Noka Laboratory

Noka of China Co., Ltd.

BS 研究中心

BS Research Center

Noka 标签技术白皮书

Noka art white book

文档类别	技术手册
版　　本	V6.0.14
密　　级	不秘密
起 草 人	谢方建
审核部门	
确认部门	

二〇一六年六月

使用申明

本标签库知识产权归诺亚(中国)开源科技所有，同时本标签遵循开源组织原则，源码公开，不对使用源码作任何限制，不对源码用途承担任何责任，提供技术交流平台，不提供技术支持.对成品组件的使用(既 Noka-x.x.jar)保留知识产权及追究使用责任的权力(对标签参与开发内部人员不在此限制内)。为了本标签更好的发展，对源码的任何重大修改请以邮件或其它方式通知开发组织。

Noka Laboratory
Rebin

网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com

目录

网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com

网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com

网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com

第1章 标签配置

1.1 Noka 6.0 新特性

注：Noka 6.0以后的版本集成了JDOM和JXL以及prototype1.6.3框架。测试环境为jdk1.5.0、tomcat6.0、WebLogic 9.2。要保证Noka标签的正常运行，你需要以下的运行环境。精简了重复加载的资源文件，更适合于异步操作的高性系统。

从6.0.1开始，con对像采用全局配置，推荐用con拦截器注入。

名称	版本	备注
JDK	1.5.0(以上)	
Tomcat	5.5(以上)	需要支持jsp2.0标准
WebLogic	9.2(以上)	
resin	4.0(以上)	

Noka 6.0在以下数据库方面采用了数据库本身的分页方法，支持海量数据分页，列表以外的数据库为普通分页算法，在海量数据处理上可能效率不高。也可以通过自定义数据处理引擎进行其它数据库扩展，详细见配置项dbgrid-dataserver的说明。

数据库	版本	备注
SQL Server	2000	
SQL Server	2005	
Oracle	9i(以上)	
MySQL	4.0(以上)	
PostgreSQL	7.0(以上)	
DB2	9.7(以上)	
amoeba	3.0	支持水平切分查询分页

Noka 6.0 在浏览器兼容方面做了大量的优化处理，以下列出了兼容的浏览器类型（均通过严格测试）。

浏览器名称	版本	备注
Internet Explorer	6.0(以上)	
Mozilla Firefox	2.0(以上)	
Netscape Navigator		
Opera		
Apple Safari		
Google Chrome		

1.2 基础配置

Noka 6.0 的基础配置非常简单，具体步骤如下：

1、将jar目录下面的noka-6.0.x.jar文件拷贝到工程目录的lib目录下面。

2、将configxml目录下的noka-config.xml、noka-tag-x.x.x.dtd文件拷贝到工程目录下的任意一个目录里面，一般建议拷贝在classes下面。

3、打开工程的web.xml文件，在开始处加入如下代码（一般建议将noka-tag拦截器作为web.xml的第一个拦截器）。

```
<!-- noka 拦截 -->
<filter>
    <filter-name>NokaTagFile</filter-name>
    <filter-class>org.nokatag.system.NokaTagFilter</filter-class>
<init-param>
    <param-name>config</param-name>
    <param-value>WEB-INF/classes/noka-config.xml</param-value>
</init-param>
</filter>
<filter-mapping>
    <filter-name>NokaTagFile</filter-name>
    <url-pattern>/*</url-pattern>
</filter-mapping>
```

4、在JSP页面开始处引入该标签。

```
<%@ taglib prefix="n" uri="/noka"%>
```

至此，Noka标签的配置完成了，祝你使用愉快。

1.3 高级选项

在noka-config.xml文件中可以对noka标签的高级选项进行进一步的配置。如下所示：

约定：${rootdir}为系统根目录，${rootpath}为系统虚拟根目录，其中${rootpath}可以在jsp页面上使用。

1、option选项说明

default-language为系统默认语言，它的值是标准的语言选项值，如简体中文（zh-cn）等。语言文件在noka-6.0.x.jar的lunguage目录里面，命名规则为language_语言种类.xml。

is-dbug表示noka标签是否以dbug模式运行，如果是dbug模式运行时，在后台会打印一些调试信息，以便调试程序。其值为{true|false}。在正式发布时建议以非dbug模式运行。

sql-checkvalue防SQL注入sql过滤关键字，用|分隔。

pass-key推荐每个项目按实际应用改成不同的值，用于加密关键数据。

db-connection-class数据连接接口实现类，noka-tag内置六种实现类，不设置该项，六个类分别为：

a、 org.nokatag.Connection.HibernateConnection

b、 org.nokatag.Connection.ProxoolConnection

c、 org.nokatag.Connection.DBJdbcConnection

d、 org.nokatag.Connection.IbatisConnection

e、 org.nokatag.Connection.SpringHibernateSessionFactory

f、 org.nokatag.Connection.SpringIbatisSessionFactory

可用自定义类代替，自定义类需要实现org.nokatag.Connection.DBConnectionInterface接口，如果是Hibernate实现方式，需要自自己维护session功能里，可以通过ServletNokaContext实现，在getConnection方法内调用ServletNokaContext.setHibernateSession()方法设置需要操作session，在closeConnection方法内使用ServletNokaContext.getHibernateSession()取出设备的session对象。

a)直接从Hibernate中获取数据库连接，默认为该实现方式。

b)从Proxool连接池中获取数据库连接，该方式适合采用了Proxool连接池而没有使用其它任何其它IOC框架的时候使用

c)从JDBC中获取数据库连接，通过db-driver，db-url，db-user，db-password数据库连接参数直接建立JDBC连接。

d)通过Ibatis建立数据库连接，要使用该方式需要配置ibatis环境，同时ibatis的配置文件约定名称为ibatis-config.xml，并且只能存放在classes目录下。

e)是在SSH架构中或是Spring框架中由Spring接管HibernateSessionFactory之后使用Spring的hibernateSessionFactory获取连接的操作类。需要配合spring-sessionFactory选项以指代Spring中配置的bean id属性，如Spring如下配置

```
<bean id="sessionFactory"
class="org.springframework.orm.hibernate3.LocalSessionFactoryBean">
    <property name="configLocation">
        <value>classpath:hibernate.cfg.xml</value>
    </property>
</bean>
```

以上配置在noka-config中相对应的配置为：

```
<option key="db-connection-class"
value="org.nokatag.Connection.SpringHibernateSessionFactory "></option>
```


网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com

```
<option key=" spring-sessionFactory " value="sessionFactory"></option>
```

其中spring-sessionFactory中的value对应于bean id属性(sessionFactory)

f)在Spring接管ibatis的SqlSessionFactory之后，采用Spring的ibatisSqlSessionFactory获取数据库连接的方法，使用方法同e，此时同样需要配置spring-sessionFactory以指代Spring中的bean id。

Spring配置实例如:

```
<bean id="sessionFactory" class="org.mybatis.spring.SqlSessionFactoryBean">
    <property name="dataSource" ref="dataSource" />
</bean>
```

上例所示的noka配置如下：

```
<option key="db-connection-class"
value="org.nokatag.Connection.SpringIbatisSessionFactory"></option>
<option key="spring-sessionFactory" value="sessionFactory"></option>
```

dbutil-auto-close-session在调用DataUtil时，执行完数据库操作以后是否关闭该连接（包括session），默认为false，不关闭。

encoding系统编码，默认为UTF-8。

proxool-alias使用proxool连接池时，连接池的上下文名称。

db-driver在使用JDBC连接时，使用的驱动名称。

db-url在使用JDBC连接时,数据库地址。

db-user在使用JDBC连接时,数据库用户名。

db-password在使用JDBC连接时,数据库密码。

user-filter用户自定义拦截器，该拦截为实现javax.servlet.Filter接口规范的拦截器，其代码编写规范同一般拦截器相同，多个拦截器按配置文件内的先后顺序以此执行，如果其中某一个拦截器没有调用do.Filter方法，侧剩下的拦截器将不在被执行。可以指定拦截器适用于某个路径，如

```
<option key="user-filter" value="org.noka.APPFilter">/app.fs</option>
//如上配置表示 APPFilter 这个拦截器只对 app.fs 这个路径启作用,noka 内部使用 startsWith 来选择路径，因此不能使用*等通配置符
```

rootpath-rule：系统中rootpath变量规则，可选值为1-3，1代表全路径（带http协议头），2代表系统根路径（相对目录），3不作处理，返回空字符，默认为1。

2、file-save选项说明

position当使用数据库存储时，数据表的名称，noka-tag会自动建表。

type存储类型，可选DB|DISK。

3、rewritlist选项说明

key为需要替换的关键字，在文档中的写法为${key}。

revalue为取得重设置时用的关键字。

type为取得重设置的类型，支持从request、session、const、paramenter四种方式。其中request是指Attribute方法指置的变量，session是指session内设置的变量,const直接用revalue值替换，paramenter是指request.getParamenter()获取的值。

4、cache-files选项说明

Cache是否缓存，可选值true|false。

prefix外部需要缓存处理的目录，以/开头，相对于工程目录，多个目录用逗号分隔。

jszip是否启用压缩，该压缩只对js文件起作用。

cszip是否启用压缩，该压缩只对css文件起作用。

skip例外（不需要缓存处理的文件），多个文件用逗号全隔。

extension文件后缀名，形如*.js。

网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com

expires客户端缓存期限，数字带单位，可选单位D表示天，h表示小时，m表示分，s表示秒。

5、dbgrid-dataserver选项说明

该选项传入一个类路径，该类用于全局的DBGrid控件SQL操作，需要继承org.nokatag.dbgrid.NDBGridData并在覆盖父类的NDBGridDataItem DataQuery(NDBGridDataQueryItem dq,Connection con)方法，在该方法内部con不能关闭，NDBGridDataQueryItem前端控件传回的参数对象。NDBGridDataItem是返回关端的数据对象。

实例：

```
//1、制定义数据库连接实现类(Hibernate 实现方式)
import java.sql.Connection;
import org.db.Connection.DBConnectionInterface;
import org.hibernate.Session;
import org.nokatag.system.HibernateUtil;
import org.nokatag.system.ServletNokaContext;

public class HibernateConnection implements DBConnectionInterface{
public HibernateConnection(){
    //初始化
}
/**
 * 获取连接
 */
public Connection getConnection() {
    Session hsession=HibernateUtil.currentNewSession();//获取 hibernate session
    ServletNokaContext.setHibernateSession(hsession);//如果非 hibernate 不需要这句
    return hsession.connection();//获取 connection
}
/**
 * 关闭连接
 */
public void closeConnection(Connection con) {
    try{
        if(null!=con){
            con.close();//关闭连接
```


网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com

```
     //关闭当前 session，非 hibernate 不需要这句
    HibernateUtil.closeSession(ServletNokaContext.getHibernateSession());
    }
        }catch(Exception se){
        }
    }
}
//2、自定义拦截器
import java.io.IOException;
import javax.servlet.FilterChain;
import javax.servlet.FilterConfig;
import javax.servlet.ServletException;
import javax.servlet.ServletRequest;
import javax.servlet.ServletResponse;
import javax.servlet.Filter;

public class OtherFilter implements Filter{
    @Override
    public void destroy() {
        //拦截器消费方法同普通 Filter
    }
    @Override
    public void  doFilter(ServletRequest request, ServletResponse
response,FilterChain chain) throws IOException, ServletException {
        chain.doFilter(request, response);//继续向下执行其它拦截器，如果不继续执行
侧不调用该方法
    }
    @Override
    public void init(FilterConfig arg0) throws ServletException {
        //拦截器初始化方法，同普通 Filter，可以通过 FilterConfig 读取 web.xml 里面的配
置信息
    }
}
//3 自定义 dbgrid 数据处理引擎(以 MySQL 为例)
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.ResultSetMetaData;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
```


网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com

```
import java.util.Map;

import org.nokatag.dbgrid.CellParItem;
import org.nokatag.dbgrid.NDBGridData;
import org.nokatag.dbgrid.NDBGridDataItem;
import org.nokatag.dbgrid.NDBGridDataQueryItem;
import org.nokatag.system.BugInfoWrint;

public class MySQLDataServer extends NDBGridData{
    public NDBGridDataItem DataQuery(NDBGridDataQueryItem dq,Connection con) {
        try{
            //-------加载查询条件-----------------------------------
            String sql=dq.getSql();//获取查询 SQL 语句
            if(dq.getWsql()!=null)//查询条件，已经格化成了?
                sql="select   * from ("+sql+")
temptable_"+System.currentTimeMillis()+" where 1=1 "+ dq.getWsql();
            else
                sql="select   * from ("+sql+")
temptable_"+System.currentTimeMillis()+" where 1=1 ";
            //------计算总记录数-----------------------------
            PreparedStatement pa= con.prepareStatement(" select count(1) as conet
from  ("+ sql + ")  temptable_" +
System.nanoTime(),ResultSet.TYPE_SCROLL_INSENSITIVE,ResultSet.CONCUR_READ_ONLY);
            for(CellParItem c:dq.getParList()){//设置查询条件
                pa.setObject(c.getIndex(),c.getValue(),c.getType());
            }
            ResultSet rsf=pa.executeQuery();
            rsf.next();
            Long lastRow = rsf.getLong("conet");// 一共有多少条记录
            rsf.close();//内部产生的 ResultSet 一定要关闭
            pa.close();//内部产生的 PreparedStatement 一定要关闭
            //---------获取当前页数据-----------------------------------
            Long showPage =dq.getShowPage();
            Long pageSize
=dq.getPageSize()==null?(lastRow<1?1:lastRow):dq.getPageSize();
            Long pageConut = (lastRow % pageSize == 0) ? (lastRow / pageSize): (lastRow
/ pageSize + 1);// 总的页数
            if (showPage > pageConut)
                showPage = pageConut;
            if (showPage < 1)
                showPage = 1L;
```


网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com

```
            Long posion = (showPage - 1) * pageSize;// 游标位置
            String selectSql = "select  * from ( "+sqlpx(dq,sql)+" )
table_"+System.currentTimeMillis()+" limit "+posion+","+pageSize+" ";
            PreparedStatement pa2=
con.prepareStatement(selectSql,ResultSet.TYPE_SCROLL_INSENSITIVE,ResultSet.CONCUR_R
EAD_ONLY);
            for(CellParItem c:dq.getParList()){
                pa2.setObject(c.getIndex(),c.getValue(),c.getType());
            }
            ResultSet rss = pa2.executeQuery();
            ResultSetMetaData rsmd = rss.getMetaData();// 取得列名
            List<List<String>> rows = new ArrayList<List<String>>();
            List<String> cellnames=new ArrayList<String>();//列名
            while (rss.next()) {//组织数据
                List<String> row=new ArrayList<String>();
                Map<String, Object> map=new HashMap<String, Object>();//列名=值
                cellnames=new ArrayList<String>();//列名
                for(int a=1;a<=rsmd.getColumnCount();a++){
                    cellnames.add(rsmd.getColumnName(a));//列名
                    String cellvalue= rss.getString(a);
                    map.put(rsmd.getColumnName(a), cellvalue);
                }
                for(int a=0;a<cellnames.size();a++){
                    String
value=formatValue(map.get(cellnames.get(a)),dq.getFormatcell(),cellnames.get(a),dq.
getRequest(),map);//格式化数据
                    row.add(value);
                }
                rows.add(row);
            }
            rss.close();//内部产生的 ResultSet 一定要关闭
            pa2.close();//内部产生的 PreparedStatement 一定要关闭
            //-----组织返回数据 pageConut:总页数,showPage:当前页码,posion+1:当前页
起始行,(posion+rows.size()):当前页结束行,Long.valueOf(rows.size()):当前页总记录
数,lastRow:共多少条记录,rows:数据,cellnames:列名数据
            NDBGridDataItem nd = new NDBGridDataItem(pageConut, showPage, posion+1,
(posion+rows.size()), Long.valueOf(rows.size()),lastRow,rows,cellnames);
            return nd;
        }catch(Exception se){
            se.printStackTrace();
            BugInfoWrint.Print(se);
```


网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com

```
        }
        return null;
    }
}
```

1.4 关于 JSON 数据支持说明

Noka 6.0.12及其以后的版本新增加了对JSON数据的支持，解析与生成引擎使用的是Google的gson引擎，为了不与其它冲突，所有包移植到了org.noka.gson下，其具体使用方法可参照gson官方文档。

Gson使用实例

```
//-------字符串到对象转换----------
{
 String s="{'name':'name0','age':0}";
 Gson gson = new Gson();
 Person person = gson.fromJson(s,Person.class);
 System.out.println(person.getName());
}
//------字符串到数组----------------
{
 String s="['aaa','bbbb','cccc']";
 Gson gson = new Gson();
 List person = gson.fromJson(s,List.class);
 System.out.println(person.get(0));
}
//------字符串到数组对象--------------
{
 String s="[{'name':'name0','age':0},{'name':'name0','age':0}]";
 Gson gson = new Gson();
 List<Person> person = gson.fromJson(s,new TypeToken<List<Person>>(){}.getType());
 System.out.println(person.get(0).getName());
}
//-------对象到字符串 JSON-------------
{
 Person p = new Person();
 p.setAge(39);
 p.setName("ccc");
 Gson gson = new Gson();
```

```
 System.out.println(gson.toJson(p));
}
//--------数组到字符串 JSON-------------
{
 List<String> a= new ArrayList<String>();
 a.add("aa");
 Gson gson = new Gson();
 System.out.println(gson.toJson(a));
}
//-------数组对象到字符串 JSON----------
{
Person p = new Person();
p.setAge(39);
p.setName("ccc");
List<Person> a= new ArrayList<Person>();
a.add(p);
a.add(p);
Gson gson = new Gson();
System.out.println(gson.toJson(a));
}
//---------------实例类-----------------------------
public class Person {
    private String name;
    private int age;
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public int getAge() {
        return age;
    }
    public void setAge(int age) {
        this.age = age;
    }
    @Override
    public String toString()
    {
        return name + ":" +age;
    }
```


网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com

```
}
```

在前端JavaScript层，引入了prototype-1.6.0.3框架，因此在noka标签内，可以使用该框架的内置方法进行JSON数据转换，任何JSON字符对象都可以调用evalJSON()方法将字符串对象转换成JSON对象，如将数组，javaScript对象转成JSON字符串侧可以调用toJSON()方法转成JSON字符串。

```
function test(a){
        var c = a.responseText.evalJSON();//在 ajax 回调方法中，转换成 js 对象
}
//-----------------------------------
var v=’[“aaaa”,”bbbb”,”csss”]’;
v.evalJSON();//字符串转换成 json 对象
//-----------------------------------
var a = [‘cccc’,’cc’];
a.toJSON();//对象转换成 JSON 字符串
```

1.5 关于 http 安全协议头配置

http安全协议配置根据需要可分目录控制，良好的好http安全协议头配置可有效抵御常见的http网络攻击。

```
<http-headers>
    <headers-public>
        <header key="x-frame-options">SAMEORIGIN</header>
        <header key="X-XSS-Protection">1;mode=block</header>
        <header key="X-Content-Type-Options">nosniff</header>
    </headers-public>
    <headers-pravate dir="/script" forpublic="false">
        <header key="x-content-security-policy">default-src 'self';</header>
    </headers-pravate>
</http-headers>
```

headers-public公共协议头，针对所有的路径启作用。

headers-private私有协议头，只对dir配置的目录启作用，forpublic指示是否继承public协议头，可选值true|false。

http安全协议头说明如下：

1、Strict-Transport-Security

HTTP Strict Transport Security，简称为HSTS。它允许一个HTTPS网站，要求浏览器总是通过HTTPS来访问它。现阶段，除了Chrome浏览器，Firefox4+，以及Firefox的NoScript扩展都支持这个响应头。

我们知道HTTPS相对于HTTP有更好的安全性，而很多HTTPS网站，也可以通过HTTP来访问。开发人员的失误或者用户主动输入地址，都有可 能导致用户以HTTP访问网站，降低了安全性。一般，我们会通过Web Server发送301/302重定向来解决这个问题。现在有了HSTS，可以让浏览器帮你做这个跳转，省一次HTTP请求。

要使用HSTS，只需要在你的**HTTPS**网站响应头中，加入下面这行：

strict-transport-security: max-age=16070400; includeSubDomains

includeSubDomains是可选的，用来指定是否作用于子域名。支持HSTS的浏览器遇到这个响应头，会把当前网站加入HSTS列表，然后在max-age指定的秒数内，当前网站所有请求都会被重定向为https。即使用户主动输入http://或者不输入协议部分，都将重定向到 https://地址。

Chrome内置了一个HSTS列表，默认包含Google、Paypal、Twitter、Linode等等服务。我们也可以在Chrome 输入chrome://net-internals/#hsts，进入HSTS管理界面。在这个页面，你可以增加/删除/查询HSTS记录。例如，你想一 直以https访问某网址，通过“add Domain”加上去就好了。

2、X-Frame-Options

X-Frame-Options，已经转正为Frame-Options，但现阶段使用最好还是带上X-。Chrome4+、Firefox3.6.9+、IE8+均支持，详细的浏览器支持情况看这里。使用方式如下：

x-frame-options: SAMEORIGIN

这个响应头支持三种配置：

网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com

DENY：不允许被任何页面嵌入；

SAMEORIGIN：不允许被本域以外的页面嵌入；

ALLOW-FROM uri：不允许被指定的域名以外的页面嵌入（Chrome现阶段不支持）；

如果某页面被不被允许的页面以<iframe>或<frame>的形式嵌入，IE会显示类似于“此内容无法在框架中显示” 的提示信息，Chrome和Firefox都会在控制台打印信息。由于嵌入的页面不会加载，这就减少了点击劫持（Clickjacking）的发生。

3. X-XSS-Protection

顾名思义，这个响应头是用来防范XSS的。最早我是在介绍IE8的文章里看到这个，现在主流浏览器都支持，并且默认都开启了XSS保护，用这个header可以关闭它。它有几种配置：

0：禁用XSS保护；

1：启用XSS保护；

1; mode=block：启用XSS保护，并在检查到XSS攻击时，停止渲染页面（例如IE8中，检查到攻击时，整个页面会被一个#替换）；

浏览器提供的XSS保护机制并不完美，但是开启后仍然可以提升攻击难度，总之没有特别的理由，不要关闭它。

4. X-Content-Type-Options

互联网上的资源有各种类型，通常浏览器会根据响应头的Content-Type字段来分辨它们的类型。例如："text/html"代表html文 档，"image/png"是PNG图片，"text/css"是CSS样式文档。然而，有些资源的Content-Type是错的或者未定义。这时，某 些浏览器会启用MIME-sniffing来猜测该资源的类型，解析内容并执行。

例如，我们即使给一个html文档指定Content-Type为"text/plain"，在IE8-

网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com

中这个文档依然会被当做html来解析。利 用浏览器的这个特性，攻击者甚至可以让原本应该解析为图片的请求被解析为JavaScript。通过下面这个响应头可以禁用浏览器的类型猜测行为：

X-Content-Type-Options: nosniff,这个响应头的值只能是nosniff。

5、X-Content-Security-Policy

csp是一些指令的集合，可以用来限制页面加载各种各样的资源。目前，IE浏览器只支持CSP的一部分，而且仅支持X-Content- Security-Policy header。 相比较而言，Chrome和Firefox则支持CSP的1.0版本，csp的1.1版本则还在开发中。通过恰当的配置csp，可以防止站点遭受很多类型 的攻击，譬如xss和UI补偿等相关问题。

csp总共有10种配置形式，每一种都可以用来限制站点何时加载和加载何种类型的资源。他们分别是：

default-src：这种形式默认设置为 script-src, object-src, style-src, img-src, media-src, frame-src, font-src和connect-src.如果上述设置一个都没有的话，user-agent就会被用来作为default-src的值。

object-src - 决定从哪里加载和执行插件。

style-src - 决定从哪里加载css和样式标记。

img-src - 决定从哪里加载图片。

media-src - 决定从哪里加载视频和音频资源。

frame-src - 决定哪里的frames 可以被嵌入。

font-src - 决定从哪里加载字体。

connect-src - 限制在 XMLHttpRequest, WebSocket 和 EventSource 中可以使用哪些类型的资源。

网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com

sandbox - 这是一个可选形式，它决定了沙盒的策略，如何将内容嵌入到沙盒中以保证安全。

当这个策略被特定的url违反了，我们也可以用报告地址直接发送报告。这样做有利于debug和当攻击发生时通知我们。

此外， 我们可以定义Content-Security-Policy-Report-Only 的 header不强制遵守csp，但是会发送潜在的威胁到一个报告地址。它遵守和csp一样的语法和规则。

实例

```
x-content-security-policy: default-src *;script-src *.facebook.net https://*.noka.com:*;
```

第2章 表单系列标签

2.1 基础录入框

基础录入框基础本属性列表。

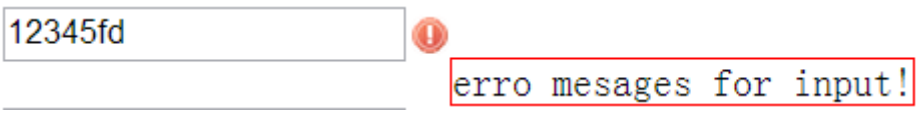
属性	是否必须	说明
name	是	
accept	否	
accesskey	否	
align	否	
alt	否	
border	否	
checked	否	
classstyle	否	同html中的class
dir	否	
disabled	否	
height	否	
id	是	
ismap	否	
istyle	否	

网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com

lang	否	
maxlength	否	
onblur	否	
onchange	否	
onclick	否	
ondblclick	否	
onfocus	否	
onhelp	否	
onkeydown	否	
onkeypress	否	
onmousedown	否	
onmousemove	否	
onmouseout	否	
onmouseup	否	
onselect	否	
readonly	否	
size	否	
width	否	
value	否	
title	否	
tabindex	否	
style	否	
src	否	
allownull	否	是否允许为空
maxclass	否	使用输入放大效果时，放大div的class属性
onshowmax	否	在使用放大效果时，调用js方法,可用于格式化放大显示的值
注：	以上属性未特别说明者与html同属性	

