HG XDOC USER GUIDE

XDOC 用户手册

(6.1.9)



HG WORKROOM 版权所有 2005-2010

前 言

报表是一个老生常谈的一个话题,谈了搁下,过了又谈。始终找不到好的解决方法。究 其原因主要有两种:数据复杂、格式变幻无常。报表是数据和格式的结合,其实针对到其中 的某一个问题,还是有解决方法的:

数据:目前数据处理最深入人心的是 SQL, SQL 以其功能强大,易学易用被人接受。

格式: Wps、MsOfficeWord,等被人熟练使用。

有一个问题,这两个东西都有了,是不是问题就解决了。不是的,还需要一个纽带将这 两者结合起来,这个纽带就是程序,因为报表本身确实是程序。但不能让普通用户写程序。 除非能可视化的操作,通过拖拽实现。

如果能实现以上的想法,做出来的东西将是很具有吸引力的。我想来实现我的梦想。

是梦想,因为距离很遥远,我的精力也有限。做这些就涉及到绘图、排版打印等一系列 东西。而我对这一窍不通。好在我已经实现了 HGSQL 引擎,一个异构 XML 数据库引擎。 完成后的成果是很具有吸引力的,试想一下,来之网络、文本、数据库、excel 文件等数据 同时展现在一张报表上...

实现是很困难的,在一个 JComponent 上实现一个编辑器,还要支持图文混排等。所有的代码都有自己写,选择文本,设置字体,前后移动光标等,这些平时用起来不当回事的动作都要实现,我以前对这些东西从来没有接触过。好多次都打击的我不想再做了,但还是坚持下来了,成功的喜悦也是别人难以体会的。

虽然,现在做出来的东西,我不满意,您可能也不满意。但这是我目前能做到的,我会继续努力的,我想它一点会一天一天好起来的,这离不开您的支持。

请记住我的 email: <u>sqlhub@163.com</u>,您可以到 <u>http://www.hgsql.com</u>上获得最新的版本 和帮助文档。

作者 2006-08-20

目 录

1.	安装	ē部署		5
	1.1.	安装	Java 运行环境	5
	1.2.	安装	应用服务器	5
		1.2.1.	下载安装包	5
		1.2.2.	解压安装	5
	1.3.	部署	程序	6
		1.3.1.	下载程序包	6
		1.3.2.	部署	6
2.	客户	'端	······	7
	2.1.	启动	J	7
		2.1.1.	服务器	7
		2.1.2.	设计器	8
	2.2.	主界	·面	8
	2.3.	连接	到服务器	8
		2.3.1.	连接	8
		2.3.2.	断开	9
	2.4.	管理	!目录	9
		2.4.1.	创建	9
		2.4.2.	删除	9
		2.4.3.	重命名	9
	2.5.	管理	1文件1	0
		2.5.1.	添加1	0
		2.5.2.	重命名1	0
		2.5.3.	删除1	0
		2.5.4.	运行 XDOC1	0
		2.5.5.	上传1	0
		2.5.6.	下载1	1
	2.6.	查看	·服务器授权1	1
3.	设计	-器		1
	3.1.	启动	11	1
		3.1.1.	网络1	2
		3.1.2.	本地1	2
		3.1.3.	服务器1	2
	3.2.	主界	面1	3
	3.3.	工具	栏1	3
		3.3.1.	主工具栏1	3
		3.3.2.	编辑工具栏1	7
		3.3.3.	选项栏1	7
	3.4.	基础	设置区1	8
		3.4.1.	XML1	8
		3.4.2.	参数1	9

	3.4.3.	大纲	
	3.5. 辅	助设计区	
	3.5.1.	数据选择	
	3.5.2.	格式库	
	3.5.3.	图形库	
	3.5.4.	元件库	
	3.6. 属	性编辑器	
	3.7. 设	计视图	
	3.7.1.	文档标签	
	3.7.2.	字体工具栏	
	3.7.3.	段落工具栏	
	3.7.4.	排列与叠放层次	
	3.7.5.	组件工具栏	
	3.7.6.	图形工具栏	
	3.7.7.	辅助工具栏	
	3.7.8.	设计背景、前景	55
	3.7.9.	选择图形	55
	3.7.10.	复制粘贴	55
	3.7.11.	智能输入	
	3.8. 运	行视图	
	3.9. 查	看 XML	
	3.10. 其	它	
	3.10.1.	快捷键	
	3.10.2.	多语言	
4.	文档语法		
	4.1. 文	档语法	
	4.1.1.	嵌入表达式	
	4.1.2.	嵌入程序块	60
	4.1.3.	打印期表达式	60
	4.2. 文	档函数	60
	4.3. 函	数简化引用	62
	4.4. 计	算过程	
	4.5. 数	据集引用	
5.	扩展升发	<i>b</i>	
	5.1. 准	备	
	5.2. 丁	展兀忤	
	5.3. 丁	展您镜 早见换亡	
	5.4. 引	展图	
	5.5. 丁	炭图形库	
	5.6. 步,	展 ADAIA 述痒 タ 映版に	
	5.7. 版	务奋监听	
6	5.8. 引	면	
6.	JPD 开反	半 扣	
	6.I. N	轵 恎/户/牛	

6.2.	调月	月接口	67
	6.2.1.	JpdGraphics	67
	6.2.2.	Jpd	68
	6.2.3.	JpdViewer	74
	6.2.4.	Canvas	76
	6.2.5.	示例	80
6.3.	注册	丹为虚拟打印机	
6.4.	结界	导使用	82
	6.4.1.	独立应用程序	
	6.4.2.	嵌入到 swing 应用中	
	6.4.3.	嵌入到网页中	
6.5.	画图	图板	
6.6.	浏步	5器快捷键	

1. 安装部署

1.1.安装 Java 运行环境

安装地址: <u>http://www.java.com</u>

1.2. 安装应用服务器

XDOC 可以部署在所有 J2EE 应用服务器中,下面以 Tomcat 为例:

1.2.1. 下载安装包

地址:

http://archive.apache.org/dist/tomcat/tomcat-6/v6.0.20/bin/apache-tomcat-6.0.20.zip

1.2.2. 解压安装

安装完成后可以打开浏览器: <u>http://localhost:8080</u>,如果看到如下界面,表示安装成功:



1.3.部署程序

1.3.1. 下载程序包

地址: <u>http://www.hgsql.com/down/xdoc.war</u> 如果下载后文件名为 xdoc.zip,请重命名为 xdoc.war

1.3.2. 部署

方法 1:

点击 Tomcat 首页的 Tomcat Manager 进行部署,如下图:

WAR file to deploy						
Select WAR file to upload	C:\Users\wanghg\Desktop\xdoc.war	Browse				
	Deploy					

方法 2:

将 xdoc.war 直接复制到 tomcat 的 webapps 目录下。

打开浏览器: <u>http://localhost:8080/xdoc</u>,如果看到如下界面,表示部署成功:



- 2. 客户端
- 2.1.启动
- 2.1.1. 服务器



2.1.2. 设计器



2.2.主界面



2.3. 连接到服务器

2.3.1. 连接



@ `连接		×
服务器URL:	http://localhost:8080/xdoc/xdoc	٦
用户名:		
口令:		
	✔ 确定 ◆ 取消	

服务器 URL 为部署目录加"/xdoc"

2.3.2. 断开

@XDOC客户端6.1.4	
文件(E) 工具(D) 帮助(H)	

2.4.管理目录

@XDOC客户端6.1.4	
文件(E) 工具(D) 帮助(H)	
) 🗙 🛍 🕃 🕼 📣	
] 🕂 🗙 🥒 🧇	
根目录	

2.4.1. 创建

选择目录,点击添加按钮

2.4.2. 删除

选择目录,点击删除按钮。 如果目录下存在不可操作的文件类型的文件,目录将不能被删除。

2.4.3. 重命名

选择目录,点击重命名按钮。

第9页共84页

2.5.管理文件

XDOC 服务器可接收的文件类型有: xdoc、zdoc、jpg、png、gif、bmp

睂 XDOC客户端6.1.4	
文件(E) 工具(D) 帮助(H)	
] 🗙 🖺 🛃 🖏 🖪	
] 🕂 🗙 🥒 🧇	🚺 🕂 🥒 🗙 🔾 🕁
1 根目录	名称
	index.xdoc

2.5.1. 添加

打开文件选择框,选择文件上传到服务器。

2.5.2. 重命名

重命名列表中选择的文件

2.5.3. 删除

从服务器中删除列表中选择的文件

2.5.4. 运行 XDOC

在列表中选择 xdoc 或 zdoc, 点击运行按钮, 弹出如下对话框:

们 运行	×
保存到:	
格式: jar文档	
比例: 100	•
🗙 直接打开 📄 压缩 🔷 确定 🔷 取消	

"保存到"如果不填,将自动保存到临时目录。压缩选项

2.5.5. 上传

打开文件选择框,选择文件上传到服务器,覆盖已经存在的文件。

第 10 页 共 84 页

如果 XDOC 设计器已经打开,并且选择的文件类型为 xdoc 或 zdoc,会提示是否上 传编辑器中的文档。

2.5.6. 下载

下载列表中选择的文件。

如果 XDOC 设计器已经打开,并且选择的文件类型为 xdoc 或 zdoc, 会自动用 XDOC 设计器打开。

2.6. 查看服务器授权

点击主工具条的关于按钮,显示关于信息:



