name="Quick swith to Hand tool" key="Space" id="DWTools\_Hand\_Override" />

# 改变菜单和菜单命令

通过编辑menus.xml文件，我们可以把菜单命令从一个菜单移动到另一个菜单，添加分隔符或从菜单中移除它们，从一个菜单栏移动到另一个菜单栏。

可以从上下文菜单移进或移出项，用和操作其它菜单一样的方法。

详细信息，参阅menus.xml文件。

**移动一个菜单命令**

1、退出dw,

2、备份menus.xml文件

3、用文件编辑器打开menus.xml文件（不要使用dw打开）。

4、剪切menuitem实例标签，从<menuitem到/>.

5、将插入点定位到新的菜单命令位置。（确保它是在menu标签和相对应/menu标签中）。

6、粘贴菜单命令到新的地方。

**创建子菜单当移动菜单命令时**

1、将插入点移动到menu标签内（menu到/menu之间）。

2、在menu标签内插入一对新的menu 和 /menu 标签。

3、给新的子菜单添加菜单命令。

**在菜单命令之间添加分隔符**

·把separator/标签放到两个menuitem标签之间。

**移除已存在的分隔符**

·删除对应的sepatator/ 线。

**移动菜单**

1、退出DW

2、备份menus.xml文件

3、用文本编辑器打开menus.xml文件

4、剪切menu实例和内容，从menu到/menu之间。

5、定位到新位置。确保新位置是在menubar标签内。

6、粘贴菜单到新位置。

# 修改菜单及菜单命令名称

可以通过修改menus.xml轻松修改任何的菜单及菜单命令名称。

1、退出dw

2、备份menus.xml

3、文件编辑器打开menus.xml

4、修改菜单命令，找到menuitem标签的name属性。修改菜单，找到menu标签的name属性。以上都不要改变id属性的值。

5、保存menus.xml文件，再次打开dw时就可以看到修改结果。

# 修改键盘快捷方式

如果觉得默认的键盘快捷方式用着不舒服，你可以添加，修改，删除键盘快捷方式。最快捷的方法是使用快捷方式编辑器。同样的你也可以修改menus.xml文件，但容易出现错误。

1、退出dw

2、备份menus.xml

3、文件编辑器打开menus.xml

4、对照dw有效的键盘快捷方法对照表，找一个没有被使用的快捷方式。

改一下快捷方式对照表，以方便你的使用。

5、如果你新分配了一个键盘快捷方式，在菜单命令中已有注册，可以移除已存在的菜单命令的key="shortcut" 属性。

6、分配或重新分配菜单命令。

7、如果你新分配了一个键盘快捷方式，找到对应的菜单命令。如果没有快捷方式，就在mentuitem标签添加key="" 属性.

8、在key属性的属性值“”内放入新的键盘快捷方式。使用+连接多个键值。更多信息可以查看menuitem标签的描述

如果你用了一个和现有重复的快捷方式，那么快捷方式将运行在menus.xml出现在上面的命令。

9、更新你的快捷键对照表。

# 弹出菜单和上下文菜单

dw在许多面板和弹窗内提供弹出菜单和上下文菜单。上下文菜单一些在menus.xml文件中定义，一些在其它的XML文件中定义。你可以添加，移动，修改这个菜单项目，但写一个扩展实现这些改变更好一些。

下面是一些dw中弹出菜单和上下文菜单，在xml文件中被指定。在menus.xml文件中使用相同的标签。

·数据源是在DataSources.xml文件中指定，在DataSources文件夹的子文件夹中

·服务器行为在ServerBehaviors文件的子文件夹的ServerBehaviors.xml文件中规定。

·服务器格式在ServerFormats文件的子文件夹的ServerFormats.xml文件中规定。

·在绑定面板上的项目弹出菜单在ServerFormats文件夹的子文件夹中的Formats.xml文件

中指定。你可以使用 dw提供的添加Format对话窗添加新实例到菜单中。

·标签库对话窗菜单命令在TagLibraries/TagImporters/TagImporters.xml文件中规定

·在对话框生成器生成菜单命令参数，是服务器行为生成器的一部分，在

Shared/Controls/String Menu/Controls.xml规定。

·与ColdFusion组件相关联的上下文菜单中的项目在Components/ColdFusion/CFCs/CFCsMenus.xml规定

·与ColdFusion数据源相关联的上下文菜单中的项目在Components/ColdFusion/DataSources/DataSourcesMenus.xml规定

·上下文菜单项与各种服务器组件相关联的XML文件中指定的，在Components文件夹的子文件夹

# 简易菜单命令示例

一个简单的完成撤消和重做菜单命令的示例。撤消菜单命令恢复用户编辑操作，重做是恢复撤消操作并保存用户最后的操作效果。

可以通过写js代码，修改Menu文件夹的命令文件来创建菜单命令的示例。

# 创建菜单命令

创建一个包含Undo和Redo菜单命令的菜单，取名MyMenu把下面的html菜单标签添加到menus.xml文件中最后一个封闭的</menu>标签后。

<menu name="MyMenu" id="MyMenu\_Edit">

<menuitem name="MyUndo" key="Cmd+Z" file="Menus/MyMenu.htm" arguments="'undo'" id="MyMenu\_Edit\_Undo">

<menuitem name="MyRedo" key="Cmd+Z" file="Menus/MyMenu.htm" arguments="'redo'" id="MyMenu\_Edit\_Redo">