该标签生成一个录入框，具有基本的验证功能，并可以自定义方法验证和自定义正则表达式验证。

- **实例图**



- **属性说明**

属性	是否必须	说明
mesg	否	错误提示消息
chtype	否	验证类型可选类型：js|server|regex，分别表示js、服务端、正则表达式验证
veri	否	验证内容，根据chtype其值有所不同

该控件验证结果可以通过该控件对象的veri()方法获得，为true表示验证通过，为false表示验证未通过。如该控件的id为“ss”时，可以用$(“ss”).veri()来获得验证结果。

使用自定义js验证方法时，chtype=“js” very=“js方法的名称”，方法接收一个参数，根据验证结果返回true或false。

```
function bbsnp(a){
   alert(a);//需要验证的内容
   return true;//验证成功
}
```

服务器验证方法如下：（服务器调用地址为：inputtextcheck[id].itck)其中[id]为控件的ID值。

serverCheck:为服务器的全路径方法，如：

org.noka.Ftest.rut(this.value,’a’,56,:id_java.lang.String,request)

其中，this.value表示取该控件的值，为固定写法。

‘a’为常量参数，表示字符串的写法

56为常量参数，为数字型的写法

以:开始的，表示在验证时从页面的上的其它控件件里获取值，传递到后端，:后面的id表示页面控件的id，_为固定分隔符，java.lang.String为数据类型，其支持的数据类型有：java.lang.String, java.lang.Integer, java.lang.double，如需要将页面上id为c的控件值在验证时一起传递到后端，侧可以这样写：org.noka.T.a(this.value,:c_java.lang.String)，此时后端的验证方法原型为public String a(String a,String c);

网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com

如果需要传入request对象，则可直接写request(全部小写)，建议用ServletNokaContext代替，体具使用方法见6.8节。

方法的参数，可以是任意多个，也可不带参数,服务器上的写法如下：

```
package org.noka;
import javax.servlet.http.HttpServletRequest;
public class Ftest {
public static String rut(String v,String a,Integer b, HttpServletRequest request){
System.out.println("v:"+v);
System.out.println("a:"+a);
System.out.println("b:"+b);
return a+":"+b; }
}
```

服务器上的方法必须返回字符串类型，0表示验证失败，1表示验证成功，

使用正则表达式验证方法时，chtype=“regex” very=“正则表达式”

该控件对象用javascript设置值时，建议用setValue(value)方法，如该控件的id为“aa”，则可以用$(“aa”).setValue(‘fff’)，该方法有两个参数，第一个参数是需要设置的值，第二个参数为验证结果，如果只有一个参数，则在设置值时会重新进行验证，如$(“aa”).setValue(‘fff’)会重新对值进行验证，如果是$(“aa”).setValue(‘fff’,true)则表示验证通过，为false为验证不通过

使用实例如下

使用正则表达式验证。

```
<n:InputText name="accs" id="s" allownull="false" chtype="regex"
onshowmax="dcd" veri="[0-9]{5}"></n:InputText>
```

其中dcd为js方法，该方法实例如下所示:

```
function dcd(a){
    return a+',';
}
```

2.2 密码录入框

该控件的所有操

同2.1的inputtext控件

2.3 Checks 选框组

该标签在页面生成一组多选框组，提供 JavaScript 的设置值和获取值方法。

➢ **属性说明**

属性	是否必须	说明
id	是	选择框的ID
name	否	选择框的name，不同name为一组
sql	否	以sql方式设置初始选项，sql必须包含value，label两个属性，name，nid为可选列，nid为各选项框的ID
onclick	否	单击事件
json	否	以json方式设置选项框
formatcell	否	格式化列，如需要对某列的值进行格式化后输出，可以这样写：包名.类名.方法名(java类型 列名1，java类型 列名n，java类型 ’常量’) as 列名。多个列用|相隔。这里的类为自定义的普通类。
value	否	选项框的初始值，多个值用逗号分隔
readonly	否	只读，可选true|false
selectsun	否	至少需要选择的项目数，默认为0，可选值为0-n，n不能超过总的checks数量
classstyle	否	选项框的class属性
style	否	选项框的style
stype	否	选项框的label方向，可选right|left
disabled	否	选项框的disabled属性，可选disabled|no
width	否	选项组的整体宽度
divstyle	否	外部div的style属性
divclassstyle	否	外部div的class属性

写法如下：

```
<n:Checks id="ds" json="[{value:'1',label:'wan'},{value:'2',label:'man'}]" value="1"
```

```
onclick="nclick"></n:Checks>
```

Onclick:该属性为单击选项框时调用的外部js方法，属性传入的是方法名称（不带括号），该方法在调用时传一个参数，该参数是所点击选框的对象，如下所示：

```
function nclick(a){
  alert(a.value);
  return true;
}
```

该方法需要返回一个boolean类型的值，返回false表示阻止选的继续执行，返回true表示正常执行。

该控件具有一些外部操作数据的js方法，调用方式是先取得该控件对象，然后调用其内置的外部方法，如该控件的ID为“ds”，可以使用以下方式调用

```
//----------------------------设置选项框控件----------------------------------------------------
var json = [{value:'1',label:'a'},{value:'2',label:'b'},{value:'3',label:'c'}];
$('ds').setJson(json);
//----------------------------设置值----------------------------------------------------
$('ds').setValue ('1,2');
//----------------------------获取选项值----------------------------------------------------
var    va = $('ds').getValue ();
for(i=0;i<va.length;i++){
   alert(va[i]);
}
```

2.4 Radios 选择框组

该标签在页面生成一组单选框组，提供JavaScript的设置值和获取值方法。

➢ **属性说明**

属性	是否必须	说明
id	是	选择框的ID
name	否	选择框的name，不同name为一组
sql	否	以sql方式设置初始选项，sql必须包含value，label两个属性，name，nid为可选列，nid为各选项框的ID

网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com

onclick	否	单击事件
json	否	以json方式设置选项框
formatcell	否	格式化列，如需要对某列的值进行格式化后输出，可以这样写：包名.类名.方法名(java类型 列名1，java类型 列名n，java类型 ’常量’) as 列名。多个列用|相隔。这里的类为自定义的普通类。
value	否	选项框的初始值
readonly	否	只读，可选true|false
allownull	否	是否允许为空，可选:true|false
classstyle	否	选项框的class属性
style	否	选项框的style
stype	否	选项框的label方向，可选right|left
disabled	否	选项框的disabled属性，可选disabled|no
width	否	选项组的整体宽度
divstyle	否	外部div的style属性
divclassstyle	否	外部div的class属性

写法如下：

```
<n:Radios id="ds" json="[{value:'1',label:'wan'},{value:'2',label:'man'}]" value="1" onclick="nclick"></n:Radios >
```

Onclick:该属性为单击选项框时调用的外部js方法，属性传入的是方法名称（不带括号），该方法在调用时传一个参数，该参数是所点击选框的对象，如下所示：

```
function nclick(a){
  alert(a.value);
  return true;
}
```

该方法需要返回一个boolean类型的值，返回false表示阻止选的继续执行，返回true表示正常执行。

该控件具有一些外部操作数据的js方法，调用方式是先取得该控件对象，然后调用其内置的外部方法，如该控件的ID为“ds”，可以使用以下方式调用

```
//----------设置选项框控件----------
var json = [{value:'1',label:'a'},{value:'2',label:'b'},{value:'3',label:'c'}];
$('ds').setJson(json);
```

```
//----------------------------设置值-----------------------------------------------------
$('ds').setValue ('1');
//----------------------------获取选项值-----------------------------------------------------
var    va = $('ds').getValue ();
for(i=0;i<va.length;i++){
   alert(va[i]);
}
```

2.5 日期选择框

DateTime标签生成一个输入框。此输入框右边有一个图标按钮，点击此图标弹出一个日期时间选择窗口。可以设定初始化当前日期和时间。

该标签可通过javascript方法设置其值，方法是：obj.setValue(v)，其中v是需要设置的值，obj指控件对象自身。

➢ **实例图**

➢ **属性说明**

属性	是否必须	说明
prave	否	控件参数，详见参数说明
isInitNowDate	否	是否初始化日期
注：其它属性同基本录入框		

➢ **使用实例**

其中的id是必须有的属性。

```
<n:date name="dddd" id="dss" prave="dateFmt:'yyyy-MM-dd HH:mm:ss'"
isInitNowDate="yes" ></n:date>
其中dateFmt:'yyyy-MM-dd HH:mm:ss'表示格式化时间字符串，多个参数用逗号隔开，如
dateFmt:'yyyy-MM-dd HH:mm:ss', isShowWeek:true
```


网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com

➢ 日期格式

格式	说明
y	将年份表示为最多两位数字。如果年份多于两位数，则结果中仅显示两位低位数。
yy	同上，如果小于两位数，前面补零。
yyy	将年份表示为三位数字。如果少于三位数，前面补零。
yyyy	将年份表示为四位数字。如果少于四位数，前面补零。
M	将月份表示为从 1 至 12 的数字
MM	同上，如果小于两位数，前面补零。
d	将月中日期表示为从 1 至 31 的数字。
dd	同上，如果小于两位数，前面补零。
H	将小时表示为从 0 至 23 的数字。
HH	同上，如果小于两位数，前面补零。
m	将分钟表示为从 0 至 59 的数字。
mm	同上，如果小于两位数，前面补零。
s	将秒表示为从 0 至 59 的数字。
ss	同上，如果小于两位数，前面补零。
w	返回星期对应的数字 0 (星期天) - 6 (星期六) 。
D	返回星期的缩写 一 至 六 (英文状态下 Sun to Sat) 。
W	返回周对应的数字 (1 - 53) 。
WW	同上，如果小于两位数，前面补零 (01 - 53) 。

➢ 动态变量

格式	说明
%y	当前年
%M	当前月
%d	当前日
%ld	本月最后一天
%H	当前时
%m	当前分
%s	当前秒
#{}	运算表达式，如:#{%d+1};表示明天
#F{}	{}之间是函数可写自定义JS代码

➢ 参数列表

属性	类型	默认值	说明

网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com

el	Element 或 String	null	指定一个控件或控件的ID,必须具有value或innerHTML属性(如input,textarea,span,div,p等标签都可以),用户存储日期显示值(也就是dateFmt格式化后的值)
vel	Element 或 String	null	指定一个控件或控件的ID,必须具有value属性(如input),用于存储真实值(也就是realDateFmt和realTimeFmt格式化后的值)
dateFmt	string	'yyyy-MM-dd'	日期显示格式
realDateFmt	string	'yyyy-MM-dd'	计算机可识别的,真正的日期格式
realTimeFmt	string	'HH:mm:ss'	无效日期设置(disabledDates),最大日期(maxDate),最小日期(minDate)以及快速日期都必须与它们相匹配
realFullFmt	string	'%Date %Time'	建议使用默认值
minDate	string	'1900-01-01 00:00:00'	最小日期(注意要与上面的real日期相匹配)
maxDate	string	'2099-12-31 23:59:59'	最大日期(注意要与上面的real日期相匹配)
startDate	string	''	起始日期,既点击日期框时显示的起始日期 为空时,使用今天作为起始日期(默认值)
isShowWeek	bool	FALSE	是否显示周
highLineWeekDay	bool	TRUE	是否高亮显示 周六 周日
isShowClear	bool	TRUE	是否显示清空按钮
isShowToday	bool	TRUE	是否显示今天按钮
isShowOthers	bool	TRUE	为true时,第一行空白处显示上月的日期，末行空白处

网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com

			显示下月的日期,否则不显示
readOnly	bool	FALSE	是否只读
errDealMode	int	0	纠错模式设置 可设置3中模式 0 - 提示 1 - 自动纠错 2 - 标记
autoPickDate	bool	null	为false时 点日期的时候不自动输入,而是要通过确定才能输入 为true时 即点击日期即可返回日期值 为null时(推荐使用) 如果有时间置为false 否则置为true
qsEnabled	bool	TRUE	是否启用快速选择功能
quickSel	Array	null	快速选择数据,可以传入5个快速选择日期 注意:日期格式必须与 realDateFmt realTimeFmt realFullFmt 相匹配
disabledDays	Array	null	可以使用此功能禁用周日至周六所对应的日期 0至6分别代表 周日至周六
disabledDates	Array	null	可以使用此功能禁用所指定的一个或多个日期
opposite	bool	FALSE	默认为false, 为true时,无效天和无效日期变成有效天和有效日期
onpicking	function	null	此四个参数为事件参数
onpicked	function	null	
onclearing	function	null	
oncleared	function	nul	

➢ 同置方法

函数名	返回值类型	作用域	参数	描述
$dp.$	Element	全局	el [string]: 对象的ID	相当于 document.getElementById
$dp.show	void	全局	无	显示日期选择框
$dp.hide	void	全局	无	隐藏日期选择框

网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com

$dp.$D	String	全局	id [string]: 对象的ID	将id对应的日期框中的日期字符串,加上定义的日期差量,返回使用real格式化后的日期串
			arg [object]: 日期差量,可以设置成	
			{y:[值],M:[值],d:[值],H:[值],m:[值],s:[值]}	
			属性 y,M,d,H,m,s 分别代表 年月日时分秒	
			{M:3,d:7} 表示 3个月零7天	
			{d:1,H:1} 表示1天多1小时	
$dp.$DV	String	全局	v [string]: 日期字符串	将传入的日期字符串,加上定义的日期差量,返回使用real格式化后的日期串
			arg [object]: 同上例的arg	
以下函数只在事件自定义函数中有效				
$dp.cal.getP	String	事件 function	p [string]: 属性名称 yMdHmswWD分别代表年,月,日,时,分,秒,星期(0-6),周(1-52),星期(一-六)	返回所指定属性被格式字符串格式化后的值[单属性]
			f [string]: format 格式字符串	
			设置方法参考	
dp.cal.getDateStr	String	事件 function	f [string]: 格式字符串,为空时使dateFmt	返回所指定属性被格式字符串格式化后的值[整个值]

2.6 组合选择框

selectDouble生成两个select选择框。左边的为可选数据选择框，右边为已选数据选择框。生成的选择框有排序功能。通过设置可以决定是否可以重复选择等。

- **实例图**

属性	是否必须	说明
Listshowname	否	左边选择框上面显示的信息，默认为：可选数据项
reightshowname	否	右边选择框上面显示的信息，默认为：已选数据项
lietselectname	否	左边选择框的名字
rightselectname	否	右边选择框的名字
lietselectheight	否	左边选择框的高，需要带单位
rightselectheight	否	右边选择框的高，需要带单位
lietselectid	否	左边选择框的id属性，等同于html中的select的id属性
rightselectid	否	右边选择框的id属性，等同于html中的select的id属性
lietlist	否	左边选择框的初始化的数据，为json对像
rightlist	否	右边选择框的初始化的数据，为json对像
lietSql	否	左边选择框初始化的sql语句
rightSql	否	右边选择框初始化的sql语句
buttionClass	否	选择框中间按钮的class属性，等同于html中的class属性
lietselectwidth	否	左边选择框的宽度。等同于html中的select的width属性
rightselectwidth	否	右边选择框的宽度。等同于html中的select的width属性
isfind	否	是否允许添加重复项，其值为{v|t|vt}，v表示按值检查，t按显示值检查，vt按值及显示值检查

rformatcell	否	左边选框格式化列，如需要对某列的值进行格式化后输出，可以这样写：包名.类名.方法名(java类型 列名1，java类型 列名n, java类型 ’常量’) as 列名。多个列用|相隔。这里的类为自定义的普通类。
lformatcell	否	右边选框格式化列，如需要对某列的值进行格式化后输出，可以这样写：包名.类名.方法名(java类型 列名1，java类型 列名n, java类型 ’常量’) as 列名。多个列用|相隔。这里的类为自定义的普通类。
readonly	否	是否只读，同HTML标签属性

➢ **使用实例**

```
<n:selectDouble
lietshowname="可选数据项"
reightshowname="已选数据项"
 lietSql="select job_id as value,job_desc as text from jobs"
 />
```

Formatcell:可用类型有：

java.lang.String

java.lang.Integer

java.lang.Double

javax.servlet.http.HttpServletRequest,建议用ServletNokaContext代替，体具使用方法见6.8节。

json对像格式如：

{text:'fffffffffff',value:'1'},{text:'aaaaaaaaaaaa',value:'2'}

2.7 下拉选择框

select生成一个html中的select选择框，可以设只读属性和初始值。该标的ajax加载路径为：nselect[ID].select，其中[ID]为控件的实际ID。

该标签会自动生成一个javascript方法来设置该标签的值这个方法如下：

```
Obj.setValue(values)
```

其中Obj为该标签对象，values为需要设置的值。例如，如果设置的为aa,需要向该标签设的置为2，则应该这样写

```
$( ‘aa’ ).setValue(‘2’)
```

该方法有两个参数，第一个参数是需要设置的值，第二个参数为验证结果，如果只有一个参数，则在设置值时会重新进行验证，如
$(“aa”).setValue(‘fff’)会重新对值进行验证，如果是
$(“aa”).setValue(‘fff’ ,true)则表示验证通过，为false为验证不通过

通过javascript设置选项，方法原型: function obj.setJson (json)，其中obj为该标签对象，json为json对象。例如标签的id为aa时，当需要通过javascript动态改变选项时，可以用下面的代码：

```
function stc(){
var obj = [{value:’1’,text:’ffff’},{value:’2’,text:’csss’}];
$( ‘aa’ ).setJson (obj);
 }
```

当使用sql加载时，可使用obj.Reload刷新选项数据，其obj由实际控件对象替换。

➢ **效果图**

属性	是否必须	说明
name	是	控件名字
id	是	控件id
Value	否	控件的初始值
sql	否	控件的sql语句形如:select ttt as value,name as text form table
json	否	传入json字符串，生成选项列表

网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com

		[{value:'1',text:'ffff'},{value:'2',text:'csss'}]
height	否	控件高，超过这个高度会显示滚动条
width	否	控件宽度
onchange	否	在值改变时，调用的js方法，该方法传入一个javascript对象，该对象包括value和text以及其它sql中的属性(属性全部小写)。
readonly	否	是否只读，可选readonly|no
allownull	否	是否可以为空，可选true|false
lower	否	多个select级联时，下一个关联的控件ID
slang	否	在允许为空时，提示请选择的信息
selectwidth	否	下拉选择层的宽度，如果设置了该属性，选项会以紧凑的方式进行排列
optionwidth	否	在selectwidth不为空的时候，通过该属性设置每个选项为固定宽度

➢ **使用实例**

```
<n:select id="fsd" json="[{value:'1',text:'ffff'},{value:'2',text:'csss'}]"/>
```

使用sql时，sql的第一个字段会被用来作为值使用，第二个字段会被用来作为显示值使用。对字段名没有特别要求。

Onchange方法，如果指定了该方法，需要在该方法里面返回true来继续执行选择事件，如果返回false或是不返回，则放弃执行选择事件。

通过对象的veri()方法获取为空验证结果。

关于多外select级联，在级联时只能使用sql进行级联，级联的关联参数为选择框的值，如下所示：

```
<n:select id="a" sql="select value,text from a" lower="b"/>
<n:select id="b" sql="select value,text from b where bid=?" lower="d"/>
<n:select id="c" sql="select value,text from c where cid=?"/>
```

上例是三个控件的级联，在a控件选择时，会自动将a控件的值作为参数传给b控件，b控件将传过来的值作为sql参数重新加载选择项，同样在b控件选择时，会自动将b控件的值作为参数传给c控件，c控件将传过来的值作为sql能数重新加

网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com

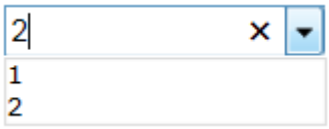
载选择项，sql为标准的sql语句，参数以?代替，其中的bid和cid表示sql的查询条件字段。会以上一个关联的控件值自动填充。

Onchange属性设置的值为js方法的名称（不要扩号和参数），如下面所示的实例。

```
<n:select id="tran_disc"
name="tran_disc"
sql="SELECT DIID,DINAME,DIDISC,DIREC, DIVOUC FROM NK_OPE_DISC"
onchange="disconchang"/>
<script type="text/javascript">
    function disconchang(node){
        $('trmoney_id').value=node.direc;// 自定义参数direc
        $('trdjq_id').value=node.divouc;// 自定义参数divouc
        alert(node.value+':'+node.text);//当前选种的value和text
        return true;
    }
    </script>
```

2.8 可录入下拉选择框

该标签在页面上打印一个可以输入内容的下拉选框。在录入内容有自动查找功能。该标的ajax加载路径为：nselect[ID].select，其中[ID]为控件的实际ID。



属性说明

属性	是否必须	说明
name	是	控件名字
id	是	控件id
Value	否	控件的初始值
sql	否	控件的sql语句形如:select ttt as value,name as text form table
json	否	传入json字符串，生成选项列表[{value:'1'},{value:'2'}]
height	否	控件高，超过这个高度会显示滚动条
width	否	控件宽度

网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com

onchange	否	在值改变时，调用的js方法，该方法传入一个value值
readonly	否	是否只读，可选readonly|no
allownull	否	是否可以为空，可选true|false

➢ **使用实例**

```
<n:selectinput id="fsd" json="[{value:'1' },{value:'2'}]"/>
```

使用sql时，sql的第一个字段会被用来作为值使用，对字段名没有特别要求。

Onchange方法，如果指定了该方法，需要在该方法里面返回true来继续执行选择事件，如果返回false或是不返回，则放弃执行选择事件。

通过对象的veri()方法获取为空验证结果。

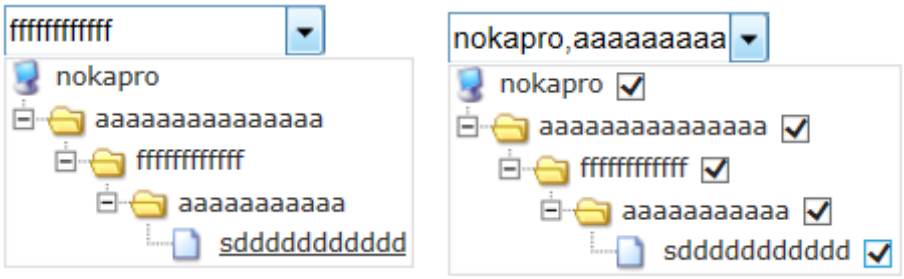
2.9 下拉树形选择框

下拉选择框树形菜单主要是在html中的下框中能显示树形的选择结构。根父ID取值为所有父ID当中最小的一个。也可以通过指定spid属性来指定其父id。

该标的ajax加载路径为：ntree[ID].tre，其中[ID]为控件的实际ID。

该控件所选择的值会存在一个与该控件同名，同id的input框内，如果需要取得该控件秘有勾选的值时，可以通过该id得到，如果要取得显示值时可以通过该id加上_sinput，如$(‘id’).value为所选的全部值，该值会在级联选择时将没有加载的值也一并记录，该特性需要leve支持。显示值则用$(‘id_sinput’).value取得。

➢ **实例图**



➢ 属性说明(treeselect)

属性	是否必须	说明
id	是	控件的id，可通过该id像访问普通input控件一样访问里面的值
sql	是	sql语句
url	否	指定一个全局的连接url
target	否	指定一个全局的连接目标
formatcell	否	格式化列，该功能只能格式化name，title两个字段，其它自定义字段不能格式化 格式化列,如需要对某列的值进行格式化后输出，可以这样写：包名.类名.方法名(java类型 列名1,java类型 列名n, java类型 ’常量’) as 列名。多个列用|相隔。这里的类为自定义的普通类。
pramevar	否	指一个连接需要携带的参数，由sql字段产生
spid	否	指定一个父节点id，用于开始的root节点
level	否	预加载的节点深度，默认加载2级
titleonclick	否	点击标题时，调用的方法
nodeonclick	否	点击节点前面的图标（展开或关闭节点）时调用的方法
name	否	控件的name
checktype	否	在多选时，节点间的级联关系 级联选项，可选值｛all | next | previous | nomuch｝，all：全部级联选择，nex：向下级联选择，previous：向上级联选择，nomuch：非级联多选默认为单选。

网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com

titlecliek	否	节点标题是否响应点击事件，可选值{true|false}默认为true，表示响应
checkboxonclick	否	点击选择框调用的方法，只在设置了checktype属性之后才能启用
value	否	控件的初始值
width	否	宽度
readonly	否	是否只读，可选项readonly|no
checkname	否	选择框的名字
allownull	否	是否可以为空，可选true|false

➢ **使用实例**

```
<n:treeselect
checktype="all"
id="sfd"
level="10"
sql="select tid as id,tpid as pid,tname as name,tname as title,tlvie as leve,ttype from nk_sys_treebook"
titleonclick="snp"/>
```

formatcell:可用类型有：

java.lang.String,java.lang.Integer,java.lang.Double,

javax.servlet.http.HttpServletRequest建议用ServletNokaContext代替，体具使用方法见6.8节。

在sql中，id,pid,name,title,leve为必须的字段，其中id,pid指定了树形关系，nam和title用于显示在界面上，leve为树形的级别号，用于指定其下级节点，如果sql时面指定了url、target属性则优先使用sql里面的设置，其数据看起来可如下：

TID	TPID	TNAME	TTEXT	TTYPE	TPX	TUSER	TDATE	TURL	TLVIE
1	-1	nokapro	(Null)	7	1	(Null)	(Null)	(Null)	0_-1
14	1	ddddddddd	sss	7	1	(Null)	(Null)	(Null)	0_-1_1
28	14	qwwqwq	sss	7	1	(Null)	(Null)	(Null)	0_-1_1_14
29	28	qwww	sss	7	1	(Null)	(Null)	(Null)	0_-1_1_14_28
30	14	uuuuuuuuuuu	sss	7	1	(Null)	(Null)	(Null)	0_-1_1_14
39	29	3333333333	sss	7	1	(Null)	(Null)	(Null)	0_-1_1_14_28
40	39	eeeeeeeeeee	sss	7	1	(Null)	(Null)	(Null)	0_-1_1_14_28