点击授权按钮,查看服务器授权



3. 设计器

3.1.启动

本地必须安装 Java 运行环境,安装地址: http://ww.java.com

3.1.1. 网络

完整版地址: <u>http://hgsqldoc.appspot.com/hxdoc.jnlp</u> 精简版地址: <u>http://www.hgsql.com/xdoc/lib/hg.jar</u>

3.1.2. 本地

下载: <u>http://www.hgsql.com/down/xdoc.war</u>,重命名为 xdoc.zip,解压。 Windows 中运行目录下的 XDocBuilder.bat, Linux 中运行目录下的 XDocBuilder.sh

3.1.3. 服务器



3.2.主界面

@ XDOC设计器6.1.4		
文件(E) 编辑(E) 工具(D) 帮助(H)		
] [* "] ൙ 🗀 🔛 🖪 🗎	📸 🕼 😂 🦻 🖧 🗍 🛹 💛 🗈 🛍 💰 🗶 🗎	🗈 💣 🕩 🚺 125 🗨 100 🛡
XML 参数 大纲	• 10 • B <i>I</i> <u>U</u> T S	0 点 🛱 #000000 🏾 🗮
地址: 🔁	新文档1	() √ ø
e o 🥖 🗙 🕂 🖉 📓	✓	▲ 日 名称: e49
查找: 🔍		
		近い、
		宽度: 124
	82 • • • • • • • • • • • • • • • • • • •	高度: 72
●列表 〇 方框 ▼ 〇 〔ì		▲ 排版方向: 横排 ●
	?	i i i i i i i i i i i i i i i i i i i
数据 格式 元件 图形		
查找:		
 		
□ ▲ 券粉		
□ 标签 □ 合计	With the second seco	
		1/1

3.3.工具栏

3.3.1. 主工具栏



新建

基于默认模板创建新文档,默认模板使用配置项 doc.template。

由模板创建

基于选择的文件创建新文档。设计器会弹出模板选择对话框,如下图:

打开							×
Ø Documents			•	۲	\$	ыř	
🥕 My eBooks							_
🔅 My RTX Files							
🔅 Remote Assistance Logs							
🕭 RTXC File List							
Tencent Files							
(*.xdoc;*.zdoc;*.txt;*.html;*.	htm;*.doc	;*.docx)					•
				(打开		取消

打开

编辑打开文档。设计器会弹出文件选择对话框,文件类型支持:xdoc、zdoc、html、doc、docx。

关闭

关闭当前编辑的文档,如果没有保存,会提示,如下图:



保存

保存当前编辑的文档,如果是新建没有保存过的文档,会弹出保存对话框,如果客 户端已经连接到服务器,会显示保持到服务器对话框,如下图:

保存到服	G务器	×
0	名称 xdoc.xdoc	
	确定取消	

另存为

将当前编辑的文档保存到其它目录下或保存为其它格式。设计器会显示另存为对话 框,如果客户端已经连接到服务器,会显示保持到服务器对话框。

文件名不输扩展名时,系统自动将文件类型的做扩展名,也可以直接输入扩展名转 为其它类型。

页面设置

设置当前文档的纸张类型、边距和方向。

@页面设:	5		×
纸张	biso-a4		•
宽 度(px)		595 🗣 高度(px)	841 🚔
	\square	无边距	
左边距(px)		72 🛢 右边距(px)	72 🖨
上边距(px)		72 🔷 下边距(px)	72 🚔
方向	💿 纵向		◯ 横向
🗸 确定 🔷 取消			

可以选择纸张,纸张改变后宽度和高度会自动改变,只有选择自定义,宽度和高度 才可以输入。

打印

打印当前编辑的文档。

点击取消, 会弹出保存到文件对话框。

▲打印	×
常规(G)页面设置(S)外观(A)	
打印服务 名称(N): Microsoft XPS Document Writer	 属性®…
状态: 接受任务 类型:	
信息:	☐ 打印到文件(E)
打印区域	份数
● 全部(L)	打印份数(②): 1 🗐
○ 页码范围(E) 1 至 1	□比较(C)
	打印取消

文档向导

辅助创建 XDOC。

公共多语言翻译

编辑公共多语言库,公共多语言库使用配置项 doc.language。公共多语言库为所有 XDOC 提供语言翻译服务。

XDOC 客户端

打开 XDOC 客户端。使用 XDOC 客户端可以连接到远程 XDOC 服务器,编辑、部署、运行 XDOC。

SQLHUB 管理器

打开 SQLHUB 管理器。

加载 Java 程序库

动态加载 JAVA 程序库。

退出

退出设计器,如果有修改未保存的文档,会依次提示保存。

关于

显示版本、JDK 环境等信息,如下图:



3.3.2. 编辑工具栏



3.3.3. 选项栏



显示比例

系统支持比例缩放,默认显示比例是125%

运行最大记录数

运行最大记录数用于控制文档计数过程中查询到的的记录数据,系统默认为100, 0表示不控制。设置比较小的值可以加快预览速度,方便设计。

第 17 页 共 84 页

简化引用

布局向导是否用简化方式引用字段、属性等。简化方式下对象、参数等名称会提示 自动转换为小写。

3.4. 基础设置区

3.4.1. XML

引入参考 XML



输入 XML 或 HTML URL 地址,点击引入参考按钮,如: <u>http://news.163.com/special/00011K6L/rss newstop.xml</u>
目前能识别的格式有: xml、html、xls、txt。
html、xls 和 txt 会自动转换为 xml
txt 以回车为行分割,如果首行存在 tab 符号,以 tab 分割为多列,如果存在",",以","分割为多列,否则识别为一列。
引入参考后进行布局。

自定义 XML



自定义 XML 结构,调用时依照定义的 XML 传入实际的 XML,输出结果。 点击添加元素、属性等定义 定义完成后布局,也可以导出定义的 XML。

布局

在参考 XML 树中找到待布局的数据,用鼠标拖拽到 XDOC 编辑器中。布局生成的结果由布局选项决定。可以在查找中输入内容进行快速定位。

布局选项由三类:



列表 布局为列表循环框,适合多个相同标签的元素,单个元素和属性也可以。
方框、文本... 布局绝对引用项,由5种子选项 方框:文本框 文本:不带框的文本 名称:xml标签名称 HTML:显示HTML XDOC:显示XDOC
统计 对XML 值进行统计,统计方式有5种:计数、求和、平均值、最大值、最小值

3.4.2. 参数

公共参数



公共参数可以被所有文档使用,使用配置项 doc.param

公共数据库连接

公共数据库连接可以被所有文档使用

```
1<?xml version="1.0" encoding="UTF-8"?>
2
3<data>
4
   5
     <row>
6
       <NAME>HGNET</NAME>
7
       <DRIVER>com.hg.jdbc.HgDriver</DRIVER>
8
       <URL>jdbc:hg:net:@localhost:1980:hg</URL>
9
       <USER>HG</USER>
10
       <PASSWORD>HG</PASSWORD>
     </row>
11
12
   13</data>
```

参数

使用参数可以增强文档的灵活性。系统支持公共参数和文档私有参数。

新建

选择参数结点后点击添加按钮或点击弹出菜单的添加菜单项



@创建参数	×
名称:	
标题:	
输入类型:	(文本 -
缺省值:	
可选值:	
非空:	
排序:	
	✔ 确定 ◆ 取消

输入参数名称和参数值后点击确定按钮。参数值可以使用嵌入式表达式。

编辑

选择参数,双击鼠标或按下回车键,系统打开参数修改窗口,修改后点击确定保存。

第 20 页 共 84 页

删除

选择参数,点击工具条件的删除按钮或按下删除键。

导入内置参数

XML 参数 大纲
🔁 🕂 🥒 🗙 🤣 🧃
 ₩14 ① ③ 公共参数 □ ③ 公共参数
≤选择
_xdata,XML数据
_xdataurl,XML数据URL
_scale,显示比例
_format,格式
_charset,字符集
_language,语言
_totype,目标类型
_pnt,打印机
_urito,输出URL
✓ 确定 ◆ 取消

数据库连接

新建

选择数据库连接结点后点击添加按钮或点击弹出菜单的添加菜单项



🌺 创建连接			×
HGNET	-		i
HGLOC		名称:	ORACLE
ORACLE	- 100	驱动:	oracle.jdbc.driver.OracleDriver
DB2	10000	URL	oc:oracle#nin:@localhost:1521:ora
SQLSERVER	335	用户:	scott *
SYBASE		口令:	TIGER
MYSOL	-		/
	,	🗸 确定	◆ 取消

左面为 JDBC 连接配置,选择后对应的连接方法会在右面对应的输入项中显示,辅助创建。该配置使用 conf/jdbc.xml,可以打开后手动编辑。属性可以使用嵌入表达式。