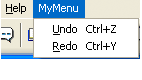
</menu>

key属性声明键盘快捷键，用户可以通过输入加载菜单命令。

file 属性指定当DW加载命令时dw运行的菜单命令文件的地址。

arguments 属性的值声明dw通过调用receiveArguments()函数接收到的值。

下图为菜单命令的图例



# 菜单javascript代码

当用户选择MyMenu菜单中的Undo或Redo时，dw通过menuitem标签的file属性的值，找到MyMenu.html文件并运行这个文件中的javascript代码。在dw的Configuration/Menus文件夹添加MyMenu.html命令文件，并添加三个菜单命令函数,

canAcceptCommand(),receiveArguments(),setMenuText(),实现逻辑相关的撤消和重做的菜单项。以下介绍这三个函数。

**canAcceptCommand()**

dw调用canAcceptCommand()函数以确定MyMenu菜单的菜单项哪些是可用，哪些是不可用的。在MyMenu.htm文件中，canAcceptCommand()函数检测arguments[0]的值，确定是否dw运行了Redo或Undo。如果参数是"undo"，canAcceptCommand()函数调用函数dw.canUndo()并且返回该函数的返回值。如果参数是"redo",canAcceptCommand()函数调用函数dw.canRedo()并且返回该函数的返回值。如果canAcceptCommand()函数返回false,dw退去菜单项调用的函数。以下为示例中canAcceptCommand()函数代码：

function canAcceptCommand(){

var selarray;

if(arguments.length!=1)return false;

var bResult=false;

var whatToDo=arguments[0];

switch(whatToDo){

case 'undo':

bReslut= dw.canUndo();

break;

case 'redo':

bResult=dw.canRedo();

break;

}

return bResult;

}

**receiveArguments()**

dw通过调用receiveArguments()函数处理menutitem标签中定义的参数。Undo和Redo菜单项，receiveArguments函数根据参数arguments[0]的值是'undo'或'redo'来调用dw.undo()函数或dw.redo()函数。dw.undo()函数将撤消用户在文档窗口，弹出框或当前面板的操作。dw.redo()函数将重做最后的撤消操作。

下面是示例中receiveArguemtns()函数的代码

function receiveArguments(){

if(arguments.length!=1)return;

var whatToDo=arguments[0];

if(whatToDo=='undo'){

dw.undo()

}else if(whatToDo=='redo'){

dw.redo();

}

}

在这个命令中,receiveArguments函数通过处理参数来运行命令。复杂的菜单命令需要调用不同的函数来运行命令。例如下面的代码检查第一个参数是'foo'，如果是调用doOperationX()函数并用传入第二个参数。如果第一个参数是'bar'调用doOperationY()函数并传入第二个参数。函数doOperationX或doOperationY负责运行命令。

function receiveArguments(){

if(arguments.length!=1)return;

var whatToDo=arguments[0];

if(whatToDo=='foo'){

doOperationX(arguments[1])

}else if(whatToDo=='bar'){

doOperationY(arguments[1])

}

}

**setMenuText()**

dw调用setMenuText函数来确定菜单项显示的文字。

如果没有定义setMenuText()函数，dw使用在menuitem标签的name的属性值。

setMenuText函数检测从dw传送进的参数arguments[0]。如果参数值是'undo'，dw调用dw.getUndoText()函数，如果值是'redo',dw调用函数dw.getRedoText()函数。

dw.getUndoText()函数返回dw将撤消操作的文本.例如,如果用户运行复杂的Redo操作，dw.getUndoText()会返回返回菜单文字'Undo Edit Source'.同样的，dw.getRedoText()函数返回的文本，指定dw将要进行redo操作。如果用户运行多步撤消操作，dw.RedoText函数在菜单上返回'Redo Edit Source'文本。

示例中setMenuText函数代码

function setMenuText(){

if(arguments.length!=1)return;

var whatToDo=arguments[0];

if(whatToDo=='undo'){

return dw.getUndoText();

}else if(whatToDo=='redo'){

return dw.getRedoText();

}else return "";

}

# 在Menu文件夹放入命令文件

为了实现撤消和重做的菜单项命令，必须dw的Configuration/Menus文件夹或子文件夹放入你创建的MyMenu.htm命令文件。文件位置是你在menuitem标签指定的file的地址。

并通过重启dw或重加载扩展来使其生效。更多重加载的信息参阅reloading extendsions

# 菜单命令的API函数

* [canAcceptCommand()](http://help.adobe.com/en_US/dreamweaver/cs/extend/WS5b3ccc516d4fbf351e63e3d117f508c8de-7fb4.html)
* [commandButtons()](http://help.adobe.com/en_US/dreamweaver/cs/extend/WS5b3ccc516d4fbf351e63e3d117f53d704a-7ff0.html)
* [getDynamicContent()](http://help.adobe.com/en_US/dreamweaver/cs/extend/WS5b3ccc516d4fbf351e63e3d117f508c8de-7f9e.html)
* [isCommandChecked()](http://help.adobe.com/en_US/dreamweaver/cs/extend/WS5b3ccc516d4fbf351e63e3d117f508c8de-7fb7.html)
* [receiveArguments()](http://help.adobe.com/en_US/dreamweaver/cs/extend/WS5b3ccc516d4fbf351e63e3d117f508c8de-7fb6.html)
* [setMenuText()](http://help.adobe.com/en_US/dreamweaver/cs/extend/WS5b3ccc516d4fbf351e63e3d117f508c8de-7fb5.html)
* [windowDimensions()](http://help.adobe.com/en_US/dreamweaver/cs/extend/WS5b3ccc516d4fbf351e63e3d117f53d704a-7fd6.html)