该控件会自动生成一些供外部调用的javasctip方法，用于控制该控件的数据操

作，包括刷新数据，指定置，打开指定节点。

Obj.LoadData： 刷新指定节点的数据，其参数为指定节点的值，obj用实际的控件对象替换，如：

```
Obj.LoadData( '3' );
```

Obj.setValue ： 设置控件的值，多个值用逗号分隔。

```
Obj.setValue (values)
```

其中obj为该标签对象，values为需要设置的值，多个值用逗号分隔

例如，如果标签的id设置的为aa,需要向该标签设的置为2，则应该这样写

```
$( 'aa' ).setValue('2') 或 $( 'aa' ).setValue('2,3,4')
```

Obj.opennode ： 打开指定的节点，如果它的父节点未被打开，会依次打开其父节点，注：该功能在使用之前，必须确保所打开的节点已经加载完成，否而不会生效。参为要打开的节点值。

titleonclick ： 在点击标题时，调用的javascript方法名，指定期属性时只写方法名。在调用时会传入节点对象，包括sql内的所有字段，其字端名全转换成小写，如sql中有一个字段TTYPE（不区分大小写），则访问如下：

```
//titleonclick=" snc"  属性这样设置
function scn(node){
alert(node.ttype);
}
```

nodeonclick，checkboxonclick两个属性使用方法同上。Titleonclick以及nodeonclick两个方法传入的node对象还额外包含属性名为selfid的属性，其值为该控件的自身ID，checkboxonclick方法传入的node对象额外包含名为selfid和checked两个属性，selfid指该控件自身的ID，checked属性为该节点的check框的选中状态，为true表示选中，为false表示未选中，该方法需要返回一个boolean类型的值，返回true表示继续执行余下下动作，返回false表示不执行余下的动作。可以通过对象的veri()方法获取为空验证结果。

网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com

2.10 下拉多选框

selectmtlti生成一个html中的select选择框。可以设只读属性和初始值。该标的ajax加载路径为：nselect[ID].select，其中[ID]为控件的实际ID。

该标签会自动生成一个javascript方法来设置该标签的值这个方法如下：

```
Obj.setValue(values)
```

其中obj为该标签对象，values为需要设置的值。例如，如果设置的为aa,需要向该标签设的置为2和3，则应该这样写

```
$( ‘aa’ ).setValue(‘2,3’)
```

通过javascript设置选项，方法原型：function obj.setJson (json)，其中obj为该标签对象，json为json对象。例如标签的id为aa时，当需要通过javascript动态改变选项时，可以用下面的代码：

```
function stc(){
var obj = [{value:’1’,text:’ffff’},{value:’2’,text:’csss’}];
$( ‘aa’ ).setJson (obj);
 }
```

当使用sql加载时，可使用obj.Reload刷新选项数据，其obj由实际控件的对象替换。

➢ **效果图**

属性	是否必须	说明
name	是	控件名字
id	是	控件id
Value	否	控件的初始值
sql	否	控件的sql语句形如:select ttt as value,name as text form table

网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com

json	否	传入json字符串，生成选项列表[{value:'1',text:'ffff'},{value:'2',text:'csss'}]
height	否	控件高，超过这个高度会显示滚动条
width	否	控件宽度
onchange	否	在值改变时，调用的js方法，该方法传入一个javascript对象，包括value和text两个属性。
readonly	否	是否只读，可选readonly|no
allownull	否	是否可以为空，可选true|false
selectwidth	否	下拉选择层的宽度，如果设置了该属性，选项会以紧凑的方式进行排列
optionwidth	否	在selectwidth不为空的时候，通过该属性设置每个选项为固定宽度

➢ **使用实例**

```
<n:selectmtlti id="fsd" json="[{value:'1',text:'ffff'},{value:'2',text:'csss'}]"/>
```

使用sql时，sql的第一个字段会被用来作为值使用，第二个字段会被用来作为显示值使用。对字段名没有特别要求。

Onchange方法，如果指定了该方法，需要在该方法里面返回true来继续执行选择事件，如果返回false或是不返回，则放弃执行选择事件。可以通过对象的veri()方法获取为空验证的结果。

2.11 文件上传组件

文件服务标签生成一个按钮和一个隐藏控件，点击按钮后弹出一个选择和上传文件的界面，选择后在原签所在网页上显示选择文件名，多个文件名用”,”隔开。值为文件的id号（由文件服务器自动生成）。

文件上传地址为：file[id].fs，其中[id]为控件id

文件读取地址：nokatag/nokatag_uploadfile/file[id].fs?do= red&fileid=文件id

文件下载地址：nokatag/nokatag_uploadfile/file[id].fs?do= get&

网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com

fileid=文件id

在程序后台通过org.nokatag.system.UploadFileRead的UFileRead方法获取该方返回的为FileItem对象，该对象包含了文件的基础信息和inputStream对象。该方法的原型为：public FileItem UFileRead(String fileid,Connection conn)，其中filed为上传文件的id，在使用DISK存储时，Connection对象可为空。

FileItem对象包含的属性(通过get相应的属性获取)如下：

String fileid;//文件id

String filename;//文件名

String type;//文件类型，文件后缀名

String text;//备注

Date createtime;//创建时间

InputStream file;//文件输入流

前台可以通过js方法设置值，方法原型为：

function noka_tag_upload_file_setvalue_id(v)，该控件的setValue(v)方法与该方法等同，其中id用控件的id替换，v是文件的id号，多个用逗号相隔。例如，控件的id为fil，值为2324234,423534则调用设置值的js方法为

noka_tag_upload_file_setvalue_fil(‘2324234,423534’);

在使用数据存储时，请确认已经配置好db-connection-class选项

删除文件的方法DeleteFile(List<String> fileids,Connection conn)，传入多个文件ID和数据库对像，如果是非数据存储conn可以为空，该方法将批量删除指定的多个文件，任何一个文件删除失败都将直接返回false，剩下的工作停止执行，只当所有文件都删除成功时才返回true

➢ **属性说明**

网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com

属性	是否必须	说明
name	是	隐藏控件名字
id	是	隐藏控件id
filesize	否	上传单个文件的大小
filetypedescription	否	上传文件显示类型，建议输入格式为:图片(*.jpg;*.gif)等
filetypeextension	否	上传文件类型，多个类型用分号相隔，如*.jpg;*.gif
uid	否	用户自定义id
isdel	否	是否可以删除，可选值true|false
type	否	数据过滤类型，uid,sid,pid(分别为用户自定义id,session,page)
idreadio	否	是否单选，可选值true|false
showtype	否	显示方式，1,为下载，2，为查看，0，为没有操作
value	否	隐藏控件的值
onchange	否	当值改变时调用的方法形如aab(v1,v2,v3,v4)，其中v1为文件名，v2为标识ID，v3为文件读取地址，v4为文件下载地址
uploadfunction	否	外部调用上传方法名，该控件的uploadFile方法等同
readonly	否	是否只读，同HTML标签
noShow	否	是否显示，默认为false,可选{true|false}
textsun	否	备注字符数，默认为50，最大不能超过100
title	否	按钮显示信息
allownull	否	是否允许为空，默认为false，可选{true|false}

➢ **例用实例**

```
<n:fileupload name="dddd" id="fffffdd"></n:fileupload>
```

2.12 照片上传组件

照片上传组件在页面生成一个照片显示框，点击该框可实现上传照片。该组件继承于文件上传组件。值为文件的id号（由文件服务器自动生成）。

文件上传地址为：file[id].fs，其中[id]为控件id

文件读取地址：nokatag/nokatag_uploadfile/file[id].fs?do= red&fileid=文件id

文件下载地址：nokatag/nokatag_uploadfile/file[id].fs?do= get&fileid=文件id

在程序后台通过org.nokatag.system.UploadFileRead的UFileRead方法获取该方返回的为FileItem对象，该对象包含了文件的基础信息和inputStream对象。该方法的原型为：public FileItem UFileRead(String fileid,Connection conn)，其中filed为上传文件的id，在使用DISK存储时，Connection对象可为空。

FileItem对象包含的属性(通过get相应的属性获取)如下：

String fileid;//文件id

String filename;//文件名

String type;//文件类型，文件后缀名

String text;//备注

Date createtime;//创建时间

InputStream file;//文件输入流

前台可以通过js方法设置值，方法原型为：

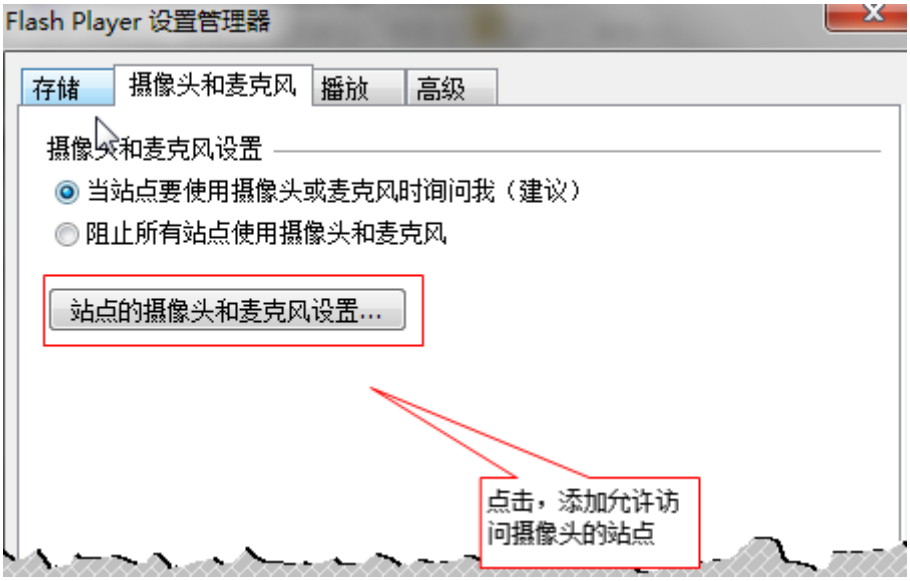
function noka_tag_upload_file_setvalue_id(v)，该控件的setValue(v)方法与该方法等同效果，其中id用控件的id替换，v是文件的id号，多个用逗号相隔。例如，控件的id为fil，值为2324234,423534则调用设置值的js方法为

noka_tag_upload_file_setvalue_fil(‘2324234,423534’);

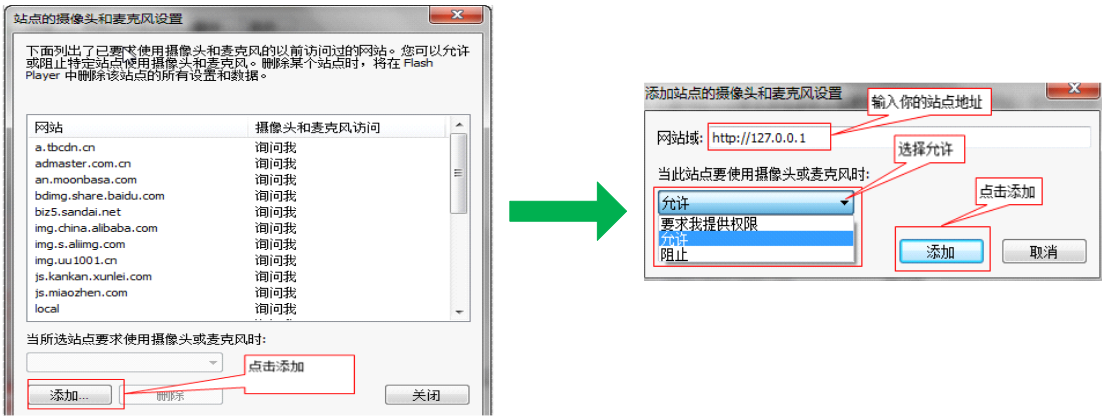
在使用数据存储时，请确认已经配置好db-connection-class选项

使用在线拍照时，需要浏览器支持flashPlay10以上的播放器，需要对安全控制进行设置，以便能访问客户机上的摄像头。设置如下：

在有“拍照”按钮的区域内，点击鼠标右键，选择“全局设置”在弹出的设置管理器中，选择“摄像头和麦克风”，点击“站点的摄像头和麦克风设置”如下图所示。



在弹出的“站点摄像头和麦克风设置”窗体中，添加你的站点，在下拉选框中选择允许，如下图所示：



➢ 属性说明

属性	是否必须	说明
Name	是	隐藏控件名字
Id	是	隐藏控件id
filesize		上传单个文件的大小
uid		用户自定义id

isdel		是否可以删除，可选值true|false
type		数据过滤类型，uid,sid,pid(分别为用户自定义id,session,page)
value		隐藏控件的值
onchange		当值改变时调用的方法形如aab(v1,v2,v3,v4)，其中v1为文件名，v2为标识ID，v3为文件读取地址，v4为文件下载地址
uploadfunction		外部调用上传方法，该控件件uploadFile与该方法等同效果
width		照片显示的宽
height		照片显示的高
msg		照片的提示信息
title		提示信息，同html标签的title属性
textsun		备注字符数，默认为50，最大不能超过100
utype		照片获取方式：1,为上传，2为拍照,3为上传加拍照，默认为1

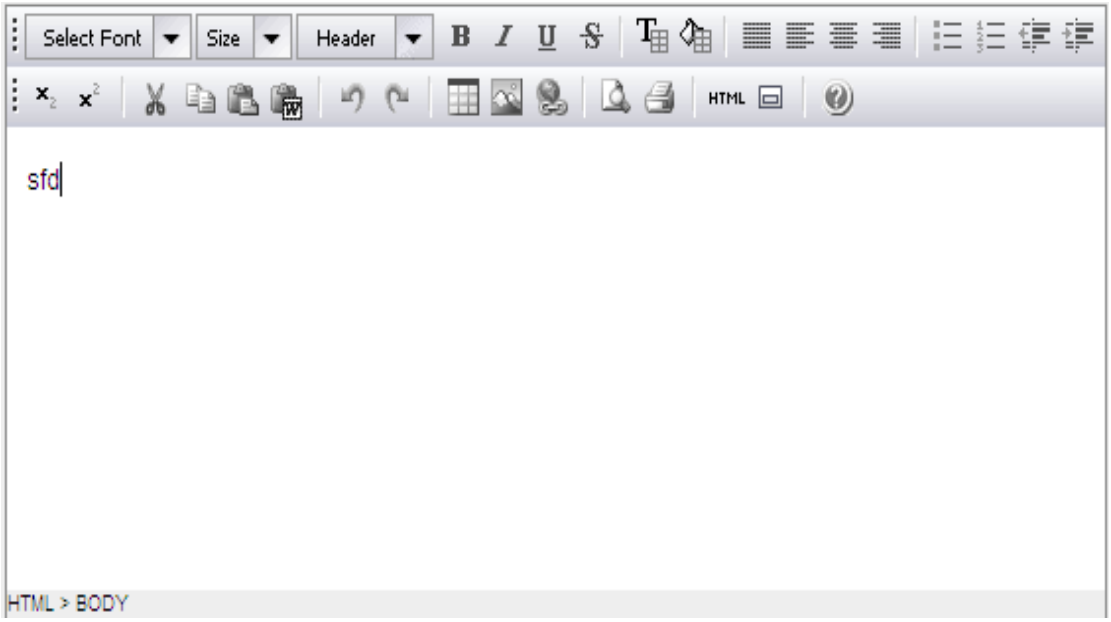
➢ **例用实例**

```
<n:ImageUpload name="sfd" id="dsf"/>
```

2.13 Html 编辑器

在网上生成一个可以编辑html的编辑框，该编辑具备文件上传与图片上传功能，上传文件的存放同上传组件使用的是同一处理机制。

➢ **效果图**



➢ 属性列表

属性	是否必须	说明
Name	是	隐藏控件名字
Id	是	隐藏控件id
width		编辑器的宽
height		编辑器的高
uid		用户自定义id
isdel		是否可以删除上传的图片，可选值true|false
type		数据过滤类型，uid,sid,pid(分别为用户自定义id,session,page)
filesize		上传文件的大小（附件和图片都使用该属性限定大小）
Value		编辑器的值
insertHTML		自定义插入内容的javascript方法
getValue		自定义获取内容的方法名
setValue		自定义设置内容的方法名
textsun		备注字符数，默认为50，最大不能超过100
annextypeextension		附件上传的类型，形如*.docx;*.doc;*.zip
annextypedescription		附件上传时显示允许上传的类型，形如annex(*.docx,*.doc,*.zip)

其中insertHTML为自动生成向编辑器里插入内容的javascript方法名，如insertHTML=“insertBody”，则在javascript里可以调用insertBody(要插入的

网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com

内容)。

getValue为自动生成的用于获取编辑器内容的javascript方法名，如getValue=“getvalue”，则在javascript里可以调用getvalue()来获取编辑器里的内容。

setValue为自动生成的用于设置编辑器内容的方法，用法同insertHTML。注，setValue与insertHTML的区别在于setValue会覆盖编辑器里面的内容，insertHTML是向编辑器里插入内容。

setValue和getValue以及insertHTML方法均有对应的对象调用方法，如$(id).setValue(v)，其中id为控件的id。

上传图片的路径为file[id].fs，其中ID为该控件的ID值。

➢ **使用实例**

```
<n:htmledit id="dsf" name="sfsdfd" value="ddddddddddddddddddd"/>
```

2.14 颜色选择框

在界面生成一个普通的录入框，该录入框的单击事件会弹出一个颜色选择器，选择的颜色值（十六进制）会填入录入框内，录入框的背景色是当前选择的颜色。

➢ **效果图**

网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com

➢ 属性说明

属性	是否必须	说明
customcolors	否	是否允许自定义颜色，可选值{true|false}
备注：		其它属性参考2.1基础录入框

➢ 使用实例

```
<n:ColorSelect name="dd" id="sssssssssss" ></n:ColorSelect>
```

2.15 验证码录入框

验证码录入框是在界面生一个录入框和一个验证码图片。同时会在session里面设置该验证码的值。默认的session名字为：noka_code。

➢ 效果图

➢ 属性说明

属性	是否必须	说明

网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com

fontColor	否	字体颜色，默认为红色，值为RGB格（255,0,0）
length		验证的位数
cheight		图片生成高度
cwidth		图片生成宽度
cmaxSize		最大字体（为整数值）
cminSize		最小字体（为整数值）
counter		干绕方式，共五种（值为整数，从0开始）
msg		更换图片的提示信息
codename		设置在session里面的名字。
kword		验证码可选值的ID，默认为1
imageStyle		验证码图片的style属性
imageClas		验证码图片的class属性
备注：		其它属性参考2.1基础录入框

➢ **使用实例**

```
<n:CodeImage name="dfd" id="df" cheight="73" msg="看不清图片？"></n:CodeImage>
```

其中kword的值对应于noka-config.xml里面的kword后的ID号，value里面的值为验证码的可选值。

服务器调用生成图片的路径为：codeimg_[id].codeimg，其中[id]为控件的ID值。

注：在多应用节点集群部署的环境下，支持cookie加密验证，建议在后端验证时直接调用org.noka.cookie.CookieUtil中的CookieImg方法，该方法需要传入request和接收到的验证码值，验证成功返回true，该方法验证机制是将传入的验证码与从cookie里面或得的验证码解密之后做对比，该验证方法在多应用节点集群不熟时不需要做session同步，任然可以正常验证。在使用cookie验证时，对应的配置文件中的option里面有一个cookie-domain选项，可以设置cookie的domain选项，以支持跨域共享。如：

```
CookieUtil cu = new CookieUtil();
String v = request.getParameter("sfff");//前台验证码控件名称为sfff
boolean bs = cu.CookieImg(request, null,v);//null这里是指控件设置的codename，没有设置侧设为null
out.println("bs:="+bs);
```

Noka-config.xml里面的配置如下。

```
<kword id="1" value="absdeghjkmnpqrstuwxyzABCDEFGHIJKLMNPQRST23456789"></kword>
<kword id="2" value="FDABZXSRETOP你好这是什么东西"></kword>
```

2.16 套接字

Socket标签在页面生成一个flash对像，该对像具有Socket的全部功能，通过这个对像可以实现用Socket与服务器通信。

➢ **属性说明**

属性	是否必须	说明
id	是	生成的flash对像的id
onDataFunction	是	在接收数据后触发的方法
openFunction	否	自定义打开套接字的方法名
sendFunction	否	自定义发送消息的方法名
closeFunction	否	自定义关闭连接的方法名
onConnection	否	在连接时的方法名，默认noka_+id+_onCon，该方法传入一个boolean值，true表示连接成功,false表示连接失败
isbug	否	是否启动调试模式，可选值true|false
ip	否	连接的tcp服务的服务器地址
port	否	Tcp连接服务端口，当ip和port都不为空时，控件将开启自动连接功能
sendmsg	否	发送消息的内容
interval	否	间隔发送的时间，当sendmsg和interval都不为空时，控件将开启自动间隔发送功能

该标签在生成时需要指定各个方法的调用名字,使用实例如下：

```
<script type="text/javascript">
function ondatas(text){
document.getElementById('ddd').value=text;
}
</script>
<n:socket isbug="true" onDataFunction="ondatas" id="cmsid" openFunction="opens"
sendFunction="sends" closeFunction="colsed" >
</n:socket>
<textarea rows="10" cols="60" id="ddd"></textarea>
```

```
<input type="button" name="df" value="联接" onclick="opens('127.0.0.1',8099);" />
<input type="button" name="ddf" value="发送" onclick="sends('中文');" />
<input type="button" name="ddf" value="关闭" onclick="colsed();" />
```

如上例所示：其中opens方法需要两个参数：一个为服务器的IP或域名，第二个为端口。

onConnection方法有一个参数，为true表示连接成功，为false表示连接失败

各种方法的名字在标签初始化时设置。

服务器端代码见SocketRun.java

该控件在控件所在页面关闭时自动调用关闭事件，除特殊情况下无须手动调用，该控件具有以下方法：

open(ip,port);//打开连接，如open(‘127.0.0.1’,8080);

send(msg);//发送数据，如send(‘aaa’);

sendTherd(msg, interval);//间隔一定时间自动发送线程interval为间隔时间，单位毫秒

close();//关闭连接

2.17 多行文本框

多行文本框为HTML文当中的Textarea替代品，主要集成了字符数的控制功能。

➢ **属性说明**

属性	是否必须	说明
name	是	
accept	否	
accesskey	否	
align	否	
alt	否	
border	否	
checked	否	

classstyle	否	同html中的class
dir	否	
disabled	否	
height	否	
id	是	
ismap	否	
istyle	否	
lang	否	
maxlength	否	
onblur	否	
onchange	否	
onclick	否	
ondblclick	否	
onfocus	否	
onhelp	否	
onkeydown	否	
onkeypress	否	
onmousedown	否	
onmousemove	否	
onmouseout	否	
onmouseup	否	
onselect	否	
readonly	否	
size	否	
width	否	
value	否	
title	否	
tabindex	否	
style	否	
src	否	
cols		
rows		
注：	以上属性未特别说明者与html同属性	
charsize	字符数	

- **使用实例**

```
<n:Textarea name="as" id="dfd" charsize="300"></n:Textarea>
```

- **实例图**

网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com

2.18 弹出框

该标签在页面上生成一个弹出框，可加载其它页面和自定义内容，支持自动关闭。

➢ **属性说明**

属性	是否必须	说明
id	是	控件的id
width	否	弹出框的宽
hight	否	弹出框的高
lotype	否	加载类型
onopen	否	显示时调用的方法
onclose	否	关闭时调用的方法

该标签写法如下：

```
<n:NokaBox id="bbccsn" width="600" height="700">http://www.baidu.com</n:NokaBox>
```

如上例所示：标签内部的http://www.baidu.com是弹出框要加载的地址，该控件件有两个方法分别为show和hide，其中show可带参数，也可不能带参数，没有参数的情况下，表示显示标签配置的信息（包括大小、加载地址等），有参数里，show方法的参数会覆盖标签自身的配置信息。hide表示隐藏弹出框。

Lotype可选参数为：iframe| html| url| image

iframe:直接在标签内部配置加载路径，以iframe方式显示(支持内部js)

html:直接在标签内部写要显示的内容

url:以ajax方式加载（同iframe，该方式不支持内部js）

image:直接加载图片

上例中的show方可以为：

```
//直接调用
$('bbccsn').show();
//没有动画，没有关闭按钮，自动宽度/高度，自定义的样式
show({html:'This is a warning!',animate:false,close:false,boxid:'error',top:5})
// Ajax后，固定宽度/高度，光掩模，自定义垂直分裂
show({url:'post.php',post:'id=16',width:200,height:100,opacity:20,topsplit:3})
//自定义的位置，自动隐藏
show({html:'The entry has been updated
successfully!',animate:false,close:false,mask:false,boxid:'success',autohide:2,top:
-14,left:-17})
// iframe，绝对位置，无框，关闭回调
show({iframe:'http://www.scriptiny.com/',boxid:'frameless',width:750,height:450,fix
ed:false,maskid:'bluemask',maskopacity:40,closejs:closeJS})
// Ajax加载
show({url:'advanced.html',width:300,height:150})
//加载图片
show({image:'images/rhino.jpg',boxid:'frameless',animate:true,openjs:function(){ope
nJS()}})
```

Show中的openjs和closejs等同于标签的onopen和onclose属性，分别为在打开和关闭时调用的javascript方法名称

如果以iframe加载网页时，在被加载的网页中可以像下面这样操作前网页内容。

```
window.parent.$(" bbccsn ").hide();//关闭弹出框
```

2.19 表单标签

表单标签(form)在页面生成一个form标签，集成了ajax提交数据方法，所有noka的form系列标签均可以做为该标签的子标签出现。该from标签在配置noka其它表单元素标签时，可实现提交时自动验证（包括ajax和普通提交）。

- **属性说明**

属性	是否必须	说明
subfunction	是	Ajax提交数据时调用的javasctipt方

		法
onsuccess	是	执行成功时的回调方法
onfailure	否	执行失败时的回调方法
onsubmit	否	在提交数据时调用的方法，该方法返回true继续提交，返回false停止提交
style	否	Form内部div的style属性
classstyle	否	Form内部div的class属性
labeldefaultwidth		在使用标签布局时的label默认宽度
labeldefaultheight		在使用标签布局时的label默认高度
inputdefaultwidth		在使用标签布局时的input栏默认宽度
inputdefaultheight		在使用标签布局时的input栏默认高度
注：其它属性同html的form属性		