编辑

选择数据库连接,双击鼠标或按下回车键,系统打开数据库连接修改窗口,修改后 点击确定保存。

删除

选择数据库连接,点击工具条件的删除按钮或按下删除键。

3.4.3. 大纲

显示当前 XDOC 的大纲结构,大纲是根据段落的标题属性生成的。 点击大纲结构树的节点,可以定位到大纲对应的段落。

3.5.辅助设计区

3.5.1. 数据选择

系统为简化输入操作,提供了数据项选择向导,该向导为编辑器和图形编辑对话框 共用。为提高性能,数据项选择窗口的内容不会自动刷新,在内容改变后需要选定结点 手动刷新,如下图:



图形元素下会显示属性结点,有子图形的会显示子对象结点,对记录集合循环框和 图表循环框,会显示出字段,如下图:



第 23 页 共 84 页

数据选择有三个选项:方框、引用和标签。 ◆ 方框:表示拖拽产生的数据项自动用方框包起来
◆ 引用:表示数据项用表达式引用包起来
◆ 标签:表示使用数据项的名字而不是内容
下面是不同选项,产生不同结果的例子:

☑ 方框 □ 引用 ☑ 标签
□ 方框 I 引用 □ 标签
, ' 帜1⊻')%} (%v (e25' , ' 姓名')%} {%v (' e25' , ' 编号')%}

3.5.2. 格式库

可以将常用的格式添加到格式库中,提高开发效率。格式库的文件格式为 xdoc 或 zdoc。

打开格式库

点击打开按钮,弹出文件选择框,选择格式库,也可以输入新文件名,创建一个格式库。

数据 格式 元件 图形		
) 🕞 🕂 🗶 🖉 🛷		
查找:	R	
🤌 格式库:F:\a.xdoc		

添加格式

数据 格式 元件 图形 P || 🖙 🖶 🌉 🥔 🚿 А ₩. 查找: 凰 ۵ 按钮 1.2 📄 格式库:F:\a.xdoc ٠ ? B

选中图形,点击格式卡片页中的添加按钮,如下图:

设计器弹出名称输入对话框,输入名称,点击确定。

输入			×
0	名称:		
•	按钮		
		确定取消	

格式添加成功,如下图:

/数据〉格式 \图形 \元件 \
📱 🕂 🗶 🔎 😫
 → 格式库 按钮 ☆ 伝钮 ☆ 伝钮 ☆ 伝磁 ☆ 表格浅色 ☆ 表格深色
□ 醒目 按钮

可以输入用"/"分割的名称,形成树形结构,如输入:自定义格式/按钮,效果如下:

〈数据〉格式 \ 图形 \ 元件 \
🏽 🕂 🗶 🖉 😫
🗁 格式库
│ □ 标签
白… 🗁 自定义格式
└── 🗋 按钮
┃
└ □ 醒目
(
按钮

第 25 页 共 84 页

设置格式



选择图形,用鼠标双击格式,或点击设置格式按钮,如下图:

3.5.3. 图形库

添加图形

选中图形点击图形卡片页中的添加按钮,如下图:



系统弹出名称输入对话框,输入名称,点击确定。

输入		×
?	名称 圆柱体	
	确定 撤消	

图形添加成功,如下图:

数据(格式)图形(元件)			
88 🕂 🗙			
名称	预览		
圆柱体			

插入图形

在图形库中选择图形,用鼠标拖拽到编辑器中,如下图:



图形可以设置旋转角度:





3.5.4. 元件库

添加元件

选中图形点击元件卡片页中的添加按钮,如下图:



系统弹出名称输入对话框,输入名称,点击确定。

输入		×
2	名称	
•	交叉元件	
	确定 撤消	

元件添加成功,如下图:

数据(格式(图形)元件)			
🗄 🕂 🗶			
名称	预览		
交叉元件			

插入元件

在元件库中选择元件,用鼠标拖拽到编辑器中,如下图:



3.6. 属性编辑器

🖗 🗸 🤣		
名称:	name	
	☆ 位置	
顶部:	104	
左边:	188	
宽度:	260	
高度:	32	
尺寸类型:	固定 🗸 👻	
垂直对齐:	中间 🔻	222
	🕂 边线	
	☆ 埴充	
填充色:	#c0c0c0	
填充图片:		

编辑当前选中图形或当前 XDOC 的属性。

工具栏

标准的工具栏中有两个按钮,确定和刷新,属性编辑完成后要点击确定按钮,刷新 按钮用于将当前选中图形与编辑器不同的属性填充到编辑器中。

特殊图形如:记录集循环、图像等有特殊按钮,请参加组件工具栏中的说明。

分组按钮

编辑器中有 6 个分组按钮: 位置、边线、填充、滤镜、动画、其它。点击可以收缩 或展开组内的属性。

第 29 页 共 84 页

位置

	▶ ☆ 位置
顶部:	48
左边:	48
宽度:	92
高度:	92
边距:	4
相对文字:	文字上方 💽
排版方向:	横排・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・
旋转:	<u> </u>
图形旋转:	<u> </u>
变形:	
尺寸类型:	
垂直对齐:	(顶部 📃 👻

图形的布局位置。

尺寸类型:图形运算时调整大小的方式。 垂直对齐:图形中的文本垂直对齐方式。

边线

	🜔 🗘 边线	\supset
线型:	0	•
线宽度:	0.5	•
线图片:] 🗁
线颜色:	#000000	
斜线数:	0	•
角弧长:	0	•
边框线:		

线型:用逗号分割的点化线,0表示直线,如:



线图片:用图片代替颜色画线,值可以为图片文件路径、Internet图片 url,也可以指向其他组件,用该组件来绘图。指向其他组件语法格式为冒号加名称,如::pic。

第 30 页 共 84 页



边框线:设置仅显示部分边线,值用逗号分割,1表示显示,0表示隐藏。仅在角 弧长设置为0情况下有效。可以点击后面的按钮弹出编辑对话框编辑,如下图:

<u> 多</u> 边框线	×
反选	
▲ 一	

填充



填充色:图形的填充色,颜色值为 RGBA。点击后面的按钮可以弹出颜色选择对话框,如下图:

🔊 颜色选择	×
样品(S) HSB RGB RGBA	
红 0 85 170 255 252	
绿 0 85 170 255 102	
蓝 0 85 170 255	
Alpha 0 85 170 255 168	
3710	
御定 取消 重设(R)	

可以设置 Alpha 值产生半透明效果。



填充图片:用图片代替颜色填充,值可以为图片文件路径、Internet图片 url,也可以指向其他组件,用该组件来绘图。指向其他组件语法格式为冒号加名称,如::pic。

	*	*	*	*	*	*
*	8	*	*	*	*	*
	*	*	*	*	*	*
	*	*	*	*	*	*

渐变类型:使用填充色进行渐变填充。无填充色时无效。

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

渐变填充色: 第二渐变色。

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

滤镜

	☆ 滤镜
滤镜:	透明渐变 🚽 👻
滤镜参数:	

使用滤镜可以为图形增加艺术效果,目前支持26种滤镜。



部分滤镜有参数:透明渐变滤镜的参数为渐变类型,与填充色的渐变类型相同;旋转滤镜的参数为旋转角度;对比度滤镜的参数为对比度的值,0~100;阴影滤镜的参数 为阴影与原图形的距离,只有底色为白色才会产生效果。

可以对子图形设置滤镜后,再给父图形设置滤镜,产生滤镜叠加效果。下面的图形

是在对图形文字加光照滤镜后,对父图形加镜子滤镜后产生的效果。



动画

只有 jpd 格式支持

其它

	↔ 其它		
HTML超链接:			
可见:		•	

HTML 超链接:转为 HTML 时的超链接。可见:是否显示,true显示,false 隐藏。

特殊

复杂元件有自己的特殊属性,详见组件工具栏。

3.7.设计视图

3.7.1. 文档标签

─新文档1 \段落文字.xdoc	奥运城市(多语言).xdoc	\滤镜.xdoc∖

每一个编辑的文档都对应一个标签,标签的名字是文档的文件名简称,从远程服务器中 打开的文档名称以"\$:"开头,点击标签可以将相应的文档切换为当前编辑文档。可以双击 标签页关闭,也可以点击主工具条的关闭按钮或使用快捷键 Ctrl+F4,如果待关闭的文档没 有保存,会弹出保存提示对话框。

3.7.2. 字体工具栏



设置选中文本、光标所在文本、选中图形中的文本的字体属性。

3.7.3. 段落工具栏



设置选中文本、光标所在文本、选中图形中的文本的段落属性。

3.7.4. 排列与叠放层次

排列



选择图形



选择参照图形,如下图中的1号



点击排列图形工具条中的排列按钮:

◆ 水平排列



◆ 垂直排列



◆ 相同高度


叠放层次

设置图形的叠放层次,图形的叠放层次决定了运算的先后顺序,底层的最先运算。设计

第 37 页 共 84 页

过程中顶层的图形会遮住底层的图形。

3.7.5. 组件工具栏



插入状态

在插入状态下绘制的图形会作为段落中的元素插入到段落中,否则为浮动状态。

连续绘图

在非连续绘图状态下,绘图完成后,鼠标会变为选择状态,否则会保持鼠标绘图状态, 直到将选择了选择状态

线

选择直线画笔,在编辑器中按下鼠标,拖拽到目标点释放。直线刚绘制完为"\" 状,可以跳转大小到相反方向,转为"/"状,如下图:



箭头类型有5种:无、斜线、三角、圆、菱形



第 38 页 共 84 页

椭圆

椭圆可以设置开始角度和角度数形成弧。弧类型有三种:圆弧、弓形、扇形。



方框

方框是最基本的图形, 方框中可以输入文本也可以放入其他图形。



多边形

多边形

路径

路径可以表示任意复杂的图形。 路径工具栏



选择编辑器内置路径



调用路径编辑器编辑路径



基于图片生成路径



横向翻转 纵向翻转

图形文本

图形文本是以图形方式来绘制文字。使用它可以实现很多特殊效果,如空心字等。可以 设置线形、填充等属性。结合字体,可以绘制很多漂亮的字体图形。字体可以为系统字体名称,也可以直接外挂字体,外挂字体语法为":字体文件 URL",例如::c:/a.ttf,外挂字体必须为 TrueType 类型。



条件框

当条件满足时,在运算后的文档中会出现。



第 41 页 共 84 页

计数循环框

扩展方向:	纵向 👻
开始值:	1
结束值:	5

循环框的运行结果是(结束值一开始值)个方框,循环框对象的属性"i"表示当前的循环指针,下面是计数循环框的设计和运行例子:



上面循环框的扩展方向是纵向,横向效果如下:



扩展方向设置在嵌入状态下无效,会横向排列。

条件循环

扩展方向:	纵向 👻
条件:	1=2

在条件满足的情况下,循环输出多个方框,注意如果条件恒等会导致死循环。

记录集循环

🏽 🗸 🧇 🥸 🖬 🔲

第 42 页 共 84 页

记录集:	-
数据库连接:	-
	select * from 员工
查询SQL:	

可以点击查询设计对话框,构造查询 SQL,查询设计对话框会根据数据库连接不同而不同,详细参加创建数据集。

查询 sql 输入完成后可以点击布局向导按钮,系统会查询出查询 sql 结果中字 段列表,显示如下图:

🏄 布局向导				×
布局类型:	卡片			•
	标签	宽度	统计	
	编号	75		•
宁 码。	姓名	75		
于权;	职位	75		33
	上级	75		
	出生日期	75		•
格式模板:	常用			•
✓ 确定				

布局类型有五种 卡片型效果

编号	{%v('e47', '	
姓名	{%v('e47', '	
职位	{%v(°e47', '	
上级	{%v('e47', '	
出生日期	{%v(°e47', '	
	•	

第 44 页 共 84 页

字段循环显示的是查询 sql 结果中的字段,字段循环的唯一字段为"name",

¶%vr(″e15523′

字段循环

0001	王东	职员	0013	1980-02-05
0002	李海卫	销售	0006	1979-08-15
0003	张空	销售	0006	1982-08-07
0004	孔兵	经理	0009	1982-03-01
0005	刘龙维	销售	0006	1982-10-13

47' a47'	v	
INAL CALL	×	

自动型效果:

可以引用输出,效果如下:

0001	王东	职员	0013	1980-02-05
0002	李海卫	销售	0006	1979-08-15
0003	张空	销售	0006	1982-08-07
0004	孔兵	经理	0009	1982-03-01
0005	刘龙维	销售	0006	1982-10-13

| {%v (* e155* |
|--------------|--------------|--------------|--------------|--------------|
| | | | | |

	• •				
{%v (* e155'	{%v (* e155*	{%v (* e155*	{%v (* e155*	{%v (* e155'	

列表型效果					
{%v (* e155*	{%v (* e155*	{%v (* e155*	{%v (* e155*	{%v (* e155'	

动 主 刑 赤 田

编号	0001
姓名	王东
职位	职员
上级	0013
出生日期	1980-02-05
编号	0002
姓名	李海卫
职位	销售
上级	0006
出生日期	1979-08-15
编号	0003
姓名	张空
职位	销售
上级	0006
出生日期	1982-08-07

编号

编号
姓名
职位
上级
出生日期

分页记录集循环

记录集:		•
数据库连接:		•
	select 姓名,工资	fro
查询SQL:		
	 3333333 	
毎页记录数:	5	

输入查询 SQL 和每页记录数后,可以点击布局向导按钮,自动生成布局,如下图:

■ 姓名	工资 🔫	— 标题
{%v (′e25′,	{%v (* e25' 🔫	——列表
{%a('e12','i	O '7' 😽	──页号

运行效果如下:

姓名	工资		
王东	800		
李海卫	1600		
张空	1250		
孔兵	2975		
刘龙维	1250		
1,	/3		
姓名	工资		
张云	2850		
王芳	3000		
张凯	1500		
宋建	1100		
李保	950		
2,	/3		
姓名	工资		
刘海星	3000		
刘固	3000		
关心	2450		
刘程飞	5000		
3/3			

自动布局里面生成了一个记录集循环框,记录集循环框里面的查询 SQL 引用了分页记录集循环的 pageRowSet 属性,pageRowSet 表示当前页的记录集,分页记录集循环的 i 属性表示当前页,n 属性表示总页数。

循环展示部分也可以使用其它元素,例如图表,只要记录集引用 pageRowSet 属性即可。如下图:



交叉表

交叉表是一个功能强大的数据展示组件。支持上边、左边和统计数据多分组。数据 统计部分还可以使用自定义聚合函数。

记录集:		•
数据库连接:		•
查询SQL:	select * from 员工	
左分组项:	部门	_
上分组项:	职位	
统计项:	sum (工资) 工资	

输入查询 SQL、分组项和统计项, 左分组项和上分组项不能全部为空。点击确定 后运行文档结果如下:

第 47 页 共 84 页

	主管	总经理	经理	职员	销售	顾问	合计
10		5000	2450			3000	10450
20	6000		2975	1900			10875
30			2850	950	5600		9400
合计	6000	5000	8275	2850	5600	3000	30725

可以点击组件编辑对话框中的布局向导按钮自动生成展示框,如下图:



此时可以随意定义格式,和内部展示的内容,如下图:

E	取位 部门	{%v (* e163 14')%}	合计	
	{%v (* e1631	{%v (* e163		
	合计			

运行结果如下:

即位 部门	主管	总经理	经理	职员	销售	顾问	合计
10		5000	2450			3000	10450
20	6000		2975	1900			10875
30			2850	950	5600		9400
合计	6000	5000	8275	2850	5600	3000	30725

也可以只设定左分组,或上分组,如下图:

查询SQL	select * from 员工
左 分 組项:	部门,职位
上分組項:	
统计项:	sum(工资) 工资

运行效果如下:

10	总经理	5000
	经理	2450
10	顾问	3000
	小计	10450
	主管	6000
20	经理	2975
	职员	1900
	小计	10875
	经理	2850
30	职员	950
50	销售	5600
	小计	9400
合计	30725	

说明:

● 模板机制

交叉表运算过程中,以模板机制来生成结果。模板是以元素名称来识别的,左上角 元素名称为lt,左边分组元素名称为lg1...,统计项名称为ls1...,例如:第一组为lg1, ls1,第二组为lg2,ls2;上边分组元素名称为tg1...,统计项名称为ts1...,中心数据项 的元素名称为d1...。如果没有对应的元素,自动生成。

● 数据引用

交叉表元素的值为当前单元格的内容,在单元格中可以取得左分组和上分组项的 值。如上例子中 v('e16314'),将取得当前单元格的内容,v('e16314','部门')将取得部门 左分组的值。

分组表

分组表用于展示单个数据集,它可以对记录集中的数据安装某个字段进行分组统计,最 后一组还可以显示明细。

记录集:	
数据库连接:	•
	select 职位,编号,姓4
查询SQL:	
分组项:	职位
统计项:	รum (工资) 合计

输入查询 SQL、分组项和统计项,点击布局向导自动生成展示项,如下图:

第 49 页 共 84 页

<u>雇员分组表</u>

职位	{%v (* e2287		
	编号	姓名	工资

	{%v (″e2287	{%v (″e2287	{%v (* e2287
小计	{%v (′e2287		
合计	{%v (′e2287		

运行效果如下:

雇员分组表

职位	主管		
	编号	姓名	工资
	0008	王芳	3000
	0013	刘海星	3000
小计	6000.0		
职位	总经理		
	编号	姓名	工资
	0009	刘程飞	5000
小计	5000.0		
职位	经理		

说明:

分组表运算过程中,以模板机制来生成结果。模板是以元素名称来识别的,总分组 头元素名称为 gh,总分组尾元素名称为 gf,分组头元素名称为 gh1...,分组尾元素名 称为 gf1...明细项名称为 d。如果没有对应的元素,不生成。与交叉表的模板机制不同。

表格

表格元件用于设计复杂的表格式数据展示。它的每一个单元格都可以设置一个查询,进 行横向或纵向扩展。



图片

绘图类型:	拉伸 ▼
路径/wrl:	