该标签写法如下：

```
<n:form id="nk_forms" action="la.jsp" subfunction="snk" onsubmit="anb"
onsuccess="onsuccess" onfailure="onfailure">
```

如上例所示：snk为提交数据调用的方法（建议用ajaxSubmit()方法代替），如要提交该表单，调用javascript的snk()方法即可。

表单拥有JavaScript方法如下：

showInfo(msg,autohide)，显示提示信息,msg为提示信息内容,autohide指示显示autohide秒后自动关闭显示，该参数为数字，非必须参数。

showErro(msg,autohide)，显示错误提示信息，参数同showInfo()方法。

showLoad(autohide)，显示等待消息，autohie指示显示autohide秒后自动关闭，该参数为数字，非必须参数。

closeInfo(),关闭消息提示。

ajaxSubmit(url)，ajax异步提交表单数据，url为提交到服务器的url路径，非必须，如果未设置url参数，侧使用表单的action属性提交。

网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com

消息提示位置默认为置顶提示，既top和left都为0，当需要指定消息提示显示位置时，可用<n:forminfo/>标签指定其位置，该标签所在页面位置为消息提示显示所在位置，该标签只在form标签内部有效。

onsuccess和onfailure分别为成功/失败时的回调方法，两个方法返回相同的对象如下所示：

```
function onsuccess(a){
    alert(a.responseText);// 用a.responseText.evalJSON()返回json对象
}
function onfailure(a){
    alert(a.responseText);
}
```

默认情况下，如果没有设置onsuccess属性，在后台返回固定JSON数据格式时，表单控件会自动调用系统的消息提示，后台返回的JSON格式如下：

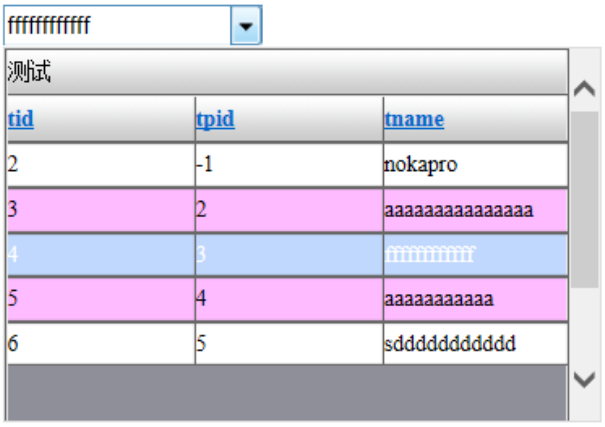
```
{s:'erro',m:'msg',a:3}//s代表消息显示类型,erro错误消息|success成功消息,m为显示的消息内容,a为多少秒以后自动隐藏，参数a可选
```

以上 JSON 格式可调用系统的表单工具类进行返回，该类位于 org.nokatag.system 下，使用实例如下：

```
FormUtil.ReturnSuccess(String msg);//返回错了消息
FormUtil.ReturnSuccess(String msg,Integer ahid);//返回错了消息
FormUtil.ReturnSuccess(String msg);//返回成功消息
FormUtil.ReturnSuccess(String msg,Integer ahid);//返回成功消息
FormUtil.ReturnJson(String json);//返回自定义JSON消息
```

2.20 自定义下拉列表选择框

自定义下拉列表选择是会在界面生成和下拉框一样的控件，下拉框里面的内容可以自己定义，其内容可以是嵌套的noka tag任何标签，也可以是自定义的任何内容，在自定义内容里面通过javasctipt方法给下拉控件设置选择值和显示名称。其效果如下图所示（该效果是嵌入了dbgrid标签）



➢ **属性说明**

属性	是否必须	说明
id	是	控件的id值，显示控件的id为该加上_sinput，如id_sinput
name	否	控件的name
width	否	控件的宽度
height	否	控件下拉列表的最大高度
value	否	控件的值
readonly	否	是否只读，可选项readonly|no
text	否	控件显示的文本

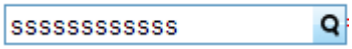
该标签写法如下：

setValue方法有两个参数，第一个参数是实际值，第二个参数是显示值

```
<n:selectustom id="sf" name="fd" height="200"  value="nnnnn" text="sss">
  <n:DBGrid width="300" height="200"  sql="select tid,tpid,tname from
nk_sys_treebook" id="cns" onrowclick="snb" title="测试"/>
  </n:selectustom>
  <script type="text/javascript">
  function snb(v){
    $('sf').setValue($(v.cells[1].id+'_div').innerHTML,$(v.cells[2].id+'_div').inn
erHTML);//给控件设置值
  }
  </script>
```

2.21 弹出选择输入框

弹出选择输入框是集普通输入框和弹出选择界面于一体的输入框，如下图所示：



➢ **属性说明**

属性	是否必须	说明
id	是	控件的id值，显示控件的id为该加上_sinput，如id_sinput
name	否	控件的name
width	否	控件的宽度
bwidth	否	弹出框的宽
bheight	否	弹出框的高
value	否	控件的值
text	否	控件的显示值
readonly	否	是否只读，可选项readonly|no
allownull	否	是否允许为空
lotype	否	加载类型，同nokabox控件
onopen	否	显示弹出框时调用的方法
onclose	否	关闭弹出框时调用的方法

该标签写法如下：

```
<n:selecbutton id="sfcc" allownull="false" onopen="scn">tj.jsp</n:selecbutton>
```

其中的tj.jsp为加载的网页，如果是显示内容，直接写需要显示的内容即可，

其它方法的使用与nokabox相同，该控件的setValue有两个参数，第一个参数为控件的值，第二个参数为控件的显示值。同样在被弹出页面可以像下面这样设置其控件的值：

```
window.parent.$("sfcc").setValue('1','sssssssssss');
window.parent.$("sfcc").hide();
```


网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com

2.22 表单布局标签(行)

该标签在表单标签内部使用，采用DIV布局模式，其作用是生成像表格布局中的行对象，如：

```
<n:frow></n:frow>
```

➢ 属性说明

属性	是否必须	说明
id	否	行div的id属性
style	否	行div的style属性
classstyle	否	行div的class属性

2.23 表单布局标签(列)

该标签在表单布局标签的行标签内部使用，采用DIV布局模式，其作用是生成像表格布局中的单元格对象，如：

```
<n:fcell></n:fcell>
```

➢ 属性说明

属性	是否必须	说明
label	否	单元格的label
style	否	单元格的style属性
classstyle	否	单元格的class属性
inputStyle	否	输入单元格的style属性，不设置时将采用style属性的值
inputclasss tyle	否	输入单元格的class属性，不设置时将采用class属性的值
labelwidth	否	Label单元格的宽
labelheight	否	Label单元格的高
inputwidth	否	input单元格的宽
inputheight	否	input单元格的高
labelid	否	Label单元格的id
inputid	否	input单元格的id

➢ 使用标签布局实例

```
<n:form id="f">
<n:frow>
    <n:fcell label="名称"><n:InputText  name="a" id="a"/></n:fcell>
    <n:fcell label="排序"><n:InputText  name="b" id="b"/></n:fcell>
</n:frow>
<n:frow>
    <n:fcell label="备注">
       <n:Textarea  name="c" id="c" cols="50" rows="5"></n:Textarea>
    </n:fcell>
</n:frow>
</n:form>
```

2.24 隐藏控件组(Hiddens)

该标签生成一组hidden控件，生成以后与普通的hidden控件操作相同，如：

```
<n:Hiddens id="df"
json="[{id:'a',name:'a',value:'a'},{id:'c',name:'c'}]"></n:Hiddens>
<!--其中json中的name属性必须要有，id和value属性可选 -->
```

在javaScript中仍然可以通过$(id)来获取该控件生成的控件对象，同时该控件提供getValue()方法获取所有hidden控件的值。isdebug开启测试模式(可选值为true|false)，由原来的hidden变为text控件，可在页面上观察其值的变化。

2.25 按钮控件(button)

按钮控件是在页面生成一个按钮，该按钮可设置拦截器、提交form表单，在需要控制按钮是否生成时，可以通过拦截器来实现（一般用于按钮权限控制），其拦截器配置是通过noka-config.xml里面的button-filter参数进行配置，其值为实现了org.nokatag.tagjava.ButtonInterface接口的类，如下：

```
<option key="button-filter" value="org.test.T"></option>
其实现类T源码如下
public class T implements ButtonInterface{
    public boolean show(PageContext pageContext,Button button) {
        return true;//返回true表示正常生成按钮，返回false表示不生成按钮
    }
}
```

按钮如果是在表单内部，可以通过formurl属性来提交所在表单，其formurl属性值为表单需要提交到的路径。如果需要提交外部表单，同时设置formid属性指定需要提交表单的ID即可。var属性是指button不在页面上生成，所生成的控件通过request.setAttribute(var,button)设置在页面变量里面，其变量名称为var指定的名称，在页面的任何地方可以通过${变量名称}获得。其它属性同html的属性。

```
<!-- 当按钮在表单内部时，提交表单到a.jsp -->
<n:button name="na" id="na" value="保存"  formurl="a.jsp"></n:button>
<!--当按钮在表单外部时，提交ID为nform的表单到a.jsp-->
<n:button name="na" id="na" value="保存"  formurl="a.jsp" formid="nform"></n:button>
<!--生成到页面变量里-->
<n:button name="na" id="na" value="保存"  var="sc"></n:button>
<!--引用上面的按钮-->
${sc}
```

2.26 超链接(a)标签

超链接标签是在页面上生成一个形如<a>的超链接标签，该标签的连接地址通过response.encodeURL编码，支持urlrewrite。对应的url编码EL表达式见EL工具章节。

```
<!—连接路径会自动补齐根路径 -->
<b:a href="index.jsp?key=8998">连接</b:a>
```

2.27 HTML 子字符串截取

该标签先将标签内的html文本清除纯txt文本以后，在进行按长度截取，可指定开始位置和截取长度，同时也可以指定格式化数，如XXX%sXXXX等。

```
<!—截取html子字符串 -->
<n:HtmlSubString len="5"  format="[%s***].doc">
<div>XXXXXXXX</div>
</n:HtmlSubString>
```

注：该标签如果指定了start，则实际len=start+len

网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com

2.28 滑动验证码

该标签生成一个可滑动的验证码，可代替原有的图形码验证功能。

```
<n:bcode width="200" name="ddc" id="ddc"/>
```

➢ 属性说明

属性	是否必须	说明
name	是	控件的名称
width	是	控件的宽度
id	是	控件的id
title	否	初始提示信息
stitle	否	验证成功以后的提示信息

在后台可调用org.noka.bver. CodeServer下的isCheck方法验证，传入的参数为控件的name属性，如上例所示，侧对应的后端验证为：

```
CodeServer cu = new CodeServer();
boolean bs= cu.isCheck(request, "ddc");
if(bs){
   //验证成功
}
```

2.29 Script 标签

Script标签等同于html的script标签，用于加载外部JS文件，在加载时，对当前request作用域内进行唯一性检测，即确保相同JS文件在同一个request作用域内只被加载一次，防止重复加载

```
<n:script src="js/stest.js" charset="utf-8" type="text/javascript"></n:script>
//chrset和type为可选参数
//src 在没有以http或是https开头的时候，将自动以系统根路径填充，其根路径生成规则见配置文件说明
```

➢ 属性说明

属性	是否必须	说明
src	是	src属性，js文件路径
type	否	type属性，默认为text/javascript

网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com

charset	否	Charset属性，默认不设置

2.30 Link 标签

link标签等同于html的link标签，用于加载外部CSS文件，在加载时，对当前request作用域内进行唯一性检测，即确保相同CSS文件在同一个request作用域内只被加载一次，防止重复加载

```
<n:link href="css/css.css"></n:link>
// rel和type为可选参数
//href 在没有以http或是https开头的时候，将自动以系统根路径填充，其根路径生成规则见配置文件说明
```

➢ **属性说明**

属性	是否必须	说明
href	是	href属性，CSS文件路径
type	否	type属性，默认为text/css
rel	否	rel属性，默认不设置

2.31 Use 标签

批量加载JS、CSS文件，在加载时，对当前request作用域内进行唯一性检测，即确保相同文件在同一个request作用域内只被加载一次，防止重复加载。

```
<n:use>script.*;js.test;</n:use>
// rel和charset为可选参数
//标签内为相对于系统根目录的相对路径，多个路径用分号分隔，如上例所示，script.*；表示加载script目录下的所有JS和CSS文件，js.text；表示加载js目录下的text文件。多级目录间用.分隔，标签自动识别CSS和JS文件类型。
```

➢ **属性说明**

属性	是否必须	说明
charset	否	加载JS文件时的charset属性

网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com

rel	否	加载CSS文件时的rel属性

第3章 数据处理系统列标签

3.1 数据表格

DBGrid用于生成一个数据显示表格。此数据表包括分页，按列排序，自带选择框等功能。目前DBGrid支持SQL Server，Oracle，MySQL，PostgreSQL，DB2五种数据库海量数据分页，其它为普通分页。

该标签在子表显示时，当使用sql和class方式时为Ajax加载方式。

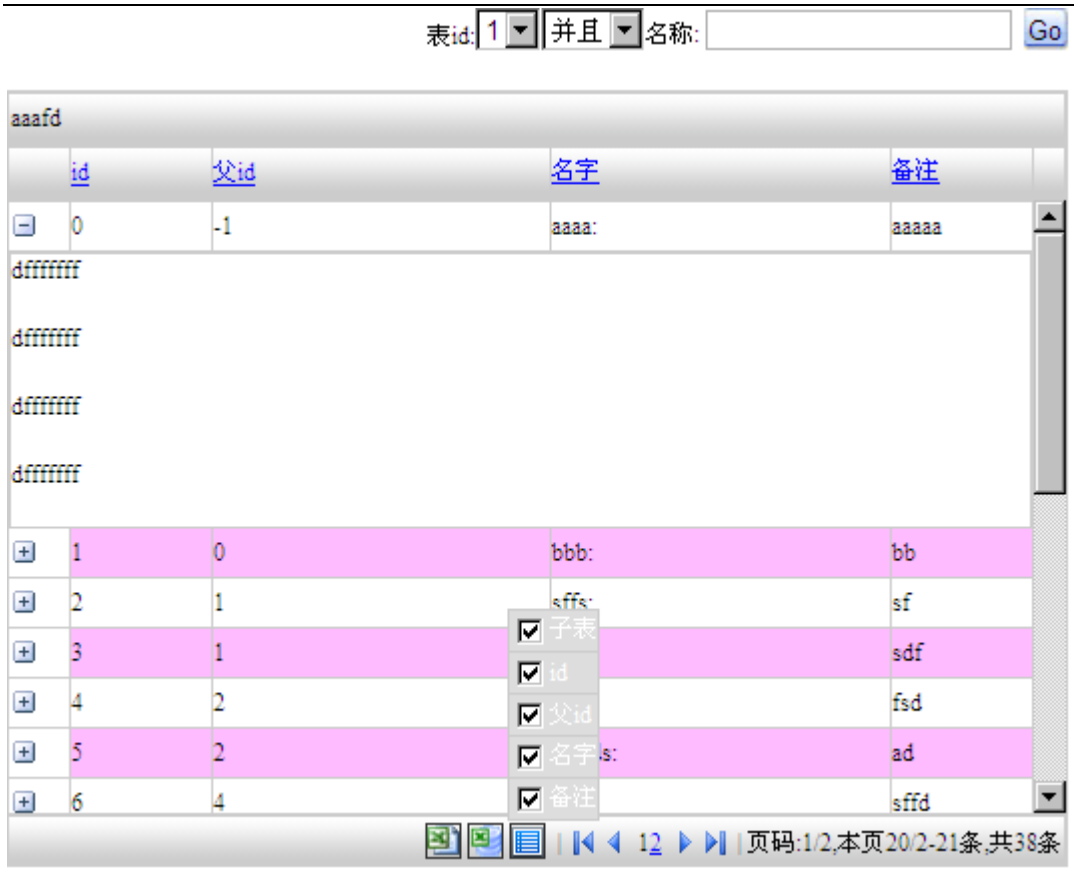
注：此标签集成排序功能，因此不能在sql语句中使用排序的语句。

从6.0.1开始，DBGrid改为异步加载数据，加载数据的url为规则为：nk[dbgrid]db.ndbgrid，其中的[dbgrid]为dbgrid的实际ID值，如一个实例的id为da_id，则加载路径为：nkda_iddb.ndbgrid，该路径noka会自动识别，如果需要对该路径作资源控制需要将处理放在noka拦截器之前。

导出到Excel的服务器路径为：outexcel[dbgrid].gridexcel

加载子表的服务器路径为: subgrid[dbgrid].sgr

➢ **实例图**



➢ **属性说明**

属性	是否必须	说明
sql	是	sql语句，像这样：select id as ID, name as 名字 from user ， Sql语句中列出再的顺序将决显示界面的列的顺序
Width	是	表格的宽，不带px单位
Height	是	表格的高，不带px单位
Id	是	表格的ID，页面内唯一
styleid		样式表的id，如果没设此属性则用系统样式,设为-1时表示使用外部样式
pageSize		页面显示的数据条数，如果不设定，则不分页
title		表格标题
outtitle		导出到文件时显示的标题
selecttitle		查询按钮的value属性
Cells		列选项如width:100,show:1多列之间用|分隔，show可选值为：1显示/可选择，2显示/不可选择,3不显示/可选择，4不显示/不可选择
number		是否显示序号，值为{true|false}
compositor		初始排序的列名，如果用了as则为as后的名字
descorasc		出始排序的方式，值为{asc|desc}

网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com

checkvalue		选择框的值，默认为SQL的第一列做为其值，用列的顺序号指定
checkname		选择框的名字
tableformname		数据表的form名字
tableformid		数据表的form的id
tableformAction		数据表的form的action
tableformtarget		数据表的form的target
tablemethod		数据表的form的method
checkd		是否要选择列，其值为{ checkbox | radio }
textColor		选种行或鼠标经过某行时该行文本的颜色
backgroundColor		选种行或鼠标经过某行时该行背景颜色
checks		选择框的初始化值，该属性为一个String，多个值以逗号隔开
subtabletype		子表类型{cell|sql| class }分别为主表的列值方式，独立的sql方式，自定义类方式。
subtablestyleid		子表标式id，只在子表类型为cell里起作用
subtable		子表参数，子表参数(如果是取主表列则直接写列序号从0开始如：1,2...)如果是sql语句则直接写sql语句，其中如果要用到主表参数直接留?号，并在subtablestyleid内加入主表列序号(从0开始)。如果是class，则直接写类的路径（包括类名）如org.user.TestGrid。该类需要继承org.nokatag.dbgrid.SubGrid。
Subformatcell		子表的列格式化参数，使用规则同formatcell参数
selectfiled		查询字符串 列名 条件 $[录入框名],列名 条件 $[录入框名]
selectinput		是否自动生成查询录入框，可选值{true|false}
formatcell		格式化列，如需要对某列的值进行格式化后输出，可以这样写:包名.类名.方法名(java类型 列名1,java类型 列名n, java类型 ’常量’) as 列名。多个列用|相隔。这里的类为自定义的普通类。
onrowclick		行的单击事件，传入该行对象实例
onrowdblclick		行的双击事件，传入该行对象实例
outexcelnopage		导出当前页到excel，默认为false，可选{true|false}
outexcelall		导出所有页到excel，默认为false，可选{ true|false }
Outcsvnopage		导出当前页数据到CSV

网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com

Outcsvall		导出所有数据到CSV
outpdfnopage		导出当前页数据到PDF,在支持pdf显示的浏览器上可直接显示，可作为打印使用
Outpdfall		导出当所有数据到PDF,在支持pdf显示的浏览器上可直接显示，可作为打印使用
outuserfile		自定义文件导出类，值为类的具体包名和类名
outcustom		导出时制定义统计列信息,只在excel和pdf时有效
checkrecord		在翻页和排序时自动记录所勾选的记录,该参数为存放记录的控件名和id，默认为表格id加上_record
custombutton		自定义信息，该信息如不为空，会在dbgrid下方的分页信息栏上显示该信息，可用作按钮。
Custom		自定义统计信息栏，可以是任何html元素
airRowClass		根据单元格值，格式化单元格样式，规则为: [cname=列名,值=classid,值=classid] 如: [cname=状态,3=data_row_1,4=data_row_2], [cname=属性,6=data_row_1,5=data_row_2] classid为dbgrid皮肤配置中的css的样式id
Onsuccess	是	执行成功时的回调方法
Onfailure	否	执行失败时的回调方法
Onsubmit	否	在提交数据时调用的方法,该方法返回true继续提交，返回false停止提交
hideselect	否	是否隐藏选择列,默认为false,可选{true|false}
Reload	否	是否开启手动刷新数据功能，默认为false，可选{true|false}
autoRefresh	否	自动刷新的时间间隔，以秒为单位，必须大于0
dataclass	否	自定义数据处理类
inputData	否	绑定输入框数据，JSON格式数据，格式为 [{field:’aa’,value:2,veri:true,show:3},{field:’aa’,value:’cellname’,veri:true,show:’cellname’}] field:为需要绑定的输入框的ID，value为输入框的绑定值，这里可以是dbgrid的列号或是列名，veri表示是否不进行检验，为true表示不验证，默认需要重新验证，show为显示值，可以是dbgrid列号或是列名，其中veri和show可选值
formNull	否	在提交form时检测到选项为空时的提示信息
formConfirmation	否	在提交form时需要在次确认时的提示信息
dataclass	否	自定义数据处理类
isHideHead	否	是否隐藏表头，可选值true|false

isHideBottom	否	是否隐藏表底部信息栏，可选值true|false
editCells	否	可编辑列，需要某一列可编辑时，需要指定编辑的不入框类型，多列以JSON数组传入[{},{}]具体见后面说明
editSubmitColor	否	编辑单元格自动提交时，状态提示背景色，JSON格式如： {loadColor:'#FFFFFF',successColor:'',failureColor:'#FFFFFF'},loadColor表示正在提交时的颜色，successColor表示提交成功时的颜色，一般建议该项留空，在提交成功时以恢复本色，failurColor表示提交失败时的颜色。
endtSubmitAction	否	在编辑某一单元格时，自动提交数据的路径，当该项不为空时，开启编辑自动提交，提交时为编辑单元格所在行的数据加上自定义附加数据，其中doty为固定参数，其值add表示添加一行，这里可能行数据为空，如果有checkbox框，侧默认为-1，其值edit表示编辑某一单元格操作，其值del表示删除某一行操作，传入的行数据根据sql列顺序从1开始，依次为c_1到c_n，表示从第1列到第n列的数据值。如，取第一列的值可以用 request.getParameter('c_1'); 后台返回为json数据格式如” {run:true,ccc:{}}”其中的run指示数据处理是否成功(true表示成功，false表示失败)，ccc为自定义返回的json对象，替换成自己的json对象，更详细说明见本节后面“关于编辑动态提交后台返回说明”
editOnSubmit	否	当endtSubmitAction不空时，在编辑某一单元格时自动提交数据之前调用该方法，该方法返回一个json对像，如{submit:true,data:[]}其中submit为true时，允许提交，为false时禁止提交，data为可选对像，当需要向后台提交附加参数时，可使用data参数，如 data:[{name:'a',value:'a'},{name:'b',value:'b'}]，提交时将data内的数组以name=value键值对的方式提交到后台 传入的参数为该行的对象实例
editOnReturn	否	当endtSubmitAction不空时，在编辑某一单元格时自动提交数据之后，服务器返回信息时调用该方法，传入js方法名，传入的参数有两个，第一个为该行的对象实例，第二个为返回的json对象。

网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com

editRow	否	是否开启行编辑功能，可选值true|false，默认为false,开启后，将在底部信息栏出现两个按钮，分别为添加行，删除行，将允许添加，删除选中行，如果添加，删除按钮需要分别控制，可传入值true,true第一个参数表示控制添加按钮，第二个参数表示控制删除按钮。
editAllSubmitAction	否	提交所有表格的数据路径，当该属性不为空时，信息栏会显示一个保存按钮，点击该按钮时，将表格的所有数据以JSON格式提交到该属性指定的路径，在后台用org.nokatag.dbgrid.NDBGridRequest类中的getEditAll方法获取提交的数据，该方法为静态方法，返回List<List>对象
onload	否	在加载数据时调用的java自定义处理方法，详见本节后面的加载数据时调用外部方法
onloadjs	否	在加载数据返回时调用的javascript自定方法，该属性为javascript方法名
outfilename	否	导出文件时，导出的文件名称（不带扩展名），如果需要在文件名称中自动加上导出的当前时间，可用$[时间格式化字符串]引入，在导出时系统根据$[]中的时间格式化字符串自动替换成指定格式的当前时间，如：信息表$[yyyy-MM-dd]，侧导出的文件名为:信息表2015-03-09
cellbody	否	如果需要自定义单元格样式，用该属性指定单元格样式ID，该ID在noka-config.xml里面通过cellbody节点指定
onend	否	表格成功渲染完成时调用的外部js方法,其用法同onloadjs

➢ **使用实例**

```
<n:DBGrid
    onfailure="NK_bbb"
    onsuccess="NK_aaa"
    outexcelall="true"
    tableformAction="la.jsp"
    selectinput="true"
    pagesize="30"
    checkrecord="wabc"
    formatcell="org.test.T.fa(java.lang.String 开卡日期) as 开卡日期 "
    checkd="checkbox"
    title="测试表"
    id="nk_dbsdd"
    height="400"
```

```
    width="900"
    checkname="wors"
    checkvalue="0"
    descorasc="desc"
    compositor="a"
    sql="select *    from nk_sys_wordbook "
    isaintercept="true"
/>
<script type="text/javascript">
function aa(v){
alert(v.cells(1).innerHTML);
}
</script>
```