绘图类型有三种:拉伸、重复、居中。



路径/url 可以为图片文件路径、Internet 图片 url,也可以指向其他组件,用该组件来绘图。指向其他组件语法格式为冒号加名称,如::pic。



图片以白色为透明色,填充色和填充图片会在白色部分显示出来。



可以点击转为内置按钮将图片保存到文档中。



图表

记录集:	-
数据库连接:	-
查询SQL:	
标题:	
类型:	饼图 ▼

图表类型有饼图、柱图、线图等。可以点击查询设计按钮构造查询 SQL 也可以手工输入。

数据识别规则如下:

- 一列: 行号作为名称列, 数据列作为值列。
- 两列: 第一列作为名称列, 第二列作为值列。
- 两列以上: 第一列作为分组列, 第二列作为名称列, 第三列作为值列。



填充图和填充色对图表有效。

特殊图表:

▶ 甘特图

第一列项目名称,第二列开始日期,第二列结束日期,第三列完成情况(0~1)

▶ 仪表盘

只处理一条记录,第一列为值,第二列为最小值,默认为0,第三列为最大值,默 认为100,第四列为单位

子文档

路径/w1:	rpt/月历.xdoc 🕞
参数:	

路径/url

当前文档的参数会自动传递到子文档中,当前文档中没有的参数,需要填写在参数属性中,语法格式:参数名=参数值,多个参数用";"分割

{%to_cnum (p (' month'))%}月	1 文档
two two two 月历文档	 ♀- □ 参数 □ month □ year 参数定义 ●- □ 数据库连接 ●- □ 数据集



扩展组件

内置扩展元件有:条形码、SWING、CLOCK、JPD、EMF



第 53 页 共 84 页

文本	
标签	
	$\langle \cdot \rangle$
大文本	
••	
确定 取消	

3.7.6. 图形工具栏

固定	•		۵	#000000 🚊		<u> </u>		• 0	-0	•
	设置当	自前图形或	当前图	日形中选中	国形的属	禹性。				

3.7.7. 辅助工具栏



- 元件类型转换
- 置为多语言标签

可以将选中的框置为多语言引用方式。如文本为"报表",将会置为:"{%t('报表')%}"。

● 隔行变色

点击会后弹出设置对话框,如下图:

🏄 隔行麥色형	置	×
循环项名称:	e6532	
奇行颜色:	#fffff	<u>.</u>
偶行颜色:	#f0f0f0	
	✔ 确定 ◆ 取消	

设置后运行效果如下:

0001	王东	职员	0013
0002	李海卫	销售	0006
0003	张空	销售	0006
0004	孔兵	经理	0009
0005	刘龙维	销售	0006

- 内容居中
- 清除背景
- 换页
- 特殊符号
- 设置文本超链接
- 设置沿路径文字

3.7.8. 设计背景、前景

背景内容将会输出在每一页下方,前景内容会输出在每一页的上方。 点击编辑背景按钮,进入背景编辑状态,如下图:

			- 10)	•	B	I	U	Ŧ	s
	新5	之档1]								
* 🖉 /		编辑背	<mark>景</mark>							
0			_							

可以在背景编辑器中输入文字或绘制图形等,可以为背景编辑器设置填充属性填充

色、渐变、填充图作为文档的背景。 背景编辑完毕后点击编辑背景按钮后回到内容编辑状态。 同样方式设计前景内容。

3.7.9. 选择图形

拖动鼠标选择图形,选择时按下 shift 键,已经选中的变为非选中状态,未选中的变为选中状态。也可以按下 shift 键后,点击要选择的图形,选择完毕后,点击空白处(不要放开 shift 键)。

3.7.10. 复制粘贴

• 从编辑器中复制的内容粘贴其它元素中,内容和格式不会丢失。粘贴到其它应用程

第 55 页 共 84 页

序中,有文本、HTML 和图片。用 Ctrl+c 进行复制,用 Ctrl+v 进行粘贴。

- 从其它应用程序中复制内容,粘贴到编辑器中,识别文本、HTML 和图片,粘贴进 来的图片会内容会保存在 XDOC 中。
- 从操作系统中复制文件,图片文件会粘贴为图片框,Excel文件会粘贴为表格,其 它文件会将文件内容粘贴进来。

3.7.11. 智能输入

- 输入"{%"后,编辑器会自动输入"%}"
- 输入"#"后,编辑器会自动输入"#"
- 输入""后,编辑器会自动输入""
- 输入"("后,编辑器会自动输入")"
- 输入"\$对象名."后,如果该对象为数据集对象,编辑器会弹出字段选择菜单,如下图:



● 输入"\$p."后,编辑器会弹出参数选择菜单,如下图:

₹%\$p	.%}	
	_language	I.
	_palette	R

● 输入"\$a.对象名."后,编辑器会弹出属性选择菜单,如下图:



● 输入"\$s."后,编辑器会弹出数据集选择菜单,如下图:



3.8.运行视图



点击运行标签页,进入运行视图,运行视图中显示当前文档的运行结果,运行视图中的 文档是只读的,在运行视图中保存、另存为、打印操作是针对结果文档的。



3.9. 查看 XML

固定		= •
🥸 设计(D)	🜔 运行(R)	x 查看⊻ML

点击查看 XML,切换到 XML 视图,如果当前视图是设计视图,显示编辑 XDOC 的 XML, 如果是运行视图,则显示结果 XDOC 的 XML。

3.10. 其它

3.10.1. 快捷键

- Ctrl+a: 全选
- Ctrl+c: 复制
- Ctrl+x: 剪切
- Ctrl+v: 粘贴
- Ctrl+z: 撤销
- Ctrl+y: 重做

- Ctrl+方向键:移动父图形中选择图形的位置
- Ctrl+Shift+方向键:调整父图形中选择图形的大小
- Ctrl+鼠标:移动选择的图形的位置
- Alt+鼠标:剪切选择的图形,粘贴到目标图形中
- Alt+Ctrl+鼠标:复制选择的图形,粘贴到目标图形中
- Ctrl+delete: 从父图形中删除选择内容
- Ctrl+n: 新建
- Ctrl+o: 打开
- Ctrl+s:保存
- F12: 另存
- Ctrl+f4: 关闭
- Ctrl+p: 打印
- Ctrl+e: 打开文本编辑框

3.10.2. 多语言

点击主工具栏的公共多语言翻译按钮可以编辑功能的多语言库,公共多语言库默认保存在 conf/docLanguage.xml 中。

文	件但	Ð	编辑	(<u>E</u>)	Ţ	具Œ	2	帮助	(H)	
	Ľ	⊳			8		۵		r	9

点击文档的多语言翻译按钮,可以编辑当前文档的多语言库,文档的多语言库保存在文档的 resource 标签中。



点击辅助工具栏的置为多语言标签按钮,可以将选中的框置为多语言引用方式。如文本为"报表",将会置为:"{%t('报表')%}"。

多语言编辑器,如下图:



首先点击添加按钮添加语言,然后添加字符串,可以点击导入字符串按钮,将当前文档 中所有的多语言引用标签直接导入,方便编辑,导入功能不会覆盖已有的字符串。 XDOC 的最终显示语言由参数_language 决定。例如:



4. 文档语法

4.1. 文档语法

XDOC 中除了 paper 元素属性和元素名称不能使用表达式引用外,其它所有元素和属性都可以使用,该语法有 3 种。

4.1.1. 嵌入表达式

嵌入表达式以"{%"开始,以"%}"结束,引擎处理时会计算表达式部分,并将结果与非表达式部分结合输出。例如: H{%1+1%}g,执行结果为: H2g

可以有多个嵌入式表达式共存。例如: H{%1+1%}g{%2+3%},执行结果为: H2g5 表达式中可以包含 SQL 函数,存储过程等,例如: H{%abs(-23) + 5%}g,执行结果为: H28g。详细的语法见《SQL 参考手册》。XDOC 扩展了 5 个函数,用于得到文档中信息。 doc_param 用于获取参数值,doc_att 用于获取元素属性,doc_val 用于获取记录集循环等元 素字段值,doc_set 用于获取记录集引用,doc_qry 可以指定连接执行任意查询 sql。这些函 数都可以用在表达式中,它们的第一个参数都是名称,引擎根据名称查找对象,如果有同名 的元素,用第一个。

4.1.2. 嵌入程序块

嵌入程序块以"{%!"开始,以"%}"结束,引擎处理时会执行程序块部分,并将结果 与非表达式部分结合输出。例如:H{%!a:=10;b:=20;c:=a+b;%}g,执行结果为:H30g。详细 的语法见《SQL参考手册》

注意: 在设计器中用嵌入程序块时,不要用回车,否则会被识别为分段。

数据集元素的 sql 属性中的 sql 语句的语法与指定的数据库连接有关系,指定了第三方数据库连接的 SQL 语句,语法必须符合第三方数据库。中间可以包含表达式引用,系统在处理完表达式引用候将结果提交给第三方数据库。未指定数据库连接的 SQL 语句,语法必须符合 HGSQL 语法,中间也可以使用表达式引用,并且可以直接使用。

4.1.3. 打印期表达式

打印期表达式夹在两个"#"中间,在打印绘制时计算,该表达式中有两个变量 pageno (页号)、pagecount(总页数)。表达式的语法为 hgsql 的数学表达式语法,可以使用内置函数,如:打印日期: #now()#。该表达式中不能引用数据库中的对象。