里面的org.ak.T.sc是一个位于org.ak下面的名为T的java类。sc表示该类的方。

法。对应的T类源码是样的。需要注意的是这里方法的参数只能是String类型。

```
package org.ak;
public class T {
public String sc(String id,String a){
return id+":"+a;
}
}
```

Formatcell:可用类型有：

java.lang.String

java.lang.Integer

java.lang.Double

javax.servlet.http.HttpServletRequest建议用ServletNokaContext代替，体具使用方法见6.8节。

如这样一条sql 语句 tname like ‘%value%’ and tsex =’1’

在noka里是这样的: tname like ‘%$[tnames,名字,text,dd]%’

#[ano,and^并且!or^或者]# tsex=' $[tesx,性别,list,1^男!2^女]'

这样会在界面成一个录入框 一个条件选择框 和一个 性别下拉选框。

录入框类型可选有:date 生成一个可以选择日期的录入框，如果在该类型设置值为today，则系统会自动以当前日期填充。datetime带日期和时间录入框，time时间录入框。

list 生成一个下拉选框，参数分别为name,显示名称,类型（这里指定为list),值(值^显示值,多个选项用!分隔),宽度(不带单位),下个级联的select名字。如果是采用SQL方式，侧值部分换成estr加密后的sql即可，如下所示：

WNAME = '%$[aaa,类型,list,${n:estr('SELECT TID AS value,TNAME AS text FROM NK_SYS_TREEBOOK')}]%'。

关于list在dbgrid中级联选择实例如下：

```
uid = $[aaf,afff,list,${n:estr('SELECT TID AS value,TNAME AS text FROM
NK_SYS_TREEBOOK')},120,adc]
| #[ano,and^并且!or^或者]#
uname = '$[adc,afff,list,${n:estr('SELECT TID AS value,TNAME AS text FROM
NK_SYS_TREEBOOK where tid=?')},120]'
//第一个list最后一个参数adc为指定的第二个级联的list的名字，如上所示，在第一个list
选择某一个值时，将自动将其选择的值传递到第二个list中，填充sql中的?后更新第二个list
的选项，从而实现多list级联选择功能
```

multi生成一个下拉选框（可多选），参数分别为name,显示名称,类型（这里指定为multi），值(值^显示值，多个选项用!分隔)，宽度（不带单位）。如果是采用SQL方式，侧值部分换成estr加密后的sql即可，如下所示：

WNAME = '%$[aaa,类型,list,${n:estr('SELECT TID AS value,TNAME AS text FROM NK_SYS_TREEBOOK')}]%'。

text生成一个普通的录入框，参数分别为name，显示名称，类型(这里指定为text)，值(空值只需保留后在面的逗号即可)，宽度(不带单位)。

mtext生成一个可用逗号分隔的多个值录入框，参数分别为name，显示名称，

网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com

类型(这里指定为mtext),值(空值只需保留后在面的逗号即可,多个值用!分隔),宽度(不带单位)。在界面查询时，输入多个值用逗号或空格分隔。

hidden生成一个隐藏控件，其控件ID为控件名加上_id，如控件为$[ta,名字,hidden,1]，则控件名为ta，控件id为ta_id，该id可以通过javascript引用，1为该控件的值。

button生成一个弹出选择输入框，其用法与2.21的弹出选择输入框相同，写法如：aaa =’$[aa_n,公司名,button, 200,700,500,${rootpath}/tj.jsp]’其中的200为控件自身的宽度，700为弹出框的宽度，500为弹出框的高度，最后一个为弹出框加载的网页地址，文件的${rootpath}表示虚拟根目录，根据自己具体的网页路径填充该参数，该控件的参数个数不能缺失。在弹出页面上可以通过该控件的第一个参数（即neame属性，该控件的name和id为同一名称）设置其值，如文件所写的实例，可以像下面这样设置控件的值：

```
window.parent.$("aa_n").setValue('1','s');//1为值，s为显示值
window.parent.$("aa_n").hide();//同时关闭弹出框
```

其中af为js方法名，形如function af(a,b);传给后台的值为隐藏控件的值，这里的a为隐藏控件的id，b为显示控件的id。

tree生成一个下拉树形单项选择框，写法如下：

```
WID =($[an_c,部门],tree, ${n:estr('SELECT TID AS ID,TPID AS PID,TNAME AS NAME,TNAME AS TITLE,TLVIE as LEVE FROM NK_SYS_TREEBOOK')},120,2,all])
```

其中an_c为控件的名字，部门是指控件的显示名称，tree是类型（固定值），sql必须使用estr方法进行加密处理，120是控件的宽度(非必须项)，2是控件的初始rootID（同树形菜单的spid，非必须项）。all为控件的级联选择方式（同tree控件）该控件的数据加载路径为ntree[控件名字_id].tre，如上例所示的加载路径为ntreean_c_id.tre

当查询表单需要换行时，在需要换行的地方加上$n（$n两端需要有一个空格）便可实现换行。

网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com

当查询表单需在使用mtext、multi以及tree多项级联选择时，其查询条件请用in查询，如uid in ($[tesx,aaaaa,multi,1^a!2^b,400])，只有mtext支持用like进行模糊查询，如uname like '%$[tesx,aaaa,mtext]%'，在多个值查询时，会自动构建为(uname like ? or uname like ?)形式。

在拖动列宽时，如果列宽小于10时，会自动隐藏该列。

- **自定义数据处理类（dataclass）**

自定义数据处理类是开发人员自己实现数据处理方法，该类需要继承org.nokatag.dbgrid.DataInterface接口，该接口只有一个方法dataRunte，该方法共有6个参数如下所示：

```
public String dataRunte(HttpServletRequest request,HttpServletResponse
response,String sql,String spage,String pagedescorname,String pagedescorasc);
```

sql为控件sql属性所配置的sql语句（已解密），

spage为当前显示的页码（纯数字）

pagedescorname为前端提交的需要排序的列名

pagedescorasc为前端提交的排序类别（分别为asc和desc）

页面的查询表单数据通过传入的request对象获取，获取方法同普通表单对象一样，如reqest.getParameter(“查询表单的名字”)。

自定义数据处理方法返回一个String类型，内部为JSON类型的数据格式如下所示：

```
{
options:{
        pagecont:'2',
        showpage:'1',
        startrow:'1',
        endrow:'2',
        pagesize:'2',
        lastRow:'3'
},rows:[
```


网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com

```
['1','6','wq','2014-10-10 13:11:55.0','1','1','1',' '],
['2','6','q','2014-10-10 13:19:01.0','2','2','2',' ']
]
}
```

pagecont:总页数

showpage:当前显示的是第几页

startrow:当前页的开始行数

endrow:当前页的结束行数

pagesize:页大小，既每页显行的记录行数

lastRow:当前查询条件下的记录总数

rows内为行对象，为javascript的数组对象，一行为一个数组，请注意，数据一律用单引号括起来，数据内部有特殊符号的需要用转义字符。

- **皮肤配置**

皮肤配置位于noka-config.xml文件的dbgrid块。Dbgrid中的id为JSP页面上指定的styleid值。

```
<dbgrid id="1">
<![CDATA[
body,div,tr,td,table{
    font-size:12px;
    }
    .nk_select_div{
        border-top: solid 1px #ccc;
        border-left: solid 1px #ccc;
        border-right: solid 1px #ccc;
        border-bottom: solid 1px #ccc;
        width:600px;
    }
    .nk_dbgrid_div{
        border-top: solid 1px #ccc;
        border-left: solid 1px #ccc;
        border-right: solid 1px #ccc;
```


网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com

```
        border-bottom: solid 1px #ccc;
    }
    .nk_title_div{
        border-top: solid 1px #ccc;
        border-left: solid 1px #ccc;
        border-right: solid 1px #ccc;
        z-index: 100;
        margin: 0;
        overflow:hidden;
    }
    .nk_head_div{
        background: none repeat scroll 0 0 #8F8F9A;
        border-spacing: 1px;
        margin: 0;
        overflow: hidden;
        white-space:nowrap;word-break:keep-all;word-wrap:normal;
    }
    .nk_body_div{
        overflow:auto;
        background: none repeat scroll 0 0 #8F8F9A;
        border-spacing: 1px;
        margin: 0;
        white-space:nowrap;word-break:keep-all;word-wrap:normal;
    }
    .nk_bottom_div{
        overflow:hidden;
        border-top: solid 1px #ccc;
        border-left: solid 1px #ccc;
        border-right: solid 1px #ccc;
    }
    .nk_title_table{
        background-color: #ccc;
        padding: 0;
        margin:0px
    }
    .nk_head_table{
        background-color: #ccc;
        padding: 0;
        word-break:break-all;
        margin:0px;
    }
```


网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com

```
.nk_data_table{
    background-color: #ccc;
    word-break:break-all;
    padding: 0;
    margin:0px;
}
.nk_bottom_table{
    background-color: #ccc;
    padding: 0;
}
.nk_title{
     background: url('${rootpath}/nokatag/tablegrid/images/hrow.gif') repeat-x;
     background-color: #FFFFFF;
     height:24px;
     overflow:hidden;
     word-break:break-all;
}
.nk_head{
     background: url('${rootpath}/nokatag/tablegrid/images/hrow.gif') repeat-x;
     background-color: #FFFFFF;
     height:24px;
     font-size: 12px;
     word-break:break-all;
     padding: 0;
     text-align: left;
}
 /*采用自定义单元格时的div样式*/
 .nk_user_cell_div{
    border: solid 1px #696969;
    margin: 3px;
    background-color:#FFFFFF;
    cursor: pointer;
 }
/* 偶数行 */
.data_cell_0{
     background-color: #FFFFFF;
     height:24px;
     font-size: 12px;
     word-break:break-all;
     padding: 0;
```

```
    }
    /* 奇数行 */
   .data_cell_1{
        background-color: #FFBBFF;
        height:24px;
        font-size: 12px;
        word-break:break-all;
        padding: 0;
        overflow:hidden;
    }
    /*警示行*/
    .data_row_1{
        background-color: #AAABFF;
        height:24px;
        width: 0px;
        font-size: 12px;
        word-break:break-all;
        padding: 0;
    }
     /*警示行*/
    .data_row_2{
        background-color: #FFFFFF;
        height:24px;
        width: 0px;
        font-size: 12px;
        word-break:break-all;
        padding: 0;
    }
    /* 分页信息 */
    .nk_page_cell_1{
       background-color: #FFFFFF;
       cursor: default;
       height:24px;
       background: url('${rootpath}/nokatag/tablegrid/images/hrow.gif') repeat-x;
    }
    /* 图片 */
    .images_date{
       border: 0;cursor: pointer;width: 16px;height: 19px;
       margin-right:-3px;
       vertical-align:-5px;
       vertical-align:middle;
```


网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com

```
}
/* 图片 */
.nk_sub_images{
    border: 0;cursor: pointer;width: 16px;height: 16px;
    margin-right:-3px;
    vertical-align:-4px;
}
/* 图片 */
.nk_images_but{
    border: 0;cursor: pointer;width: 16px;height: 16px;
    margin-right:0px;
    vertical-align:-4px;
    border:1px solid #333333;
}
 /* 排序图片 */
.nk_asc_desc_images{
    border: 0;cursor: pointer;width: 8px;height: 4px;
    margin-right:0px;
    vertical-align:0px;
}
.nk_cell_menu{
    z-index: 999999;
}
.nk_table_menu{
    background-color: #FFFFFF;
    border-bottom: solid 1px #ccc;
}
.nk_menu_td_1{
    border-top: solid 1px #ccc;
    border-left: solid 1px #ccc;
}
.nk_menu_td_2{
    border-top: solid 1px #ccc;
    border-right: solid 1px #ccc;
}
.nk_input_check{
    margin-right:-2px;
    vertical-align:-2px;
}
.nk_sub_td{
    word-break:break-all;
```

```
        padding: 0;
        background-color: #FFFFFF;
        border-top: solid 1px #ccc;
        border-left: solid 1px #ccc;
        border-right: solid 1px #ccc;
    }
    /*采用自定义单元格时的div样式*/
    .nk_user_cell_div{
        border: solid 1px #696969;
        margin: 3px;
        background-color:#FFFFFF;
        cursor: pointer;
    }
    /*查询按钮样式*/
    .select_button{
    }
    /*查询录入框样式*/
    .select_input{
      border: 1px solid #AFAFAF;
      height:19px;
      vertical-align:middle;
    }
    /*拖动列宽时的线*/
    .nk_resizeMarker{
      width: 1px;background-color: #000000;
    }
    /*自定义统计栏*/
    .nk_custom_div{
        height:30px;
    }
    /*子表表体样式*/
    .subgrid_table{
        background-color: #44B7A8;
        padding: 0;
        width: 100%;
    }
    /*子表表头行样式*/
    .subgrid_head_tr{
     b
    }
    /*子表表头单元格样式*/
```


网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com

```
    .subgrid_head_td{
     background-color: #B4E1DC;
    }
    /*子表数据偶数行样式*/
    .subgrid_cell_tr_0{
    }
    /*子表数据奇数行样式*/
    .subgrid_cell_tr_1{
    }
    /*子表数据偶数行单元格样式*/
    .subgrid_cell_0{
     background-color: #FFFFFF;
    }
    /*子表数据奇数行单元格样式*/
    .subgrid_cell_1{
     background-color: #FFCCCC;
    }
]]>
</dbgrid>
```

子表样式

```
<subgrid id="1">
<![CDATA[
你好啊名字：${0}
]]>
</subgrid>
```

关于子表的应用说明如下：

1、直接显示主表某几列时，需要设置subtable="2,3"和subtabletype="cell"两个属性，其中subtable属性为主表的列顺序号，多个列用逗号分隔，subtabletype这里设为cell表示直接取主表列。当需要自定义样式时，需要设置subtablestyleid，该属性的值为noka-config.xml中的subgrid的id值。Subgrid里面的用${1}来应用主表列，其中1表示主表列序列号，从0开始。功能列（如选择、序号、子表）列不作为序列号计算。

2、sql方式加载时需要同时设置subtable="select * from nk_sys_wordbook where wid=?"、subtabletype="sql"、subtablestyleid="0"三个属性，此时的

网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com

subformatcell可用于格式化子表的数据，使用方法同主表的formatcell。在sql模式下，subtable为sql语句，参数以?号代替，subtablestyleid为要应用给sql参数的主表列序列号，从0开始。

3、自定义方法加载子表，继承于org.nokatag.dbgrid.SubGrid接口实例

```
public class TestGrid implements SubGrid {
 public String getBody(Integer i, Map<String, Object> map,List<String> Cellname) {
String datarow="";
Integer row=i;//行号
//---------取得行所有数据------------------------
Iterator it = map.entrySet().iterator();
while (it.hasNext()) {
Map.Entry entry = (Map.Entry) it.next();
String key = entry.getKey().toString();//列名
String value = entry.getValue().toString();//值
datarow+=key+"="+value+",";
 }
///--------取得所有列名------------
for(String cell:Cellname){
datarow+=","+cell;
}
String sn="获得数据："+datarow;
return sn;
}
```

此时需要设置subtable="org.noka.function.TestGrid"和subtabletype="class"两个属性，其中subtable为自定义类。

关于ajax数据操作说明，该标签会自动生一个外部刷新数据的javascript方法，其方法名为：标签的ID _reLoadData，如标签ID为noka，则该方法为noka__reLoadData，不需要任何参数，调用该方法，既可刷新数据，注意，该方法将重置分页、排序两个属性为初始状态。

在onsuccess和onfailure属性均不为空的情况下，该标签会自动生成一个名为：标签的ID _ajaxSubmitDataForm方法（$(id).ajaxSubmitDataForm()等同），可调用该方法提交dbGrid里面的表单到服务器，根据返回状态自动调用

onsuccess和onfailure里面的方法，注意，该方法只能提交数据表单，不包括查询表单和分页表单。

双击和单击方法的使用具有相同的参数传入，以下是方法实例：

```
function detail(v){
  //i为该行的列号，从0开始，该方法用于取出双击或单击行的某个单元格数据
  alert($(v.cells[i].id+'_div').innerHTML);
}
```

关于outuserfile属性说明：

Outuserfile属性的类需要继承org.nokatag.dbgrid.NDBOutFileInterface接口，该接口有三个方法，分别对应于csv，excel，pdf三类文件的导出自定义方法，在方法内，自己实现文件的生成工作，从而实现自己控制文件的样式。在方法内部可以通过request对象获取dbgrid页面的查询录入项的值，达到更灵活的传值处理，该类实例看起可能像下面这样(下类的outuserfile属性设为：org.noka.test. Outfile)：

```
package org.noka.test;

import java.io.OutputStream;
import java.nio.charset.Charset;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;
import java.util.List;

import javax.servlet.http.HttpServletRequest;

import org.noka.csv.CsvWriter;
import org.noka.itextpdf.text.Document;
import org.noka.itextpdf.text.Element;
import org.noka.itextpdf.text.Font;
import org.noka.itextpdf.text.PageSize;
import org.noka.itextpdf.text.Paragraph;
import org.noka.itextpdf.text.pdf.BaseFont;
import org.noka.itextpdf.text.pdf.PdfPTable;
import org.noka.itextpdf.text.pdf.PdfWriter;
```

```
import org.nokatag.jxl.Workbook;
import org.nokatag.jxl.write.Label;
import org.nokatag.jxl.write.WritableSheet;
import org.nokatag.jxl.write.WritableWorkbook;
import org.nokatag.system.BugInfoWrint;
import org.nokatag.system.Concest;

public class Outfile implements NDBOutFileInterface{

//=============CSV文件================================================
public void outCsvFile(OutputStream os, List<List<String>> row,List<String> cellName,
HttpServletRequest request) {
        CsvWriter wr = null;
        try{
            String showfide=request.getParameter("showfide");
            wr =new CsvWriter(os,',',Charset.forName(Concest.ENCODING));
            //-------------------写入主标题栏----------------------
            os.write(new byte[]{(byte)0xEF,(byte)0xBB,(byte)0xBF});
        String nokatag_excel_title =  request.getParameter("nokatag_excel_title");
            if(nokatag_excel_title!=null && nokatag_excel_title.trim().length()>0)
                wr.writeRecord(new String[]{nokatag_excel_title});//写入标题
            //-------------------写入列标题栏-----------------------
            List<String> titles = new ArrayList<String>();
            for (int s = 0; s < cellName.size(); s++) {
                if(isShowCell(showfide,s)){
                    String tite = cellName.get(s);
                    titles.add(tite);
                }
            }
             wr.writeRecord(titles);//写入标题
            //-------------------写入数据-------------------------------
            for(int r=0;r<row.size();r++){//行
                List<String> m=row.get(r);
                List<String> cell = new ArrayList<String>();
                for(int s = 0; s < m.size(); s++){//列
                    if(isShowCell(showfide,s)){
                        cell.add(m.get(s));
                    }
                }
                wr.writeRecord(cell);
            }//行
```


网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com

```
        } catch (Exception e1) {
            BugInfoWrint.Print(e1);
        }finally{
            try{
                if (wr != null)
                    wr.close();
                if (os != null)
                    os.close();
            }catch (Exception e2) {
                BugInfoWrint.Print(e2);
            }
        }
    }
   //======Excel文件===================================================================
    public void outExcelFile(OutputStream os, List<List<String>> row,List<String>
cellName, HttpServletRequest request) {
        WritableWorkbook wwb=null;
        WritableSheet ws=null;
        try{
            String showfide=request.getParameter("showfide");
            wwb = Workbook.createWorkbook(os);// 创建可写工作薄
            ws = wwb.createSheet("sheet1", 0);// 创建可写工作表
            //-------------------写入主标题栏----------------
            int start=0;
            String nokatag_excel_title =
request.getParameter("nokatag_excel_title");
            if(nokatag_excel_title!=null && nokatag_excel_title.trim().length()>0){
                ws.mergeCells(0, 0, showfide.split(",").length/2-1, 0);// 列总数
                Label labelCF0 = new Label(0, 0, nokatag_excel_title);// 写入主标题
                ws.addCell(labelCF0);// 将Label写入sheet中
                start++;
            }
            //-------------------写入列标题栏----------------
            int a=0;
            for (int s = 0; s < cellName.size(); s++) {
                if(isShowCell(showfide,s)){
                    String tite = cellName.get(s);
                    Label labelCF0i = new Label(a, start, tite);// 写入列标题
                    ws.addCell(labelCF0i);// 将Label写入sheet中
                    a++;
                }
```


网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com

```
            }
            start++;
            //--------------------写入数据-----------------------
            for(int r=0;r<row.size();r++){//行
                List<String> m=row.get(r);
                int c=0;
                for(int s = 0; s < m.size(); s++){//列
                    if(isShowCell(showfide,s)){
                    //-------------写入---------------------
                        try {
    SimpleDateFormat bartDateFormat = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
                            Date date = bartDateFormat.parse(m.get(s));
                            org.nokatag.jxl.write.DateTime labelDT = new
org.nokatag.jxl.write.DateTime(c, r+start, date);
                            ws.addCell(labelDT);
                        } catch (Exception seb) {
                    Label labelCF0i = new Label(c, r+start,m.get(s));// 写入标题
                            ws.addCell(labelCF0i);// 将Label写入sheet中
                        }
                        c++;
                    }
                }
            }//行
            //-----------------------------------------------------------
            wwb.write();
        } catch (Exception e1) {
            BugInfoWrint.Print(e1);
        }finally{
            try{
                if (wwb != null)
                    wwb.close();
                if (os != null)
                    os.close();
            }catch (Exception e2) {
                BugInfoWrint.Print(e2);
            }
        }
    }
    //=======================PDF文件============================
    public void outPDFFile(OutputStream os, List<List<String>> row,List<String>
cellName, HttpServletRequest request) {
```


网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com

```
        Document document = null;
        try{
            document = new Document(PageSize.A4);
            PdfWriter. getInstance(document,os);
            document.open();
            BaseFont bfComic0= BaseFont.createFont("STSongStd-Light",
"UniGB-UCS2-H", false);
            Font title_font = new Font(bfComic0, 16);
            Font body_font = new Font(bfComic0, 9);
            //---------------计算列数--------------------
            String showfide=request.getParameter("showfide");
            Integer CellCount= 0;
            {
                for (int s = 0; s < cellName.size(); s++) {
                    if(isShowCell(showfide,s)){
                        CellCount+=1;
                    }
                }
            }
            //---------------大标题-----------------------------------------
            {
        String nokatag_excel_title =  request.getParameter("nokatag_excel_title");
            if(nokatag_excel_title!=null && nokatag_excel_title.trim().length()>0){
                Paragraph paragraph = new Paragraph(nokatag_excel_title,title_font);
                    paragraph.setAlignment(Element.ALIGN_CENTER);
                    paragraph.setSpacingAfter(20);
                    document.add(paragraph);
                }
            }
            CellCount=CellCount<1?1:CellCount;
            PdfPTable table = new PdfPTable(CellCount);//建立表格
            table.setWidthPercentage(99);
            //----------标题栏-----------------------------------------
            {
                for (int s = 0; s < cellName.size(); s++) {
                    if(isShowCell(showfide,s)){
                        String tite = cellName.get(s);
                        table.addCell(new Paragraph(tite,body_font));
                    }
                }
            }
```


网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com

```
            //--------------内容----------------------------------------------------
            {
                for(int r=0;r<row.size();r++){//行
                    List<String> m=row.get(r);
                    for(int s = 0; s < m.size(); s++){//列
                        if(isShowCell(showfide,s)){
                            table.addCell(new Paragraph(m.get(s),body_font));
                        }
                    }
                }
            }
            document.add(table);
            //--------------写入日期----------------------------------------------
            {
                SimpleDateFormat df = new SimpleDateFormat("yyyy-MM-dd HH:mm");
                Paragraph paragraph = new Paragraph(df.format(new
Date()),body_font);
                paragraph.setAlignment(Element.ALIGN_RIGHT);
                paragraph.setSpacingAfter(20);
                document.add(paragraph);
            }
        }catch(Exception e1){
            BugInfoWrint.Print(e1);
        }finally{
            try{
                if (document!= null)
                    document.close();
                if(os!=null)
                    os.close();
            }catch (Exception e2) {
                BugInfoWrint.Print(e2);
            }
        }
    }
    /**
     * 判断是否在显示列里面
     * @param showcell
     * @param cell
     * @return
     */
    private  Boolean isShowCell(String showcell,int cell){
```


网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com

```
        if(showcell==null)
            return false;
        if(showcell.indexOf(","+cell+",")!=-1)
            return true;
        return false;
    }
}
```

方法传入参数说明：

OutputStream os 系统自生成的文件输出对象，直接使用。

List<List<String>> row 传入的数据对像（已经格式化之后的数据）

List<String> cellName 数据表的列名，包含所有例，要过滤隐藏列需要借助于request里面获取的showfide对象。

HttpServletRequest request 页面传入的request对象，可以获取dbgrid页面生成的查询表单里面的值。

➢ **关于外部自定义查询表单说明：**

在查询时，如果需要在dbgrid外部自己建立查询表单时，selectfiled属性同样需要配置查询条件，selectinput属性配置外部表单（form或div等html元素）的id属性，然后提交查询表单时，直接调用dbgrid的_reLoadData()($(id).reload()方法等同)方法。

如以下所示：

```
<!—自定义查询表单，也可以使用form、table等元素—>
<div id="query_from_id">
名称:<n:InputText name="wdname" id="wdname_id"></n:InputText>
<n:select id="ano" name="ano"
json="[{value:'and',text:'并且'},{value:'or',text:'或者'}]"/>
备注:<n:InputText name="d_c" id="d_c_id"></n:InputText>
<input type="button" name="sr" value="查询" onclick="wordDb_reLoadData()"/>
</div>
```

以上自定义查询表单对应的dbgrid属性配置为(注：dbgrid的ID为wordDb)：

selectfiled="名称 like '%$[wdname,名称]%' | #[ano,and^并且!or^或者]# 备注 like '%$[d_c,备注]%'"

selectinput="query_from_id"