4.2. 文档函数

DOC_VAL

语法:

doc_val(objName, valName[, defValue]), 简写: v 功能: 取得对象的值,用于取得数据集当前记录的字段值 返回值类型: 字符串

DOC_ATT

语法: doc_att(objName, valName[, defValue]), 简写: a
功能: 取得对象的属性
返回值类型: 字符串

DOC_PARAM

语法:

doc_param(paramName[,defValue]), 简写: p

功能:

取得参数值

返回值类型:

字符串

DOC_SET

语法:

doc_set(setName[,query]),简写: s

功能:

取得数据集,setName可以是定义的公用数据集名称,也可以是数据集组件名称。可以将该函数返回的值用在 SQL 查询中作为 FROM 部分,

例如: select f1 from doc_set('set');

返回值类型:

数据集

DOC_QRY

语法:

doc_qry(connName, sql[,format]), 简写: q

功能:

指定连接,执行查询。connName 可以是定义的数据库连接名称,空表示使用 SQLHUB 连接。可以将该函数返回的值用在 SQL 查询中作为 FROM 部分

例如: select f1 from doc_qry('oracle', 'select 1 from dual'); format:返回值类型,可选值为: XML、TXT、ROWSET

返回值类型:

根据 format 返回不同类型

DOC_XSET

语法:

doc_xset(xpath),简写: xs

功能:

从 XDOC 指定 url 的 xml/html 中查询符合 xpath 的数据,返回记录集。只有一个字段\$XPATH

返回值类型:

记录集

DOC_XVAL

语法:

doc_xval([xpath,def]),简写: xv

功能:

从 XDOC 指定 url 的 xml/html 中查询符合 xpath 的数据,返回结点的文本或属性的值。

返回值类型:

字符串

DOC_TRANS

语法**:**

doc_trans(str),简写: t

功能:

多语言翻译,如果没有找到,返回原值

返回值类型:

字符串

4.3.函数简化引用

有4个文档函数可以以变量方式简化引用,	如下表:
---------------------	------

函数	示例		
doc_att	doc_att('F1','width')	\$a.f1.width	
doc_param	doc_param('param','v')	\$p.param.v	
doc_trans	<pre>doc_trans('str')</pre>	\$t.str	
doc_val	doc_val('SET', 'FIELD')	\$set.field	

由于 sql 语法不区分打小写,XDOC 中的对象名称等区分大小写,简化引用使用小写方式,不能对名称大写或包含特殊字符的对象进行引用。

4.4.计算过程

- ▶ 计算文本中包含的表达式,与非表达式文字合成新文字
- ▶ 计算图形元素,图形元素计算过程是递归进行的
- ▶ 计算过程取值的对象是源文档

4.5.数据集引用

- 数据集对象(数据集循环、字段循环、图表)每次执行时会强制重新查询数据, 被引用到的数据集对象也会强制取
- ▶ 通过 SQL 函数引用默认为非强制取,可以加参数强制取
- ▶ 为提高性能可以使用 SQL 函数取

- ▶ 取数据集合对象属性和值时不会重新查询
- ▶ 数据库连接使用优先级别:私有连接、共有连接、JNDI连接

5. 扩展开发

5.1.准备

项目中引入 hg.jar

hg.jar 可以在服务器部署包中找到,也可以直接下载,下载地址: http://www.hgsql.com/xdoc/lib/hg.jar

5.2.扩展元件

编写类继承自 com.hg.doc. ExtEleObj,覆盖 paint 方法或 getShape 方法,实现 getShape 方法后,在绘图时会根据属性设置,绘制图形、填充。

在 WEB-INF/conf/extele.xml 进行配置登记

下面是在 extele.xml 的配置示例:

配置的内容包括:

组件名(name),组件实现类(class),属性(attribute),属性可选值(option)。

5.3.扩展滤镜

实现接口 com.hg.doc.XImgFilter

```
package com.hg.doc;
/**
* 扩展图像过滤器
* @author wanghg
*/
public interface XImgFilter {
    /**
    * 过滤器名称数组,如果名称中包含",",则逗号前为名称,后为描述
    * @return
```

第 63 页 共 84 页

```
*/
public String[] getFilters();
/**
 * 过滤
 * @param img 图片
 * @param filterName 过滤器名称
 * @param param 参数
 * @return 结果图片
 */
public BufferedImage filter(BufferedImage img, String filter, String
param);
}
```

在 WEB-INF/conf/hg.conf 配置 doc.xImgFilter

5.4. 扩展图像库

```
实现接口 com.hg.doc.XImgLib
```

```
package com. hg. doc;
/**
* 扩展图片库
* @author wanghg
*/
public interface XImgLib {
   /**
    * 图片名称数组
    * @return 图片名称数组
    */
   public String[] getImgs();
   /**
    * 获取指定名称的图片
    * @param name 名称
    * @return 图片
    */
   public BufferedImage getImg(String name);
```

```
在 WEB-INF/conf/hg.conf 配置 doc.xImgLib
```

5.5.扩展图形库

实现接口 com.hg.doc.XShapeLib package com.hg.doc; /** * 扩展图片库

```
* @author wanghg
*/
public interface XShapeLib {
    /**
    * 图形数组
    * @return 图形数组
    */
    public Shape[] getShapes();
    /**
    * 图形字符串数组,字符串内容为路径表达式
    * @return 图形字符串数组
    */
    public String[] getStrShapes();
}
在 WEB-INF/conf/hg.conf 配置 doc.xShapeLib
```

5.6.扩展 XDATA 选择

```
实现接口 com.hg.doc.XDataSelector
package com.hg.doc;
/**
 * XML 数据选择器
 * @author wanghg
 */
public interface XDataSelector {
    /**
    * 选择 XML URL
    * @return
    */
    public String select();
}
```

在 WEB-INF/conf/hg.conf 配置 doc.xDataSelector

5.7.服务器监听

```
实现接口 com.hg.doc.ServerListener
```

```
package com.hg.doc;
/**
* 服务器监听类
* @author wanghg
*/
```

第 65 页 共 84 页

public interface ServerListener { /** * 请求前 * @param request * @param response * @throws Exception */ public void before(HttpServletRequest request, HttpServletResponse response) throws Exception; /** * 发生例外 * @param e */ public void onException(Exception e); /** * 处理完成 */ public void after();

在 WEB-INF/conf/hg.conf 配置 doc.ServerListener

5.8.打包

class 打包并签名, 放入 WEB-INF/lib 目录下 在配置项 doc.downJar 末尾加入 jar 名称, 以便设计器中可用

6. JPD 开发

JPD 是 Java Portable Document 的简称, 它以 Java2D 为基础, 有如下优点:

- 基于 J2SE1.4, 跨平台
- 自带阅读器,支持文档和幻灯片两种播放方式,支持多语言
- 目标文件小,通常是 pdf 的 1/5
- 页面式,减少分页带来的资源消耗
- 绘图指令优化
- 嵌入软字体,客户端不存在所需字体时,自动用软字体替代
- 可以包含图片、图形、文字
- 可以包含标题目录便于阅读检索
- 支持图形级超链接、工具提示
- 缩略图页面便于定位
- 可过滤打印文本,进行定位
- 可以包含标题、作者、公司等元数据信息
- 打印时根据用户纸张选择,自动缩放打印

第 66 页 共 84 页

- 支持 JS 方法调用、JS 绘图
- 支持命名文字运行期改变
- 可以转换为 jar、jpd、html、txt、png、jpg、gif 等多种格式

6.1.下载程序库

地址: <u>http://www.hgsql.com/down/jpd.jar</u>,在项目中引入

6.2.调用接口

6.2.1. JpdGraphics

package com.hg.jpd;

```
/**
* BookGraphics
* @author wanghg
*/
public class JpdGraphics {
   /**
    * 构造器
    * @param out 输出流
    */
   public JpdGraphics(OutputStream out) {}
   /**
    * 添加 Book
    * @param pageWidth 页面宽度
    * @param pageHeight 页面高度
    * @return
    */
   public Jpd addJpd(int pageWidth, int pageHeight) {}
   /**
    * 查看器类型: 文档
    */
   public static final int VIEWER_TYPE_DOC = JpdViewer.TYPE_DOC;
   /**
    * 查看器类型: 幻灯片
    */
   public static final int VIEWER TYPE SLIDE = JpdViewer.TYPE SLIDE;
   /**
    * 查看器类型: 图片
    */
```

```
public static final int VIEWER_TYPE_IMG = -1;
/**
 * 设置查看器类型
 * @param type 查看器类型
*/
public void setViewerType(int type) {}
/**
 * 设置查看器是否显示工具栏
 * @param visible 是否显示
*/
public void setViewerBarVisible(boolean visible) {}
/**
* 设置查看器是否显示导航栏
* @param visible 是否显示
*/
public void setViewerNavVisible(boolean visible) {}
/**
 * 设置查看器是否允许动画(幻灯片视图有效)
* @param visible 是否显示
 */
public void setViewerAnimateEnable(boolean enable) {}
/**
 * 设置查看器是否保持纵横比(幻灯片视图有效)
* @param visible 是否显示
*/
public void setViewerMAR(boolean mar) {}
/**
 * 设置是否带查看器
* @param visible 是否显示
 */
public void setWithViewer(boolean withViewer) {}
/**
* 关闭
 * @throws IOException 例外
 */
public void close() throws IOException {}
```

6.2.2. Jpd

package com.hg.jpd;

/**

* Java 便携式文档

第 68 页 共 84 页

```
* @author wanghg
*/
public class Jpd {
   /**
    * 动画类型:无
    */
                                   = Slide.type_none
   public static final int atype none
   /**
    * 动画类型: 随机
    */
   public static final int atype random = Slide.type random
   /**
    * 动画类型: 从左展开
    */
   public static final int atype_left
                                   = Slide.type_left
   /**
    * 动画类型: 从右展开
    */
   public static final int atype_right = Slide.type_right
   /**
    * 动画类型: 从上展开
    */
   public static final int atype top
                                       = Slide.type top
   /**
    * 动画类型:从下展开
    */
   public static final int atype_bottom = Slide.type_bottom
   /**
    * 动画类型: 盒子展开
    */
   public static final int atype_box
                                       = Slide.type_box
   /**
    * 动画类型: 横向百叶窗
    */
   public static final int atype_shutterx = Slide.type_shutterx
   /**
    * 动画类型: 纵向百叶窗
    */
   public static final int atype shuttery = Slide.type shuttery
   /**
    * 动画类型: 轮辐1
    */
   public static final int atype_convolute1 = Slide.type_convolute1 ;
   /**
```

```
* 动画类型: 轮辐2
*/
public static final int atype_convolute2 = Slide.type_convolute2 ;
/**
* 动画类型: 轮辐 3
*/
public static final int atype convolute3 = Slide.type convolute3 ;
/**
* 动画类型: 轮辐 4
*/
public static final int atype_convolute4 = Slide.type_convolute4 ;
/**
* 动画类型: 轮辐 8
*/
public static final int atype_convolute8 = Slide.type_convolute8 ;
/**
* 动画类型: 从左进入
*/
public static final int atype_move_left = Slide.type_move_left ;
/**
* 动画类型: 从右进入
*/
public static final int atype move right = Slide.type move right ;
/**
* 动画类型: 从上进入
*/
public static final int atype_move_top
                                    = Slide.type_move_top
/**
* 动画类型:从下进入
*/
public static final int atype_move_bottom = Slide.type_move_bottom;
/**
* 动画类型: 从左上进入
*/
public static final int atype_move_lt = Slide.type_move_lt
/**
* 动画类型: 从右上进入
*/
public static final int atype_move_rt = Slide.type_move_rt
/**
* 动画类型: 从左下进入
*/
public static final int atype move lb = Slide.type move lb
/**
```

```
* 动画类型: 从右下进入
*/
public static final int atype_move_rb = Slide.type_move_rb
/**
* 动画类型: 旋转
*/
public static final int atype_rotate = Slide.type_rotate
/**
* 构造器
* @param pageWidth 页面宽度
* @param pageHeight 页面高度
*/
public Jpd(int pageWidth, int pageHeight) {}
/**
* 设置是否嵌入字体
* @param b 是否嵌入字体
*/
public void setEmbedFont(boolean b) {}
/**
* 是否嵌入字体
* @return 是否嵌入字体
*/
public boolean isEmbedFont() {}
/**
* 作者
* @return 作者
*/
public String getAuthor() {}
/**
* 设置作者
* @param author 作者
*/
public void setAuthor(String author) {}
/**
* 公司
* @return 公司
*/
public String getCorporation() {}
/**
* 设置公司
* @param corporation 公司
*/
public void setCorporation(String corporation) {}
/**
```

```
* 备注
    * @return 备注
    */
   public String getDescription() {}
   /**
    * 设置备注
    * @param description 备注
    */
   public void setDescription(String description) {}
   /**
    * 标题
    * @return 标题
    */
   public String getTitle() {}
   /**
    * 设置总标题
    * @param title 标题
    */
   public void setTitle(String title) {}
   /**
    * 设置播放音乐,支持 mid 和 wav 格式
    * @param in 音乐流
    * @throws IOException 例外
    */
   public void setAudio(InputStream in) throws IOException {}
   /**
    * 添加标题(书签)
    * @param level 层次
    * @param text 标题文本
    * @param page 页面
    * @param x x 坐标
    * @param y y 坐标
    */
   public void addHeading(int level, String text, int page, int x, int y)
{}
   /**
    * 添加页
    * @return
    * @throws IOException
    */
   public Graphics2D addPage() throws IOException {}
   /**
    * 关闭
    * @throws IOException 例外
```
```
*/
   public void close() throws IOException {}
   /**
    * 显示到指定查看器中
    * @param viewer 查看器
    */
   public void setViewer(JpdViewer viewer) {}
   /**
    * 添加超链接
    * @param g graphics
    * @param url url
           格式 1: java 可识别的 url
    *
              例如: http://www.java.com
    *
           格式 2: javascript: js 代码,
    *
              作为 applet 嵌入在 html 中,会调用外面的 js 方法
    *
              javascript 可以简写为 js
    *
              例如: javascript:alert(1)
    *
           格式 3: #页码, x, y, 页面内定位
    *
              例如: #1, 20, 30
    *
    * @param shape 范围图形
    */
   public static void addHref(Graphics g, String url, Shape shape) {}
   /**
    * 添加提示
    * @param g graphics
    * @param tip 提示文本
    * @param shape 范围图形
    */
   public static void addToolTip(Graphics g, String tip, Shape shape) {}
   /**
    * 设置动画
    * @param g graphics
    * @param type 动画类型
    * @param order 播放顺序
    * @param length 动画时长 (秒)
    * @param times 播放次数, -1 表示循环播放
    */
   public static void setAnimate(Graphics g, int type, int order, int
length, int times) {}
   /**
    * 设置下一个 drawString 指令字符串名称
    * @param g graphics
    * @param name 名称
    */
```

```
public static void setStringName(Graphics g, String name) {}
```

6.2.3. JpdViewer

package com.hg.jpd;

```
/**
* JPD 查看器
* @author wanghg
*/
public class JpdViewer extends JComponent {
   /**
    * 查看器类型: 文档
    */
   public static final int TYPE DOC = 0;
   /**
    * 查看器类型: 幻灯片
    */
   public static final int TYPE_SLIDE = 1;
   /**
    * 构造器
    */
   public JpdViewer() {}
   /**
    * 设置是否显示工具栏
    * @param visible 是否显示
    */
   public void setBarVisible(boolean visible) {}
   /**
    * 设置是否显示导航栏
    * @param visible 是否显示
    */
   public void setNavVisible(boolean visible) {}
   /**
    * 设置命名文本的值
    * @param name
    * @param value
    */
   public void setText(String name, String value) {}
   /**
    * 页面宽度
    * @return
    */
```

第 74 页 共 84 页

```
public int getPageWidth() {}
/**
* 页面高度
* @return
*/
public int getPageHeight() {}
/**
* 设置页面大小
* @param width
* @param height
*/
public void setPageSize(int width, int height) {
   this.jpd.pageWidth = width;
   this.jpd.pageHeight = height;
}
/**
* 页数
* @return
*/
public int getPageCount() {
   return this. jpd. pageCount;
}
/**
* 设置页面大小
* @param width
* @param height
*/
public void setPageCount(int pageCount) {
   this.jpd.pageCount = pageCount;
}
/**
* 获取指定页面的画图板
* @return
*/
public Canvas canvas(int pageIndex) {
   Canvas canvas = this.jpd.getCanvas(pageIndex);
   if (canvas == null) {
       canvas = this.jpd.addCanvas(pageIndex);
   }
   return canvas;
}
/**
* 设置查看器类型
* @param type 查看器类型
```

```
*/
public void setType(int type) {}
/**
* 设置所属 frame
* @param frame 所属 frame
*/
public void setFrame(JFrame frame) {}
/**
* 打印
*/
public void print() {}
/**
* 打开
*/
public void open() {}
/**
* 打开指定 url 的 jpd
* @param url url
*/
public void open(String url) {}
/**
* 保存
*/
public void save() {}
/**
* 导航到
* @param page 页
* @param x x 坐标
* @param y y 坐标
*/
public void navTo(int page, int x, int y) {}
/**
* 关于
*/
public void about() {}
```

6.2.4. Canvas

package com. hg. jpd;

/**

* 绘图板

* @author wanghg

第 76 页 共 84 页

```
*/
public class Canvas {
   /**
    * 清空
    */
   public void clear() {
       codes.clear();
   }
   /**
    * 设置字体
    * @param name
    * @param style
    * @param size
    */
   public void setFont(String name, String style, int size) {}
   /**
    * 设置颜色
    * @param color 以#开头, 16 进制表示, RGBA, 例如: #f0c0a0ff, a 可以省略
    */
   public void setColor(String color) {}
   /**
    * 设置填充
    * @param paint 以#开头表示颜色,其它表示图片的 url 地址
    */
   public void setPaint(String paint) {}
   /**
    * 设置边框宽度
    * @param weight
    */
   public void setStroke(int weight) {}
   /**
    * 设置边框样式
    * @param weight 宽度
    * @param style 用逗号间隔的数字, 例如: 5,1,5
    */
   public void setStroke(int weight, String style) {}
   /**
    * 绘制文字
    * @param str
    * @param x
    * @param y
    */
   public void drawString(String str, int x, int y) {}
   /**
```