➢ **关于表格编辑**

表格编辑是通过editCells属性进行设置，传入一个JSON数组对象，数组的每一项为对应某一可编辑列，具体如下：

{cell:'列名',cfg:{}}，其中cell为列名，cfg为编辑配置对象，cell可以为列名或是列序号，如：

```
editCells="[{cell:'MENUID',cfg:{type:'input',allownull:false}},{cell:'MENUID2',cfg:{type:'input',allownull:false}}]"
```

1、普通录入框的配置

cfg:{type:'input',allownull:true,chtype:'',veri:'',msg:'' }

type:输入类型，这里固定为input

allownull:是否允许为空，传入值true|false,默认为true，可选

chtype:'' 验证类型，传入值同input标签，可选

veri:验证具体方法或正则表达式，传入值同input标签，可选

msg:验证失败时，提示消息，传入值同input标签，可选

2、下拉选择框

cfg:{type:'select',vid:'',json:'',sql:'',allownull:true }

type:输入类型，这里固定为select

vid:下拉框选择时，其实际值（下拉框分为显示值和实际值）所对应的列，传入参数列名，在选择时，其所在行对应的列将填充为实际值

json:选择框的json数据，传入值同select标签

网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com

sql:选择框的sql数据，需要用${n:estr(sql)}进行sql加密

allownull:是否允许为空，传入值true|false,默认为true，可选

3、日期、时间选择框

cfg:{ type:'dateselect',allownull:false,dateFmt:'yyyy-MM-dd'}

type:输入类型，这里固定为dateselect

allownull:是否允许为空，传入值true|false,默认为true，可选

dataFmt:时间格式化，传入值同时间选择框标签

4、树形下拉选择框

cfg:{ type:'treeselect',allownull:true,vid:'ID',sid:'-1',checktype:'all',sql:'${n:estr('SELECT TID AS ID,TPID AS PID,TNAME AS NAME,TNAME AS TITLE,TLVIE as LEVE FROM NK_SYS_TREEBOOK')}'}

type:输入类型，这里固定为treeselect

allownull:是否允许为空，传入值true|false,默认为true，可选

vid:下拉框选择时，其实际值（下拉框分为显示值和实际值）所对应的列，传入参数列名，在选择时，其所在行对应的列将填充为实际值

sid:树形菜单的父ID，默认为sql查询到的最小ID作为该值

checktype:树形菜单选择时的级联选择关联，值同下拉树形选择框

sql:树形菜单的sql，这里同下拉框一样，必须使用${n:estr(sql)}进行加密

5、下拉多项选择框

cfg:{type:' selectmulti', vid:'',json:'',sql:'',allownull:true }

type:输入类型，这里固定为selectmulti

vid:下拉框选择时，其实际值（下拉框分为显示值和实际值）所对应的列，

传入参数列名，在选择时，其所在行对应的列将填充为实际值

json:选择框的json数据，传入值同select标签

sql:选择框的sql数据，需要用${n:estr(sql)}进行sql加密

allownull:是否允许为空，传入值true|false,默认为true，可选

- **关于编辑动态提交后台返回说明**

开启endtSubmitAction功能以后，在编辑行单元格时，会动态向该路径提交编辑单元格所在行数据，后端以JSON格式返回，格式如下：

```
{run:true,rdata:[{cell:'c1',value:'DDDD'},{cell:'check',value:'22333'}],ccc:{}}
```

run参数为提交成功标识，true表示提交成功，false表示提交失败。

rdata参数为行更新数据（非必须），cell为列名名列序号，value为单元格的值，其中cell为check时特指该行的选择框对象。

ccc为任意自定义对象，noka tag自身不处理该对象，用于在调用外部自定义方法时，由业务逻辑自己处理。

- **关于行对象实例**

在行双击、单击、行编辑提交、提交返回方法中传入的行对象时，该对象为表格中操作行的行对象，除具备普通表格行所具有的方法以外，还具有以下方法（以传入参数row为例）：

row. getCheck() 获取该行选择框实例(checkbox或radio)

row. getRowData()获取行数据，返回的为一个数组

row. getCell(index)获取该行某一单元格对象,index为列号或列名，列号从1开始

单元格对象具有以下方法

getValue（）获取该单元格的值

网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com

setValue（value）设置该单元格的值

setStyle（）设置该单元格的style属性，该方法继承于prototype，具体使用方法请参照prototype手册

更多单元格附加方法请参照prototype手册。

➢ **关于表格数据一次性全部提交时，后台获取数据**

一次性全部提交数据是采用的JSON格式向台后提交数据，参数名称为固定值nokardata,后台建议通过内置的工具类进行获取并解析，该类为org.nokatag.dbgrid.NDBGridRequest，通过方法getEditAll()获取解析后的数据，如下例所示：

```
List<List> person = NDBGridRequest.getEditAll();//获取并解析提交的数据
for(List<String> row:person){//循环每一行数据
    for(String cell:row){//循环一行中的单元格
            System.out.println(cell);//每一个单元格的数据
   }
}
```

➢ **加载数据时调用外部方法**

在加载数据时可以通过onload和onloadjs两个主法分别调用java和javascript外部方法，onload属性传入java的类，该类需要继承org.nokatag.dbgrid.OnDBGridDataLoadInterface接口，在该类内部返回JSON对象的字符串到前端,前端通过onloadjs方法获取返回的JSON对象,如下例所示：

```
<n:DBGrid
selectfiled="MENUNAME LIKE '%$[aa,cc]%'"
onload="org.test.OnDBGridDataLoad"
onloadjs="cd"
width="100%"
height="400"
sql="select * from nk_sys_menu"
id="sf"/>
<!-- org.test.OnDBGridDataLoad为实现了OnDBGridDataLoadInterface接口的处理类-->
<!-- cd为js在加载数据返回以后调用js方法-->
<script type="text/javascript">
        function cd(a,b){//在加载数据返回之后调用的外部js方法
```

```
            //参数a为查询表单的查询值，数组对象[{name:'',value:''}]
            alert(a[0].name);
            //参数b为系统返回的对象，其中userData为自定义类OnDBGridDataLoad返回的对象，options是dbgrid的分页选项信息对象，rows表格数据，二维数组[[],[]]
            alert(b.userData.at);
            return true;//返回true继续执行数据展现工作，false禁止执行数据展现工作
        }
    </script>
//后台方法如下：
public class OnDBGridDataLoad implements OnDBGridDataLoadInterface{
    public String OnDataLoad(String sql, List<Map<String, String>> person) {
        System.out.println("sql:"+sql);//传的原始sql，同dbgrid中的sql属性设置的sql
//查询表单传回的查询参数,map中的name为页面查询表单中控件的name，value为对应的查询值
         for(Map<String,String> m:person){
              System.out.println("name:"+m.get("name")+"="+m.get("value"));
        }
        return "{'at':'n9339'}";//返回JSON对象
    }
}
```

➢ **关于自定义单元格**

自定义单元格是以方块列表的方式显示表格，此时的Grid的其它功能任然有效，只有单击和双击行事件有所区别，该功能是通过属性cellbody来控制，该属性指示的是配置文件的单元格模板定义，模板定义如下所示：

```
<cellbody id="1">
<![CDATA[
     <div style="width: 100px;height: 60px;" id="${usercellid}_rowdiv">
     <div>${2}</div>
     <div>${3}</div>
     </div>
]]>
//其中${2}表示该处是一个变量，里面的2表示列号，grid会用对应的列值替换该处的内容
//行ID组成规则为nk_[gridid]_dcell_行号,其中[gridid]为grid的id值，行号为实际的行号
//${usercellid}为系统内置变量，指示每一行自动生成的唯一行ID
//${row_no_id}为系统内置变量，指示每一行自动生成的行序号
//${row_id}为系统内置变量，指示每一行自动生成的行ID，不包含序号
```

当采用自定义单元格时，所对应的行单击事件和行双击事件，传入的参数为该行的行数据，数据类型为一个JavaScript数组，数组下标对应列序号。

网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com

➢ 关于表格JS方法

dbgrid对象自有javascript方法说明

reload()重新加载表格数据

ajaxSubmitDataForm()以ajax方式提交表格数据

getFormSelect()获取表格选中的数据，以数据形式返回

setFormSubmitInfo(formNull, formConfirmation)设置表格from的formNull和formConfirmation两个属性。

setValue()设置表格的选项值，该方法为一个字符串参数，多个值用逗号分隔，如setValue(‘1,2,3’)

getData()获取表格的所有数据，返回一个二维数组

getSelectRow()获取当前选中的行对象，多选时选回第一行

insertDataRow(rowdata)插入一行，rowdata为行数据，是一个一维数组，可以为空，该方法返回插入的行对象。

delSelectRow()删除选择的行

setOutTile(a)设置导出的大标题，同属性outtitle

setTitle(a)设置标题信息，该方法必须在设置title属性之后使用

setPageSize(a)设置页大小

setCustom(a)设置custom栏内容，该方法必须在设置custom属性之后使用

setOutCustom(a)设置导出统计栏内容，只在excel和pdf导出时有效

setCellVisible(index,v)设置某例是否显示,index为列号或列名，v为true表示显示，false表示隐藏

ajaxSubmitAllEdit()提交表格的所有数据到后台，提交路径为

网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com

editAllSubmitAction属性所指定的路径，也可以通过参数指定其提交路径，如ajaxSubmitAllEdit(‘${rootpath}/a.jsp’);

setEditCell(cellEdit);设置可编辑列，参数为一个数组，同editCells属性

ReCreateDbgrid(sql,cell)根据传入的sql重新加载表格，cell为列选项可以为空，其值为一个字符串，同cells属性，关于表格动态重新加载实例如下：

```
function rdbgrid(){//以下实例中’ sf’ 为DBGrid的id值
 var cellMode= ’width:100,show:1|width:100,show:1’;//列选项
 var sql =’${n:estr("select * from nk_sys_userinfo")}’;//动态sql需要用estr加密
 var cellEdit =[{cell:’USID’,cfg:{type:’dateselect’,allownull:false}}];//可编辑
列,cell可用列序号或列名
 $(’sf’).setEditCell(cellEdit);//根据需要，在重新加载sql时，可以重新设置可编辑列
 $(’sf’).ReCreateDbgrid(sql);//重新加载表格，也可以用
 $(’sf’).ReCreateDbgrid(sql,cellMode);表示在加时重新设置列选项
}
```

3.2 统计图表

统计图表是用flash生成各种数据统计图表，数据采用xml做为数据源。共22种统计图表。包括常见的饼图，柱状图，线图等。

➢ **属性说明**

属性	是否必须	说明
id	是	图表的id
xmlpath	否	Xml数据文件路径，其中可用$[rooturl]表应用根目录
sfun	否	直接调用服务器的方法
type	是	图表类型，各种类型及对应图表见上面图例
width	否	图表宽度
height	否	图表高度
pars	否	传到服务器处理的自定义参数

➢ **Xml文件加载例用实例**

```
<n:statchart
type="Pie3D"xmlpath="$[rooturl]/exle/Pie3D.xml" id="column3d_id">
</n:statchart>
```


网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com

Xml文件见附件。特别注意，Xml格式需要采用强验证格式，既结尾必须是<set name='France' value='7'></set>格式，部分浏览器不能很好的识别<set name='France' value='7'/>这种写法，同时该标签支持直接在标签内写XML数据文件，实例如下：

```
<n:statchart type="Pie3D" id="column3d_id">
<graph showNames='1' decimalPrecision='0' >
<set name='中国' value='20'></set>
<set name='France' value='7'></set>
<set name='India' value='12'></set>
<set name='England' value='11'></set>
<set name='Italy' value='8'></set>
<set name='Canada' value='19'></set>
<set name='Germany' value='15'></set>
</graph>
</n:statchart>
```

➢ **服务器方法加载例用实例**

```
<n:statchart
type="Pie3D" sfun="org.noka.charts.ChartsOut.getBody(java.lang.String a)"
id="column3d_id">
</n:statchart>
```

其中org.noka.charts为类所在的包，ChartsOut为类名，getBody为类中的方法，该方法需要返回一个java.lang.String类型，java.lang.String a为参数，参数的写法是：参数类型 参数名称，参数类型 参数名称。参数类型与参数名称之间有且只有一个英文空格,参数名称要与后面的pars中的名称对应，该方法的取值为pars中的值或是javascript调用时传入的值。

pars为json数据，格式为:[{name:'aa',value:'a'},{name:'bb',value:2}]

同时该控件拥有一个javascript方法，名为LoadData(pars)，传入的参数格式同pars，也可以不传入参数，如$('ac').LoadData();这里假设控件ID为ac。

加载服务器方法的路径为：inputtextcheck[id].statchart，其中[id]为控件的ID值。

该类的原型为：

```
package org.noka.charts;
public class ChartsOut{
public String getBody(String className){
StringBuffer sb=new StringBuffer();
sb.append("<graph showNames='1' decimalPrecision='0' > \n");
sb.append(" <set name='"+className+"' value='20' ></set> \n");
sb.append(" <set name='France' value='7' ></set> \n");
sb.append(" <set name='India' value='12' ></set> \n");
sb.append(" <set name='England' value='11' ></set> \n");
sb.append("</graph>\n"); return sb.toString();
}
}
```

图型和类形对应关系见附录一。

XML实例见xml文件夹。

3.3 取单行数据标签(DBDataRow)

DBDataRow标签是通过传入的SQL取出第一行数据，该标签通常在显示某一行详细信息的时候使用，取出的行数据对象以map形式存和在request对象中，其request中的变量名通过val设定。

- **属性说明**

属性	是否必须	说明
val	是	定设的变量名称
sql	是	sql语句

- **使用实例**

```
<n:DBDataRow val="va" sql="select * from nk_sys_menu"></n:DBDataRow>
 ${va.menuname}
```

上实例中va是通过val设定的变量，menuname是数据表中的列字段，列字段将统一全部转换成小写之后作为map的key。

网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com

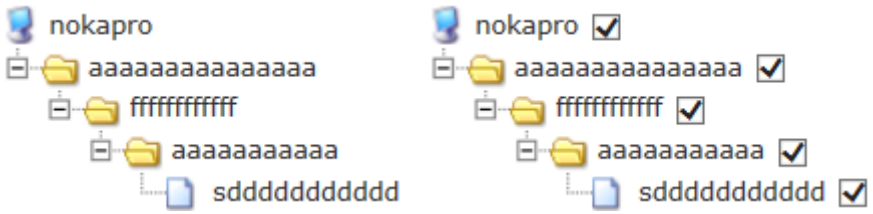
第4章 菜单系列标签

4.1 树形菜单

tree生成一个树形菜单。该标的ajax加载路径为：ntree[ID].tre，其中[ID]为控件的实际ID。

可选择自带选择框，根父ID取值为所有父ID当中最小的一个。也可通过spid指定其父id，同时会生一个与该控件同名，同id的input框，里面记录了所选择的值，在级联选择时，会将没有加载的节点值，一并记录，在采用用分级加载（level属性小于3时）建议用该input操作其值。

➢ 实例图



➢ 属性说明

属性	是否必须	说明
id	是	控件的id，可通过该id像访问普通input控件一样访问里面的值
sql	是	sql语句
url	否	指定一个全局的连接url
target	否	指定一个全局的连接目标
formatcell	否	格式化列，该功能只能格式化name，title两个字段，其它自定义字段不能格式化 格式化列,如需要对某列的值进行格式化后输出，可以这样写：包名.类名.方法名(java类型 列名1,java类型 列名n, java类型 ’常量’) as 列名。多个列用|相隔。这里的类为自定义的普通类。
pramevar	否	指一个连接需要携带的参数，由sql字段产生

网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com

spid	否	指定一个父节点id，用于开始的root节点
level	否	预加载的节点深度，默认加载2级
titleonclick	否	点击标题时，调用的方法
nodeonclick	否	点击节点前面的图标（展开或关闭节点）时调用的方法
name	否	控件的name
checktype	否	在多选时，节点间的级联关系 级联选项，可选值｛all｜next｜previous｜nomuch｝，all：全部级联选择，nex：向下级联选择，previous：向上级联选择，nomuch：非级联多选默认为单选。
titlecliek	否	节点标题是否响应点击事件，可选值{true|false}默认为true，表示响应
checkboxonclick	否	点击选择框调用的方法，只在设置了checktype属性之后才能启用
value	否	控件的初始值
readonly	否	是否只读，可选项true
checkname	否	选择框的名字
updServerFunction	否	调用后台方法或sql执行拖动节点后的修改操作，输入值为后台某个实现了org.noka.tree. NTreeMoveNode接口的类，或是以update开始的sql语句
onupdsubmit	否	在拖动节点以后，向后台提交修改之前调用的js方法
onupdend	否	在拖动节点以后，后台执行完修改操作后返回时调用

➢ **使用实例**

```
<n:tree
checktype="all"
id="sfd"
level="10"
sql="select tid as id,tpid as pid,tname as name,tname as title,tlvie as
leve,ttype from nk_sys_treebook"
titleonclick="snp"/>
 <script type="text/javascript">
  function snp(nod){
      alert(nod.ttype);
  }
```


网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com

```
</script>
```

formatcell:可用类型有：

java.lang.String,java.lang.Integer,java.lang.Double,

javax.servlet.http.HttpServletRequest

在sql中，id,pid,name,title,leve为必须的字段，其中id,pid指定了树形关系，nam和title用于显示在界面上，leve为树形的级别号(leve建议生成规则为父级节点的leve加上父级节点的ID作为本级的leve)，用于指定其下级节点，如果sql时面指定了url、target属性则优先使用sql里面的设置，其数据看起来可如下：

TID	TPID	TNAME	TTEXT	TTYPE	TPX	TUSER	TDATE	TURL	TLVIE
1	-1	nokapro	(Null)	7	1	(Null)	(Null)	(Null)	0_-1
14	1	ddddddddddd	sss	7	1	(Null)	(Null)	(Null)	0_-1_1
28	14	qwwqwq	sss	7	1	(Null)	(Null)	(Null)	0_-1_1_14
29	28	qwww	sss	7	1	(Null)	(Null)	(Null)	0_-1_1_14_28
30	14	uuuuuuuuuuu	sss	7	1	(Null)	(Null)	(Null)	0_-1_1_14
39	29	3333333333	sss	7	1	(Null)	(Null)	(Null)	0_-1_1_14_28
40	39	eeeeeeeeeee	sss	7	1	(Null)	(Null)	(Null)	0_-1_1_14_28

该控件会自动生成一些供外部调用的javasctip方法，用于控制该控件的数据操作，包括刷新数据，指定置，打开指定节点。

id_reLoadData：刷新指定节点的数据，其参数为指定节点的值，id用实际的控件id值替换，如：

```
id_reLoadData( ‘3’ );
```

id_setvalue ：设置控件的值，多个值用逗号分隔。

```
id_setvalue (values)
```

其中id为该标签的id，values为需要设置的值，多个值用逗号分隔

例如，如果标签的id设置的为aa,需要向该标签设的置为2，则应该这样写

```
aa_setvalue(‘2’) 或aa_setvalue(‘2,3,4’)
```

id_opennode ：打开指定的节点，如果它的父节点未被打开，会依次打开其父节点，注：该功能在使用之前，必须确保所打开的节点已经加载完成，否而不

会生效。参为要打开的节点值。

titleonclick ：在点击标题时，调用的javascript方法名，指定期属性时只写方法名。在调用时会传入节点对象，包括sql内的所有字段，其字端名全转换成小写，如sql中有一个字段TTYPE（不区分大小写），则访问如下：

```
//titleonclick=”snc” 属性这样设置
function scn(node){
alert(node.ttype);
}
```

nodeonclick，checkboxonclick两个属性使用方法同上。Titleonclick以及nodeonclick两个方法传入的node对象还额外包含属性名为selfid的属性，其值为该控件的自身ID，checkboxonclick方法传入的node对象额外包含名为selfid和checked两个属性，selfid指该控件自身的ID，checked属性为该节点的check框的选中状态，为true表示选中，为false表示未选中，该方法需要返回一个boolean类型的值，返回true表示继续执行余下下动作，返回false表示不执行余下的动作。

关于拖动节点操作

有两种方式来持久化节点移动操作，一种是直接写update的sql语句，一种是自己实现修改操作（需要继承org.noka.tree.NTreeMoveNode接口）。

```
//sql 方式实现
updServerFunction="UPDATE nk_sys_treebook set tpid =? ,TLVIE=? WHERE tid =?"
其中 tpid 为父亲节点 id,tlvie 为 leve，tid 为节点当前 id，顺序不能更换
//自定义方式
updServerFunction="org.test.TreeMoveNode"
//TreeMoveNode 的实现实例
public class TreeMoveNode implements NTreeMoveNode{
    public String MoveNode(String path,String nodes) {
        StringBuffer body = new StringBuffer();
        Connection con = ServletNokaContext.getConnection();
        try{
            Gson gson = new Gson();
            List<Map<String,String>> nodeList = gson.fromJson(nodes,new
```


网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com

```
TypeToken<List<Map<String,String>>>(){}.getType());
            try{
                con.setAutoCommit(false);
                PreparedStatement msta = con.prepareStatement("UPDATE
nk_sys_treebook set tpid =? ,TLVIE=? WHERE tid =?");
                for(Map<String,String> m:nodeList){
                    msta.setLong(1,Integer.parseInt(m.get("pid")));
                    msta.setString(2,m.get("leve"));
                    msta.setLong(3,Integer.parseInt(m.get("id")));
                    msta.addBatch();
                }
                msta.executeBatch();
                con.commit();//提交事务
                msta.close();
                con.setAutoCommit(true);
            }catch(Exception se){
                con.rollback();//回滚事务
            }
            body.append("{\"code\":true,\"msg\":\"\"}");//以 json 对像返回
    return body.toString();
        }catch(Exception se){
            BugInfoWrint.Print(se);
        }
        body.append("{\"code\":false,\"msg\":\"update false\"}");/     return
body.toString();
    }
}
```

onupdsubmit在提交到后之前调用的js方法，该方法需要返回一个布尔值，true表示继续执行提交动作，false表示放弃执行提交动作。该方法传入两个参数，一个是源节点（被移动节点）对象，第二个参数是移动到的目标节点对象。

onupdend在后端处理完节点移动之后执行的js方法，该方法需要返回一个布尔值，true表示继续执行后面的动作，false表示放弃执行后面的动作，该方法传入三个参数，一个是源节点（被移动节点）对象，第二个参数是移动到的目标节点对象，第三个参数为后端返回的json对象。

```
function onupdsubmit (a,b){
    //alert(a.title);
```


网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com

```
    return true;
}
function onupdend (a,b,c){
    alert(b.title);
    return true;
}
```

4.2 折叠菜单

折叠菜单由两个标签组成。

➢ 属性说明(Accordion)

属性	是否必须	说明
id	是	
width	否	菜单宽
height	否	菜单宽高
open	否	初始展开的菜单号，起始为0，-1表示展开所有，要展开多个菜单，用逗号分隔，如0,1,2
type	否	展开类型，可选{0|1}
align	否	文字对齐方式可选参数:bottom 底端，left 左边,right 右边
styleid	否	样式的id号，对应于配置文件的accord,设为-1时表示使用外部样式

➢ 属性说明(AccordionButton)

属性	是否必须	说明
name	是	Button条的名字
aligning	否	文字对齐方式可选参数:bottom 底端，left 左边,right 右边
onclick	否	单击button条时调用的方法
button	否	按钮对象
img	否	Button条的图标

按钮对象为一个List<Map<String,String>>对象。期中map的值如下。

```
Map sn = new HashMap();
sn.put("name","aaa");
sn.put("image","aa");
sn.put("url","aa");
sn.put("target","aaaa");
```


网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com

➢ 使用实例

```
<%
List<Map<String,String>> button=new ArrayList<Map<String,String>>();
for(int i=0;i<6;i++){
Map sn = new HashMap();
sn.put("name","aaa");
sn.put("image","aa");
sn.put("url","aa");
sn.put("target","aaaa");
button.add(sn);
}
request.setAttribute("b",button);
List<Map<String,String>> button1=new ArrayList<Map<String,String>>();
for(int i=0;i<6;i++){
Map sn = new HashMap();
sn.put("name","aaa");
sn.put("image","aa");
sn.put("url","aa");
sn.put("target","aaaa");
button1.add(sn);
}
request.setAttribute("b1",button);
 List<Map<String,String>> button2=new ArrayList<Map<String,String>>();
for(int i=0;i<6;i++){
Map sn = new HashMap();
sn.put("name","aaa");
sn.put("image","aa");
sn.put("url","aa");
sn.put("target","aaaa");
button2.add(sn);
}
request.setAttribute("b2",button);
%>
<n:Accordion id="dddf" type="0" open="1">
<n:AccordionButton name="df" button="${b}"> </n:AccordionButton>
<n:AccordionButton name="ddff" button="${b1}"> </n:AccordionButton>
<n:AccordionButton name="dsdff" button="${b2}" onclick="sss('dsdff')" >
</n:AccordionButton>
</n:Accordion>
<script type="text/javascript">
function sss(a){
```

```
alert(a);
}
</script>
```

4.3 树形折叠菜单

树形折叠菜单是在折叠菜单里面用树形菜单替换按钮菜单，按钮菜单和树形菜单可以共同存在，如下所示：

```
<n:Accordion id="faaa" type="1" open="0" styleid="1"  width="170" height="40">
<n:AccordionTree
name="aaa"
id="aaa"
sql="SELECT ID , PID, NAME, TITLE,LEVE FROM NK_SYS_TREEBOOK"/>
</n:Accordion>
```

➢ 属性说明(AccordionButton)

属性	是否必须	说明
name	是	Button条的名字
onclick	否	单击button条时调用的方法
img	否	Button条的图标
其它属性	用法及其作用同树形菜单，详见树菜单小节	

4.4 选项卡菜单

选项卡菜单由两个标签组成。如下所示。

➢ 实例图

网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com

➢ 属性说明(Tab)

属性	是否必须	说明
tagId	是	
tabWidth	否	菜单宽
tabHeight	否	菜单宽高
defaultTab	否	初始展开的菜单名字
tabstyle	否	皮肤名字，可选｛dark_blue｜default｜dhx_black｜dhx_blue｜dhx_skyblue_d｜glassy_blue｜modern｜silver｜winbiscarf｜winscarf｝默认为modern
tabtype	否	菜单对齐方式可选参数:bottom 底端，left 左边,right 右边

➢ 属性说明(TabConten)

属性	是否必须	说明
type	否	选项卡类型｛html｜ iframes-on-demand｜ajax-html｝其中html表示直接在显示标签签里的内容。后两种均为指定一个url
name	否	菜单名字
id	否	菜单id
width	否	菜单宽

➢ 使用实例

```
<n:tab tabWidth="400" tagId="d1" >
<n:TabConten name="ds" type="html" id="dff">fdffd</n:TabConten>
```


网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com

```
<n:TabConten name="dsdf" type="iframes-on-demand"
id="dffff">http://www.baidu.com</n:TabConten>
<n:TabConten name="ajax-html" type="html"
id="dffqff">http://www.baidu.com</n:TabConten>
</n:tab>
```

4.5 鼠标右键菜单

鼠标右键菜单是在页面上生成一个鼠标右键单击菜单，同时屏蔽系统右键菜单，该菜单由两个标签组成。

➢ 属性说明(rightMenu)

属性	是否必须	说明
id	是	菜单的ID
width	否	菜单宽
style	否	菜单的style
classStyle	否	菜单的class属性
autohide	否	在鼠标离开菜单时是否自动隐藏

➢ 属性说明(MenuItem)

属性	是否必须	说明
style	否	菜单的style
classStyle	否	菜单的class属性
jsfunction	否	菜单js方法名称

➢ 使用实例

```
<n:rightMenu id="afff" width="200">
  <n:MenuItem  jsfunction="ac">菜单1</n:MenuItem>
  <n:MenuItem  jsfunction="ac">菜单1</n:MenuItem>
  <n:MenuItem  jsfunction="ac">菜单1</n:MenuItem>
</n:rightMenu>
<script type="text/javascript">
    function ac(){
        var x = $('afff').PointerLeft();//菜单的X坐标
        var y = $('afff').PointerTop();//菜单的Y坐标
        alert(x+'  '+y);
    }
```

```
</script>
```

右键菜单里面的内容也可以是任何HTML内容和noka tag其它标签内容，如下在点击鼠标右键时弹出一个输入表单，为了便于输入，设置autohide为false，鼠标在移动时不会自动隐藏输入表单:

```
<n:rightMenu id="afff" width="200" autohide="false">
 <n:form id="faaa">
  <n:frow>
  <n:fcell label="设备名称"><n:InputText name="aaa" id="ffddaa"></n:InputText>
  </n:frow>
  <n:frow>
  <n:fcell><input type="button"  value="保存" onclick="ac()"></n:fcell>
  <n:fcell><input type="button"  value="取消" onclick="$('afff').hide();"></n:fcell>
  </n:frow>
  </n:form>
</n:rightMenu>
<script type="text/javascript">
    function ac(){
        var x = $('afff').PointerLeft();//菜单的X坐标
        var y = $('afff').PointerTop();//菜单的Y坐标
        alert(x+'  '+y);
        $('afff').hide();//隐藏菜单
    }
</script>
```

第5章 EL 工具系列

EL 工具系列是建立在 JSP2.0 标准之上的一系列 EL 自定义方法。其用法为{n:方法名(参数 1,…, 参数 n)} 。

方法	参数	返回
sp	String, String	java.util.list
备注	将字符串以指定的字符进行分隔。形如${n:sp(a,',')}	
sv	String,String,Integer	String
备注	将字符串以指定的字符进行分隔。返回指定下标的字符	
it	String	Boolean
备注	判断字符是否为空或为空白字符。	
ag	String	java.util.Long

网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com

备注	根据出生日期，返回当前的年龄。	
fd	Date,String,Boolean	String
备注	按指定格式对于日期时间类型进行格式化。Boolean为true表示在Date为空时返回当前时间，为false返回空格	
ifs	String,String	String
备注	根据传入的选项和key返回key所对应的值，第一个参数为选项形如[key=2a,show=是],[key=34,show=滞]，第二个参数为key所对应的值，返回为show对应的值	
estr	String	String
备注	加密传入的字符串，后台对应的方法为org.nokatag.elfunction.ELFunction.encryptStr(String str)	
dstr	String	String
备注	解密字符串，后台对应的方法为org.nokatag.elfunction.ELFunction.decryptStr(String str)	
a	String, PageContext	String
备注	在JSP页面中调用如${n:a('/index.jsp?key=8998',pageContext)}，该方法返回encodeURL编码之后的地址，自动加入根路径，支持urlrewrite功能	
fd	Object,String	String
备注	在JSP页面中利用DecimalFormat格式化数字显示，形如${n:df(445.3,'0.00')}显示为445.30	

第6章　其它辅助工具

6.1 服务器方法调用

该标签通过 Ajax 远程调用服务器方法。

➢ **属性说明**

属性	是否必须	说明
refunction	是	回调方法,参数为HTML对像
webfunction	是	客户端调用方法
serverfunction	是	服务器端方法
id	是	根据该ID生成服务器调用路径

➢ **写法如下**

1、带参数的写法：

```
<n:SerFun refunction="agg" webfunction="fa(a,b,c)"
serverfunction="org.user.Ftest.rut(java.lang.String a,java.lang.String
b,java.lang.Integer c)"></n:SerFun>
```

2、不带参数的写法：

```
<n:SerFun refunction="agg" webfunction="fa"
serverfunction="org.user.Ftest.rut"></n:SerFun>
```

如上例所示，可以通过 JavaScript 调用 fa(‘aa,’,’ bbb’,56)来调用服务器端方法，该例为调用服务器端的 org.user.Ftest.rut 方法，传过去的参数为：aa,bbb,和 56，当服务器执行完成后，回调方法 agg。在 serverfunction 当中可使用的类型有：java.lang.String，java.lang.Integer，java.lang.Double，javax.servlet.http.HttpServletRequest。其中，需要注意的是 webfunction 当中的参数个数要与 serverfunction 一致，可以不同名。

服务器调用路径为 inputtextcheck[id].serfun，其中[id]为控件的 ID 值。

agg 方法如下：

```
<script type="text/javascript">
function agg(a){ //期中 a 为返回的对像
  alert(a.responseText);
}
</script>
```

服务器上的方法如下：

```
package org.user;
public class Ftest {
public static String rut(String a,String b,Integer c){
System.out.println("a:"+a);
System.out.println("b:"+b);
System.out.println("c:"+c);
return a+":"+b;
}
}
```

6.2 DES 加密工具

类：org.nokatag.password.DESPlus

```
String test = "admin";
//DESPlus des = new DESPlus();//默认密钥national
DESPlus des = new DESPlus("kkkkkkkkkkkkkkssssssssssssssssssd");//自定义密钥
System.out.println("加密前的字符："+test);
System.out.println("加密后的字符："+des.encrypt(test));
System.out.println("解密后的字符："+des.decrypt(des.encrypt(test)));
```

类：org.nokatag.password.HmacPass

```
byte[]
keys={11,-105,-119,50,4,-105,16,38,-14,-111,21,-95,70,-15,76,-74,67,-88,59,-71,55,-
125,104,42};
PassStr passStr = HmacPass.Pass(keys,"xxx");
System.out.println("密码:"+passStr.getSrcpass());
System.out.println("密码:"+passStr.getPass());
byte[] salt = passStr.getSalt();
String salts="";
for (int i=0; i<salt.length;i++){
salts+=","+salt[i];
}
System.out.println("绕码:"+salts);
byte[] oc={-5,-8,-17,-44,-92,-74,122,-13,-78,-103,-7,48};
boolean istrue = HmacPass.ePass(keys,oc,"e393b3516c6a35a5d578022a265386f8","xxx");
System.out.print(istrue);
try { System.out.println("===========:"+NetworkInfo.getMacAddress());
} catch (IOException e) {
e.printStackTrace();
}
```

6.3 AES 加密工具

AES 加密类位于 org.nokatag.password. AESPlus 中，提供加密和解密两种方法，采用 AES 标准加密算法，可对字符串（支持多种语言）进行加解密。实例如下：

```
String sr=”需要加密的文本”;//需要加密的文本
String key=”noka_tag_passkey”;//加密用的密钥，可随意自定义，请参照AES密钥标准
String esr=AESPlus. encrypt (sr, key);//加密文本
String dsr=AESPlus.decrypt(msr, key); //解密文本
```

6.4 Hibernate 辅助操作工具类

方法详解请查看 javadoc

HibernateUtil:Hibernate 连接操作类，主要作用为取得 Connection 对像

DataUtil:基于 Hibernate 的数据操作类，针对数据对像执行新增，修改，删除，批量删除，分页等。包含以下方法

```
public static void beginTransaction()开始事务
public static void commitTransaction()提交事务
public static Serializable insert(Object object)插入数据
public static Serializable insertList(List<Object> objectlist)批量插入数据
public static boolean delete(Object object)删除数据
public static boolean deleteList(List<Object> objectlist)批量删除数据
public static boolean update(Object object)修改数据
public static boolean updateList(List<Object> objectlist)批量修改数据
public static List objectList(String hsql)查询数据
public static List objectList(String hsql,int start,int max)查询数据
public static Query query(String hsql)
public static Object object(String hsql)
public static DataPage dataPage(String hsql,Integer newpage,Integer pagesize)
public static DataPageList dataPagelist(String hsql,Integer cells,Integer
newpage,Integer pagesize)
public static List objectList(String hsql,List<DataWhere> where)
```

6.5 JavaScript 压缩工具标签

该标签压缩页面的上 javascript 内容，并重新输出压缩之后的内容在页面的原位置上，可以节约页面空间，减少页面容量的同时可以起到混淆 javascript 代码的作用，使用方法如下（标签位于<script type="text/javascript">内部，该标签自带缓存功能，设置 ID 属性以后开启缓存，在没有 ID 属性时缓存关闭，

该 ID 属性为整个页面唯一）：

```
<n:jszip id="javascript_id">
//----------修改提交表单--------------------
function updates(){
        nk_submitform();
}
    function onsuccess(a){
        switch(a.responseText.trim()){
        case '1':
            $('msg_util').innerHTML='修改成功';
            break;
        case '2':
            $('msg_util').innerHTML='修改失败';
            break;
        }
    }
    function onfailure(a){
    }
//-----------页面初始化------------
window.onload = function(){
    $("usname_id").focus();
};
</n:jszip>
```

6.6 CSS 压缩工具标签

该标签压缩页面的上 CSS 内容，并重新输出压缩之后的内容在页面的原位置上，可以节约页面空间，减少页面容量的同时可以起到混淆 CSS 代码的作用，使用方法如下(标签位于<style type="text/css">内部，该标签自带缓存功能，设置 ID 属性以后开启缓存，在没有 ID 属性时缓存关闭，该 ID 属性为整个页面唯一)：

```
  <style type="text/css">
<n:csszip id="css_id">
* {margin:0; padding:0; font:12px Verdana,Arial}
.${accordion.style.css_id} {width:${accordion.style.css_width}px;
list-style:none; color:#033; margin-bottom:15px}
.${accordion.style.css_id} h3 {width:${accordion.style.css_hwidth}px;
```


网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com

```
text-align:${accordion.style.css_align};border:1px solid #9ac1c9;
padding:6px 6px 8px; font-weight:bold; margin-top:5px; cursor:pointer;
background:url(${rootpath}/nokatag/accordion/images/header.gif)}
.${accordion.style.css_id} h3:hover
{background:url(${rootpath}/nokatag/accordion/images/header_over.gif)
}
.${accordion.style.css_id} .acc-section {overflow:hidden;
background:#fff;margin:0px auto; text-align:center;}
.${accordion.style.css_id} .acc-content
{width:${accordion.style.css_contentwidth}px;height:${accordion.style
.css_height}px;overflow:auto;text-align:center; margin:0px auto;
padding:15px; border:1px solid #9ac1c9; border-top:none;
background:#fff}
.${accordion.style.css_id} .acc-selected
{background:url(${rootpath}/nokatag/accordion/images/header_over.gif)
}
.noka_tag_button{background-color: transparent;cursor: pointer;border:
0px;};
</style>
</n:csszip>
```

6.7 Js、Css 压缩工具类

该压缩工具是针对 Js、css 的压缩，其引擎采用的是 YUICompressor 2.4.8，该引擎来自于 yahoo。

注：一般建议不要将中文字符放入需要压缩处理的文件内部，可能会引起中文件编码问题。

该工具类位于：org.noka.compressor 下，使用实例如下，更详细的说明请参考 YUICompressor 说明手册。

```
InputStream  fi= null;//实体化自己的 InputStream
ByteArrayOutputStream out=new ByteArrayOutputStream();
Reader in= new InputStreamReader(fi);
JavaScriptCompressor compressor = new JavaScriptCompressor(in,null);
compressor.compress(out,-1, true, false,false, false);
```

6.8 ServletNokaContext

ServletNokaContext 对象是线程安全的 http 对象集合，该对象可以在任何 noka 拦截器以后调用的方法内直接使用，该类引用路径为 org.nokatag.system.ServletNokaContext 具体方法有：

ServletNokaContext.getPageContext();//获取 PageContext 对象

ServletNokaContext.getRequest();//获取 HttpRequest 对象

ServletNokaContext.getResponse();//获取 HttpResponse 对象

ServletNokaContext.getServletContext();//获取 ServletContext 对象

ServletNokaContext.getConnection();//获取数据库连接对象

ServletNokaContext.setConnection();//设置 connection

ServletNokaContext.getHttpSession();//获取 http 的 session 对象

ServletNokaContext.getHibernateSession();//获取 hibernate 的 session。

ServletNokaContext. setHibernateSession();//设置 hibernate 的 session。

ServletNokaContext.getIbatisSession();//获取 Ibatis 的 SqlSession

注：通过 ServletNokaContext.getConnection()获取的连接对象以及 Session 对象不需要关闭，将由 SQL 拦截器统一关闭。

6.9 Ibatis 辅助操作类

Ibatis 辅助操作类 org.nokatag.system.IbatisUtil，要使用该类 ibatis 的配置文件名称约定为 ibatis-config.xml，存放于 classes 目录下。

currentSession();//获取当前线程下的 SqlSession 对象

currentNewSession();//获取一新的 SqlSession 对象

closeSession();//关闭当前线程下的 SqlSession 对象

closeSession(SqlSession session);//关闭传入的 SqlSession 对象

6.10 迷你消息提示框

该控件在页面生成一个消息提示框，可以绑定指定的网页对象使共具有消息提示的各种功能，也可以通过其 ID 独立调用。

如下实例：

```
<n:infobox id="fss" event="aa,bb,cc"></n:infobox>
```

以上实例通过 event 绑定了三个网页对象，其中 aa,bb,cc 表示三个网页对象的 id 属性，因此在三个网页对象存在的情况下，可以通过$(‘aa’)、(‘bb’)、(‘cc’)或(‘fss’)进行调用。

其 javaScript 方法如下：

showInfo(msg,autohide),显示提示信息，msg 为显示的内容，autohide 表示在显示 autohide 秒后，自动关闭显示。

showErro(msg,autohide),显示错误提示信息，用法同 showInfo()。

showLoad(autohide)，显示等待提示，autohide 表示在显示 autohie 秒后，自动关闭显示。

closeInfo()，关闭显示。

调用实例：

```
$('aa').showInfo('提示')
```


网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com

第7章 事务调度务服

事务调度可以按需要定制定时调用某个类里的某个方法。该服务基于 Timer 进行扩展，具有灵活的调用机制。

7.1 添加事务

```
NTimer nTimer = NTimer.IniterNTimerItem();//初始化
SimpleDateFormat de=new SimpleDateFormat("yyyy-MM-dd kk:mm:ss"); //格式化时间
List<Object> ClassPrave = new ArrayList<Object>();//传入调用方法的参数列表
ClassPrave.add("AD_1");//传入字符串参数
try{
NTimerItem nt =new NTimerItem(
"AD_1",//事务的 ID 号，全系统唯一，以后可以用该 ID 进行更新，删除等操作
NTimer.TIMER_INTERVAL_SECOND,//事务调度类型，以下下详细介绍
"2",//事务调度的参数，这里是指间隔 2 秒，重复调用
"TA.print",//调用的方法，ta 是指类，print 是指类里的方法
"3",//延后调用，以秒为单位,这里指 3 秒后开始调用
de.parse("2001-03-12 3:00:00"),//事务开始调用的日期
ClassPrave//参数列表
);
nTimer.addTimer(nt);
}catch(Exception se){
}
```

7.2 事务类型说明

```
TIMER_INTERVAL_SECOND;//每几秒钟执行一次,对应的事务调度的参数为秒数
TIMER_INTERVAL_MINUTE;//每几分钟执行一次，对应的事务调度的参数为分种数
TIMER_INTERVAL_HOUR;//每几小时执行一次，对应的事务调度的参数为小时数
TIMER_INTERVAL_DAY//每几天执行一次，对应的事务调度的参数为天数
TIMER_INTERVAL_WEEK//每几周执行一次，对应的事务调度的参数为周数
TIMER_DAY_OF_TIME//每天的几点几分几秒执行,对应的参数格式为：08:23:52
TIMER_WEEK_OF_TIME//每周的星期几，几点几分几秒执行,星期几，几点几分几秒(3, 08:23:52)
TIMER_MONTH_OF_TIME//每月的几号,几点几分几秒执行,几号几点几分内秒(3, 08:23:52)
TIMER_YEAR_OF_TIME//每年的几月几号几点几分几秒执行几月几点几分几秒(03-3, 08:23:52)
TIMER_MONTH_MAXDAY_OF_TIME//每个月的最后一天 08:23:52
```


网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com

以下是每种调度的详细添加实例

```
//----------每月的最后一天 09:23:22 执行--------------------
NTimerItem nt =new NTimerItem("AD_1",
NTimer.TIMER_MONTH_MAXDAY_OF_TIME,"09:23:22","TA.print","3");
nTimer.addTimer(nt);
//----------每年的 5 月 6 号 09:23:22 执行(月从 0 开始)--------------------
 NTimerItem nt2 =new NTimerItem("AD_2",
NTimer.TIMER_YEAR_OF_TIME,"05-06,09:23:22","TA.print","3");
nTimer.addTimer(nt2);
//----------每月 6 号 09:23:22 执行--------------------
NTimerItem nt3 =new NTimerItem("AD_3",
NTimer.TIMER_MONTH_OF_TIME,"06,09:23:22","TA.print","3");
nTimer.addTimer(nt3);
//----------每周星期三 09:23:22 执行(周从 0 开始)--------------------
NTimerItem nt4 =new NTimerItem("AD_4",
NTimer.TIMER_WEEK_OF_TIME,"2,09:23:22","TA.print","3");
nTimer.addTimer(nt4);
//----------每天 09:23:22 执行--------------------
NTimerItem nt5 =new NTimerItem("AD_5",
NTimer.TIMER_DAY_OF_TIME,"09:23:22","TA.print","3");
nTimer.addTimer(nt5);
//----------每 2 周执行一次(时间以当前时间为准)--------------------
NTimerItem nt6 =new NTimerItem("AD_6",
NTimer.TIMER_INTERVAL_WEEK,"2","TA.print","3");
nTimer.addTimer(nt6);
//----------每 2 天执行一次(时间以当前时间为准)--------------------
NTimerItem nt7 =new NTimerItem("AD_7", NTimer.TIMER_INTERVAL_DAY,"2","TA.print","3");
nTimer.addTimer(nt7);
//----------每 2 小时执行一次(时间以当前时间为准)--------------------
NTimerItem nt8 =new NTimerItem("AD_8",
NTimer.TIMER_INTERVAL_HOUR,"2","TA.print","3");
nTimer.addTimer(nt8);
//----------每 2 分钟执行一次(时间以当前时间为准)--------------------
NTimerItem nt9 =new NTimerItem("AD_9",
NTimer.TIMER_INTERVAL_MINUTE,"2","TA.print","3");
nTimer.addTimer(nt9);
//----------每 2 秒钟执行一次(时间以当前时间为准)--------------------
NTimerItem nt10 =new NTimerItem("AD_10",
NTimer.TIMER_INTERVAL_SECOND,"2","TA.print","3");
nTimer.addTimer(nt10);
```


网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com

7.3 事务更新

```
//----------批量删除定时器----------------------------------------------------
public void dellTimer(List<String> TimerID);
//------------单个删除----------------------------------------------------
public void dellTimer(String TimerID);
//-----------批量修改定时器----------------------------------------------------
public void updateTimer(List<NTimerItem> nTimerItems);
//--------------单个修改----------------------------------------------------
public void updateTimer(NTimerItem NTimerItem);
//--------------获取事务对像----------------------------------------------------
public NTimerItem getNTimerItem(String NTimerID);
```

7.4 事务配置

Noka 支持以配置文件方式注入事务，需要增加一个事务时，在 noka-config.xml 中直接配置事务即可，配置如下：

```
<timers>
  <timer
     ClassUrl="org.test.TimerTest.test"
     TimerPrave="3"
     DoType="TIMER_INTERVAL_SECOND"
     TimerID="box_air_timer"
     Delayed="3"
     start="2016-06-26 16:48:04"
     >
  </timer>
</timers>
```

ClassUrl:定时任务的实现类，这里配置的是全路径，org.test 指类的包名，TimerTest 为类名，test 为方法名，该实例见跟目录 expjava 下的 TimerTest

TimerPrave: 事务调度的参数，这里是指间隔 3 秒，重复调用

DoType: 事务调度类型

TimerID: 事务的 ID 号，全系统唯一

Delayed: 延后调用，以秒为单位,这里指 3 秒后开始调用,该参数非必须

start: 事务开始调用的时间，格式: yyyy-MM-dd HH:mm:ss,该参数非必须

第8章 TCP/IP 服务

TCP/IP 服务是基于 java socket 的 socket 服务封装，通过 socket 开启一个支持多客户端的多线程的 TCP/IP 服务端程序，当有新客户端连接、数据到达、断开等，会主动通知处理接口。

8.1 服务主类参数说明

org.noka.socketserver.NSocketServer 服务主类，构建方法如下：

NSocketServer(ClientInterface NClient,Integer port,Integer cmax,Integer disTimeOut)

ClientInterface NClient:数据处理接口类，需要继承 org.noka.socketserver.ClientInterface

Integer port:开启服务的端口

Integer cmax:最多支持的客户端数量

Integer disTimeOut:多长时间没有接收到客户端消息，调用超时处理方法，单位为分钟，默认为 5 分钟

StartServer()启服务

stopServer()停止服务

8.2 数据处理接口方法

Boolean onConnection(Socket socket) 有客户端连接时调用该方法，

Socket 为客户端连接对象，返回 false 表示断开连接，返回 true 表示保持连接

Boolean onReadData(Socket socket) 有数据到达时调用的方法，Socket 为客户端连接对象，返回 false 表示断开连接，返回 true 表示保持连接

void onUnConnection(Socket socket) 连接断开时调用该方法

Boolean onDisTimeout(Socket socket,Long outMine)根据构建方法的超时设置，超过指定时间没有接收客户端的数据，调用该方法，outMine 为时长，表示多长时间没有接收到客户端数据，返回 true 表示继续等待，返回 false 表示断开连接。

8.3 使用实例

```
//----------开启服务------------------------------------------------------------
NClient nc = new NClient();
NSocketServer n=new NSocketServer(nc,8090,1000);
n.StartServer();
```

Nc 类源码如下：

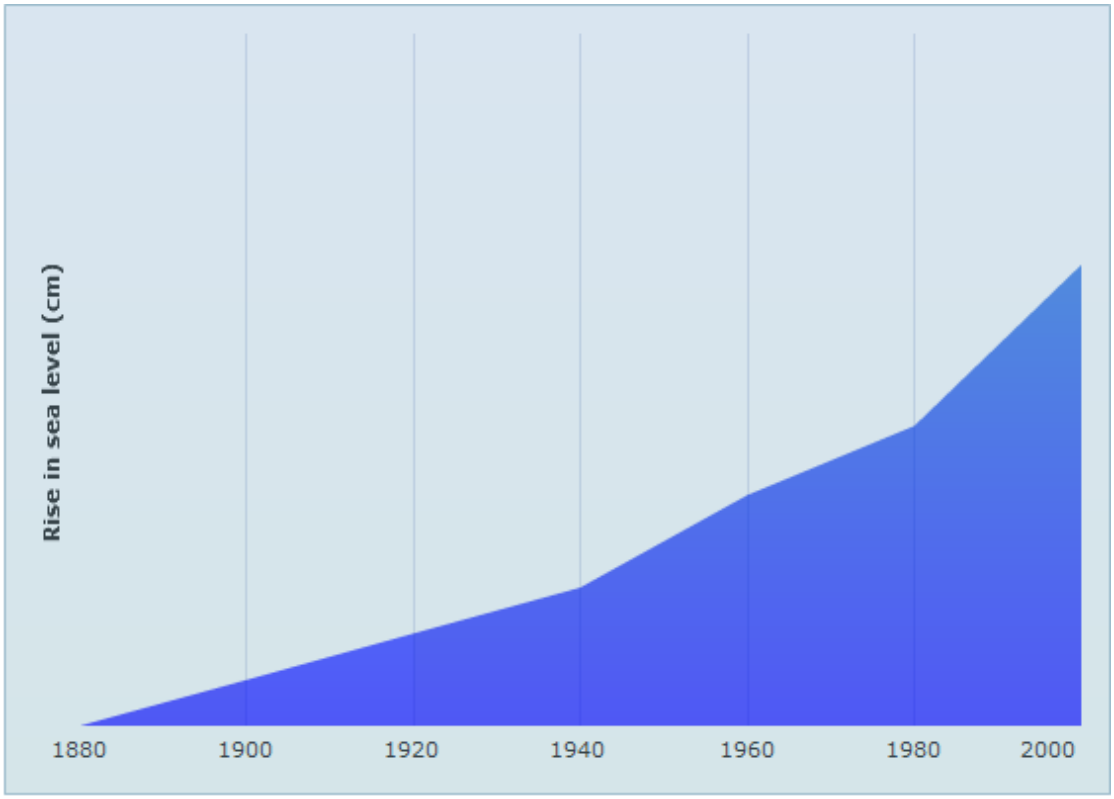
```
public class NClient implements ClientInterface {
    //-------------当客户端链接时----------------------
    public Boolean onConnection(Socket socket) {
        return true;//保持连接
    }
    //-------------有数据到达时------------------------
    public Boolean onReadData(Socket socket) {
        try{
        DataInputStream in = new DataInputStream(socket.getInputStream());//输入流
        DataOutput out = new DataOutputStream(socket.getOutputStream());//输出流
        }catch(Exception se){
            return false;
        }
        return true;
    }
    //--------------断开时-------------------------------
    public void onUnConnection(Socket socket) {

    }
```