* 画线 * @param shape * @param x1 * @param y1 * @param x2 * @param y2 */ public void drawLine(int x1, int y1, int x2, int y2) {} /** * 绘制方框 * @param x * @param y * @param w * @param h */ public void drawRect(int x, int y, int w, int h) {} /** * 绘制圆角矩形,相同圆角 * @param x * @param y * @param w * @param h * @param arc */ public void drawRoundRect(int x, int y, int w, int h, int arc) {} /** * 绘制圆角矩形 * @param x * @param y * @param w * @param h * @param arcx * @param arcy */ public void drawRoundRect(int x, int y, int w, int h, int arcx, int arcy) {} /** * 绘制椭圆 * @param x * @param y * @param w * @param h

*/

public void drawEllipse(int x, int y, int w, int h) {}

```
/**
    * 填充方框
    * @param x
    * @param y
    * @param w
    * @param h
    */
   public void fillRect(int x, int y, int w, int h) {}
   /**
    * 填充圆角矩形,相同圆角
    * @param x
    * @param y
    * @param w
    * @param h
    * @param arc
    */
   public void fillRoundRect(int x, int y, int w, int h, int arc) {}
   /**
    * 填充圆角矩形
    * @param x
    * @param y
    * @param w
    * @param h
    * @param arcx
    * @param arcy
    */
   public void fillRoundRect(int x, int y, int w, int h, int arcx, int arcy)
{}
   /**
    * 填充椭圆
    * @param x
    * @param y
    * @param w
    * @param h
    */
   public void fillEllipse(int x, int y, int w, int h) {}
   /**
    * 绘制图形
    * @param strShape
    * @param x
    * @param y
    * @param w
    * @param h
    */
```

第 79 页 共 84 页

```
public void drawShape(String strShape, int x, int y, int w, int h) {}
/**
* 填充图形
* @param strShape 用路径指令描述的图形
* @param x
* @param y
* @param w
* @param h
*/
public void fillShape(String strShape, int x, int y, int w, int h) {}
/**
* 绘制图片, 缩放到指定大小
* @param url
* @param x
* @param y
* @param w
* @param h
*/
public void drawImage(String url, int x, int y, int w, int h) {}
/**
* 绘制图片,自动大小
* @param url
* @param x
* @param y
*/
public void drawImage(String url, int x, int y) {}
```

6.2.5. 示例

```
import java.awt.Color;
import java.awt.Graphics2D;
import java.awt.geom.Rectangle2D;
import java.io.FileOutputStream;
import com.hg.jpd.Jpd;
import com.hg.jpd.JpdGraphics;
/**
 * JpdGraphics 测试
 * @author wanghg
 */
public class Test {
    /**
```

```
* @param args
*/
public static void main(String[] args) {
    try {
       //创建 BookGraphics
       JpdGraphics jg =
            new JpdGraphics(new FileOutputStream("d:/test.jar"));
       //设置查看器属性
       jg.setViewerType(JpdGraphics.VIEWER TYPE DOC);
       jg.setViewerBarVisible(true);
       jg.setViewerNavVisible(true);
       //添加 Book
       Jpd jpd = jg. addJpd(400, 300);
       //设置属性
       jpd.setTitle("BookGraphics 测试");
       jpd. setAuthor ("wanghg");
       jpd.setCorporation("http://www.hgsql.com");
       jpd. setDescription("test");
       //添加页
       Graphics2D g = jpd.addPage();
       //绘图
       g. setColor(Color. cyan);
       g.fillRect(0, 0, 400, 300);
       g. setColor(Color. yellow);
       String str = "http://www.hgsql.com";
       g.drawString(str, 40, 150);
       Rectangle2D rect =
       g.getFont().getStringBounds(str, g.getFontRenderContext());
       rect.setFrame(40, 150 - g.getFont().getSize(),
           rect.getWidth(), rect.getHeight());
       g.setColor(Color.DARK_GRAY);
       g.draw(rect);
       //设置超链接
       Jpd.addHref(g, str, rect);
       //设置工具提示
       Jpd.addToolTip(g, "go:" + str, rect);
       //设置动画效果
       Jpd. setAnimate(g, Jpd. atype_left, 1, 1, 1);
       jg.close();
   } catch (Exception e) {
       e.printStackTrace();
   }
}
```

6.3. 注册为虚拟打印机

方法一:声明引用

Class.forName("com.hg.jpd.JpdPrintService");

方法二: 注入引用

例如:

原来启动程序方法为: java -classpath "a.jar" b

修改为: java -classpath "a.jar;jpd.jar" com.hg.jpd.JpdPrintService b

在程序中使用 javax.print.ServiceUI.printDialog 调用打印对话框时,会显示一个虚拟打印机,如下图:

≦打印	X
常规(G)页面设置(S)外观(A)	
打印服务 名称(N): HG Jar 虚拟打印机	▶ 属性(R))
状态: 接受任务 类型:	
信息:	打印到文件(E)
打印区域	份数
● 全部(L)	打印份数(②): 1€
○ 页码范围(E) 1 至 1	□比较(C)
	打印取消

6.4.结果使用

以名称为 test.jar 为例

6.4.1. 独立应用程序

直接运行或使用命令行: java -jar test. jar

6.4.2. 嵌入到 swing 应用中

在项目中引入 test.jar 作为类库,实例化 JpdViewer 添加到 swing 中 JPanel panel = new JPanel(); panel.add(new com.hg.jpd.JpdViewer());

6.4.3. 嵌入到网页中

```
<applet id=" jpd" archive=" file:///d:/test.jar"</pre>
     code=" com. hg. jpd. JpdApplet. class"
     width=100% height=100%></applet>
 可用参数:
 ● url: jpd 文件 url 地址
 ● lookAndFeel: 外观
          可选值:
              system: 操作系统; cross: 跨平台; ***. LookAndFeel: 类名称
 ● viewType: 查看器类型
          可选值:
              doc: 文档; slide: 幻灯片; img: 图片
 ● onload: 加载完成后自动调用的 js 方法
   可以在 js 中调用查看器的方法,例如:
       <script>
       jpd.width = jpd.viewer.getPageWidth();
       jpd. height = jpd. viewer. getPageHeight();
       jpd. viewer. about ();
       </script>
   jpd 为 applet 标签的 id 值,其它方法参考: JPD
   未签名的 applet 不能执行保存操作,解决方法是对 jar 进行签名,或直接使用
jpd. jar, 引用方式如下:
       <applet id=" jpd" archive=" jpd. jar"</pre>
          url=" file:///d:/test.jar"
       code=" com.hg.jpd.JpdApplet.class"
       width=100% height=100%></applet>
```

6.5. 画图板

调用步骤:

- 用 JpdViewer 的 setPageCount、setPageWidth、setPageHeight 定义页面
- 用 JpdViewer 的 canvas 方法获取指定页面的 canvas
- 调用 canvas 的方法绘图, canvas 包含适合 js 和 java 调用的 api
- 调用 JpdViewer 的 repaint 方法使结果显示

调用可以在 Java 应用程序中,也可以在网页中。

```
可以直接使用 jpd.jar,也可以使用结果的 jar。
绘图对显示、打印、保存为图片有效。
<applet id="jpd" archive="jpd.jar"</pre>
code="com.hg.jpd.JpdApplet.class"
width=100% height=100%></applet>
\langle \text{script} \rangle
var canvas = jpd.viewer.canvas(0);
jpd.viewer.setPageCount(5);
canvas.setColor("#ff0000");
canvas. setStroke(4, "5, 1");
canvas.drawEllipse(100, 100, 200, 200);
canvas.setColor("#0000ff66");
canvas.fillRoundRect(100, 100, 200, 100, 20);
canvas.setStroke(2);
canvas.drawShape("M 0 25 C 25 0 75 50 100 25 L 100 75 C 75 100 25 50 0 75 L 0
25 Z", 100, 300, 200, 200);
for (var i = 0; i < 5; i++) {
    canvas = jpd.viewer.canvas(i);
    canvas.setColor("#00ff00");
    canvas.setFont("隶书", "bold,italic", 40);
    canvas.drawString("第" + (i+1) + "页", 50, 100);
}
jpd. viewer. repaint();
</script>
```

6.6.浏览器快捷键

```
    公共

  F1:关于
  F2: 切换视图
  F3: 切换工具条显示
  F4: 切换导航条显示
  Ctrl+S: 保存
  Ctrl+O: 打开
  Ctrl+P: 打印
  HOME: 首页
  END: 尾页
● 文档视图
  -: 缩小
  +: 放大
● 幻灯片视图
  Left、Up、Backspace: 上一页
  Right、Down、Enter: 下一页
  Ctrl+G: 跳转到页面
```