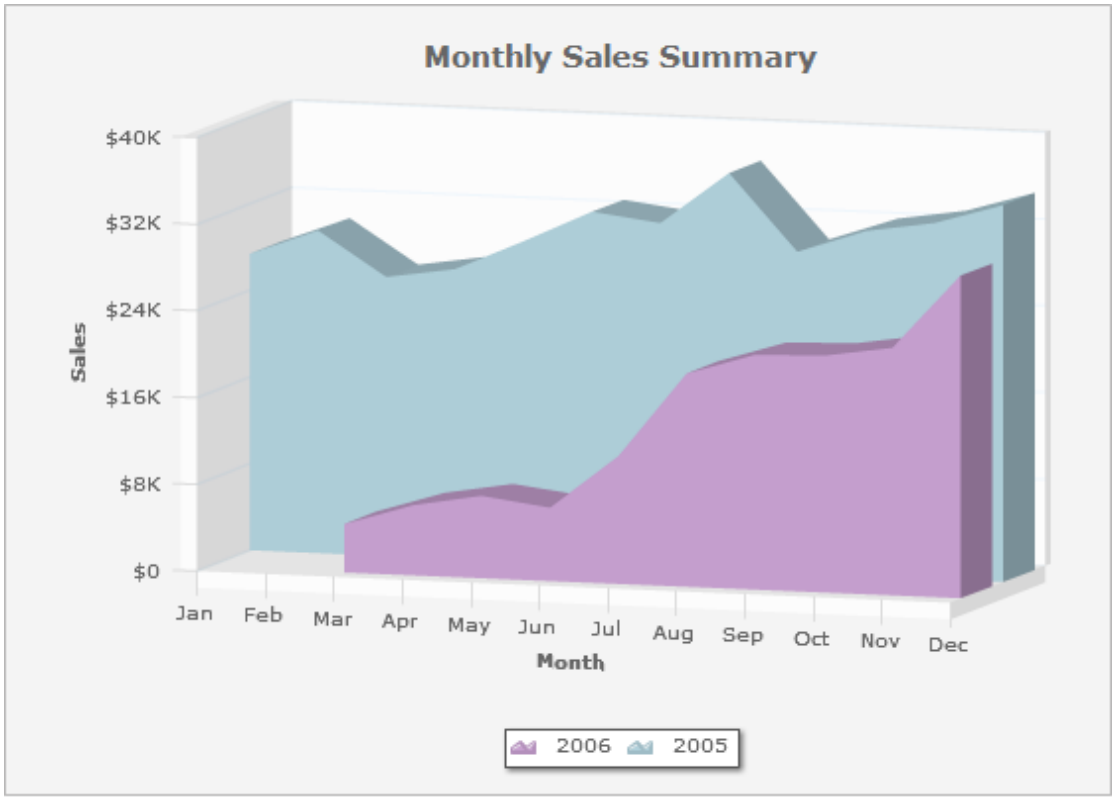
```
    //---------------超时-------------------------------
    public Boolean onDisTimeout(Socket socket, Long outMine) {
        return true;
    }
}
```

第9章　附录

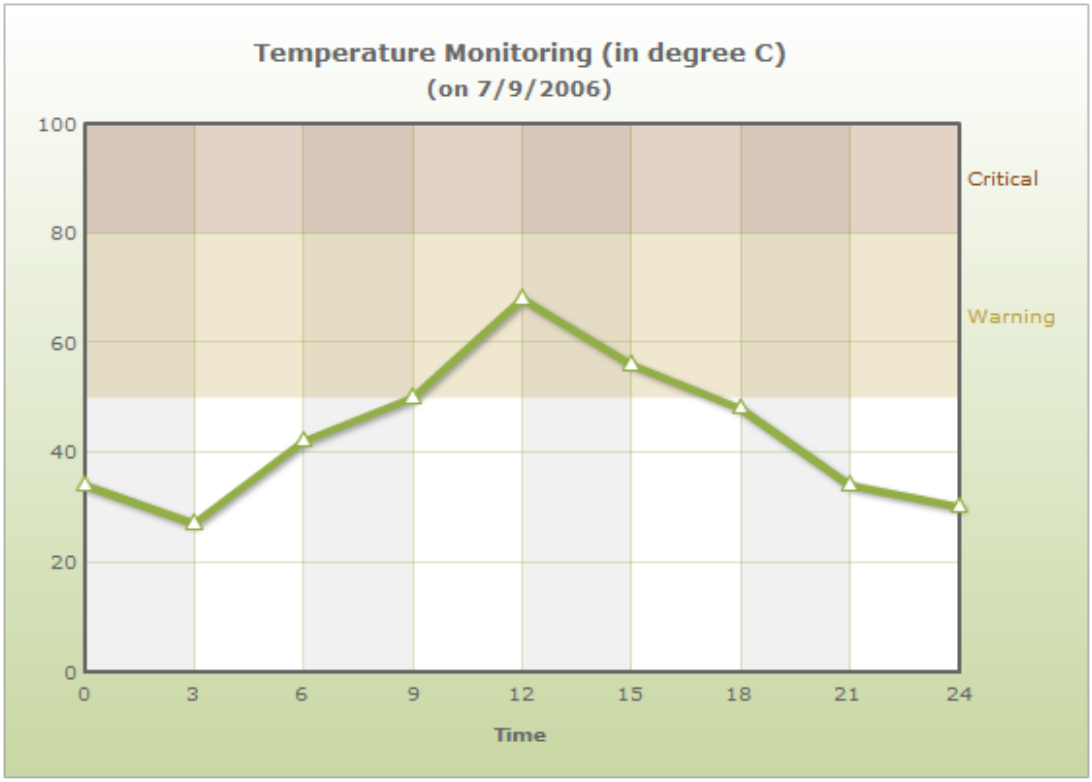
9.1 附录一（统计图表）



(类型：Area2D)

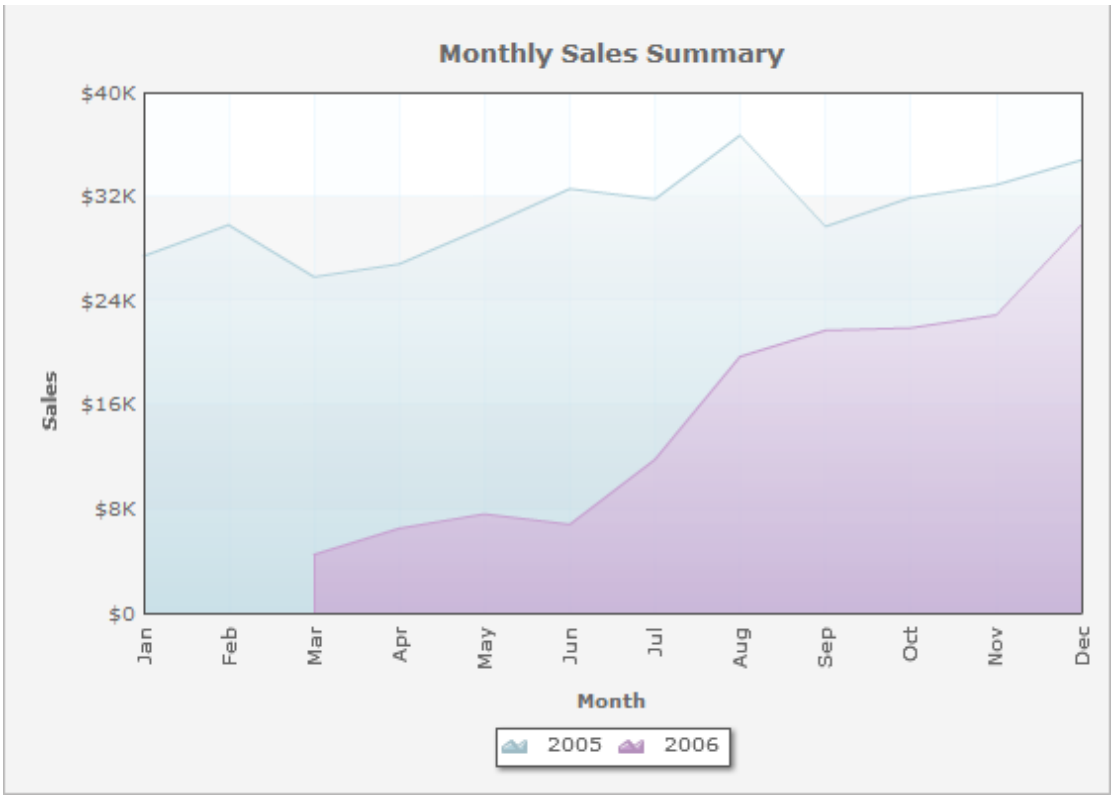


(类型：MSCombi3D)

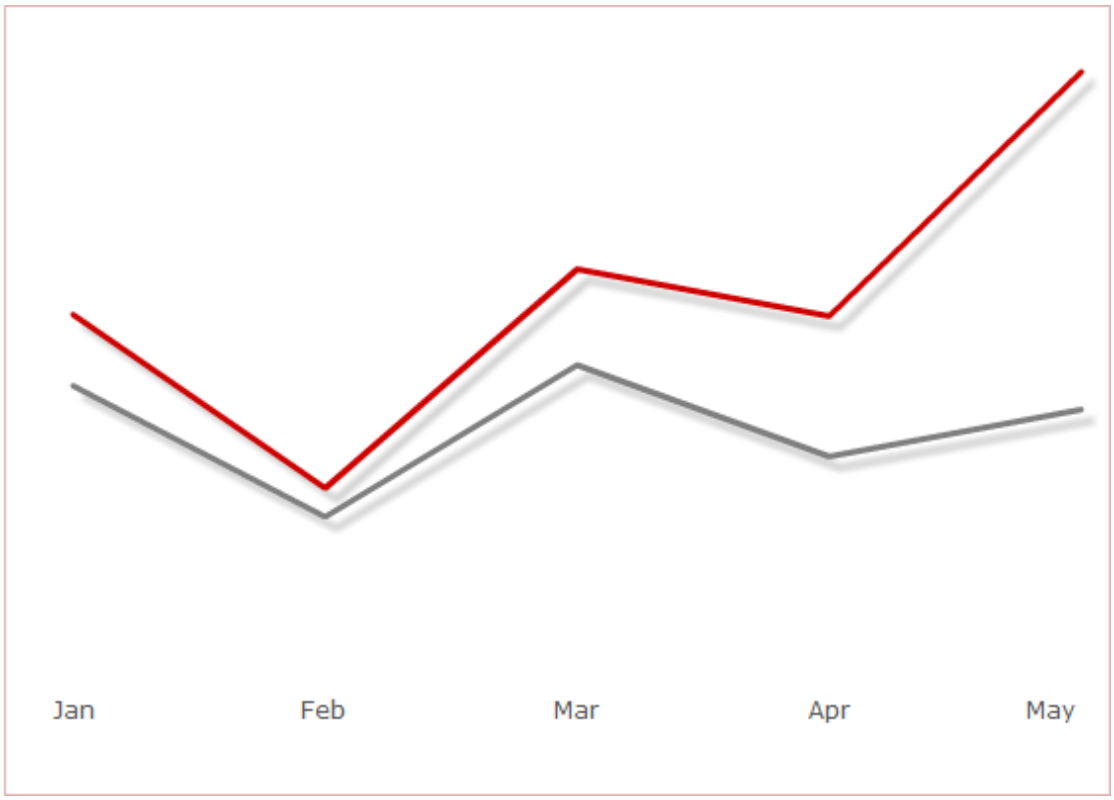


网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com

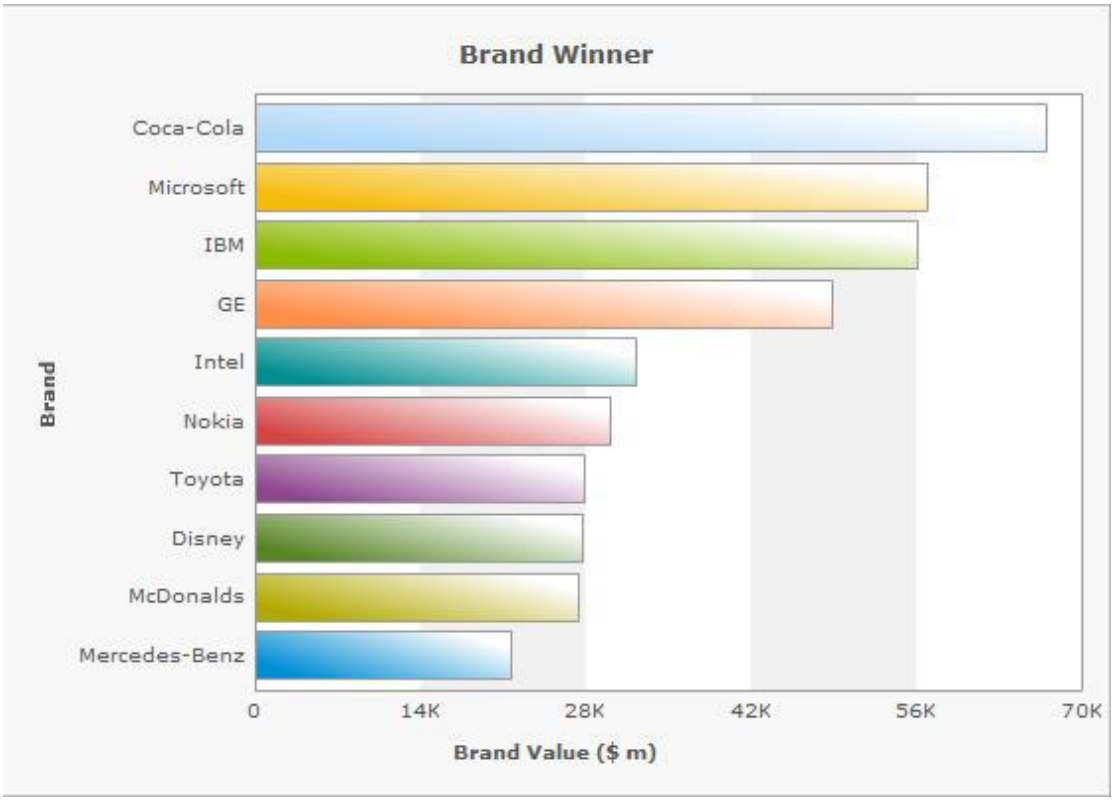
(类型：Line)



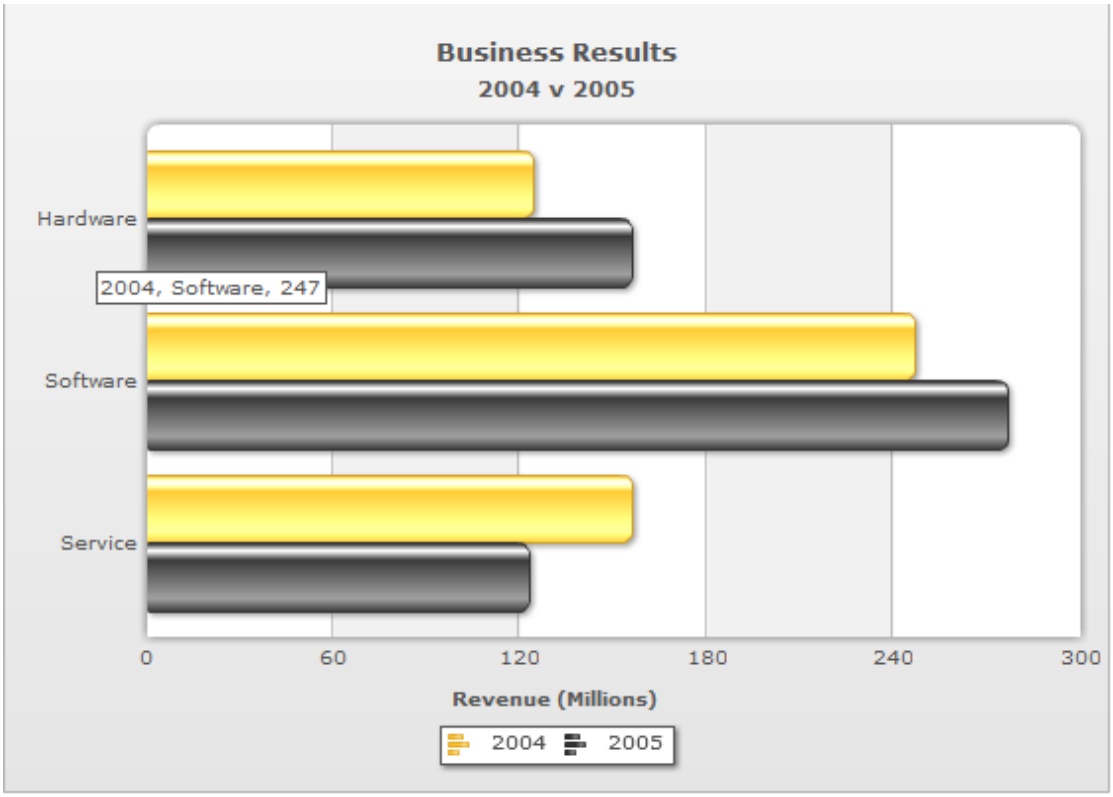
(类型：MSArea)



(类型：MSLine)

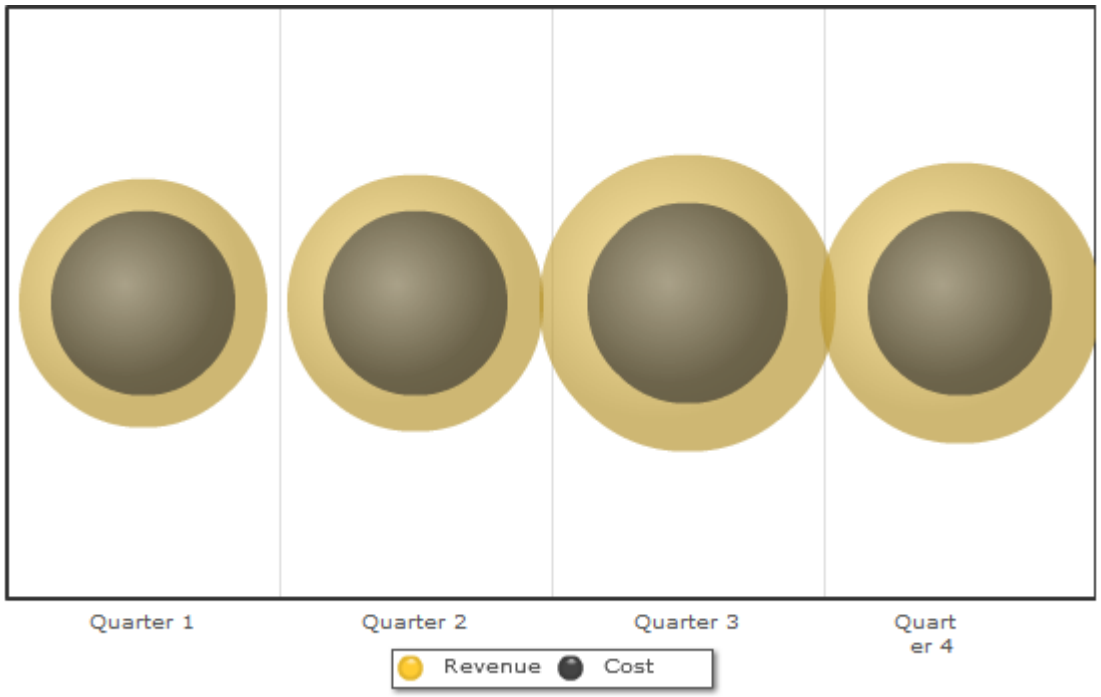


(类型：Bar2D)

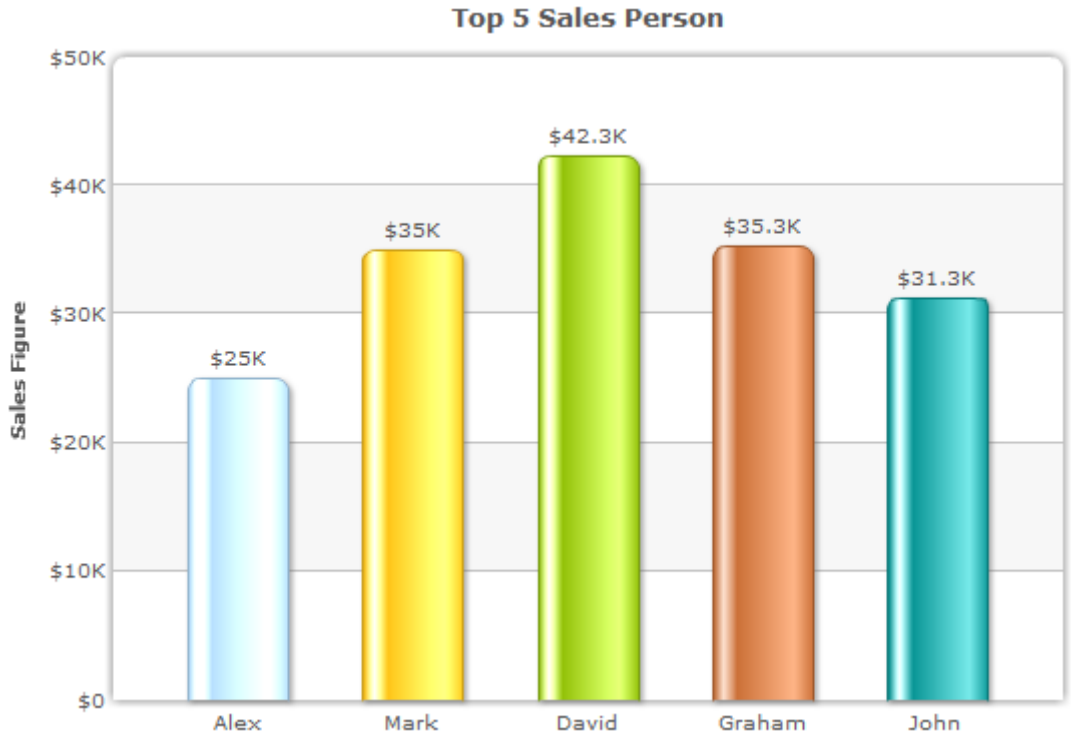


网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com

(类型：MSBar2D)

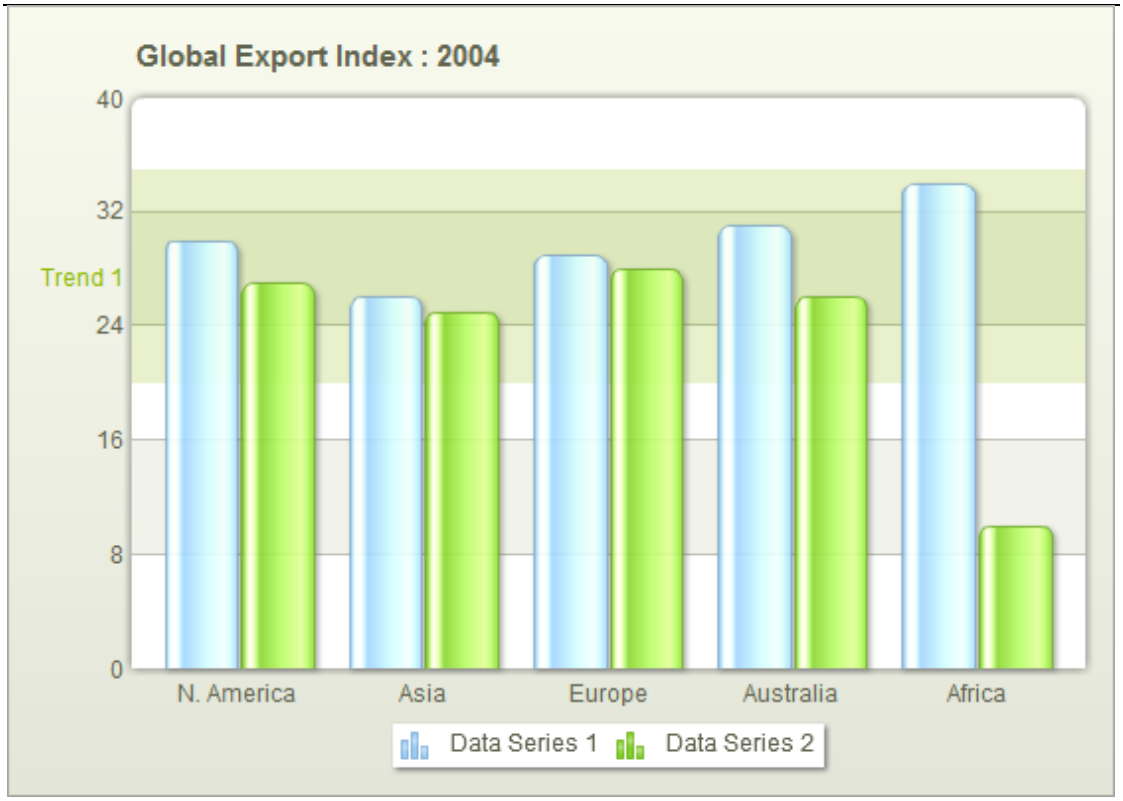


(类型：Bubble)

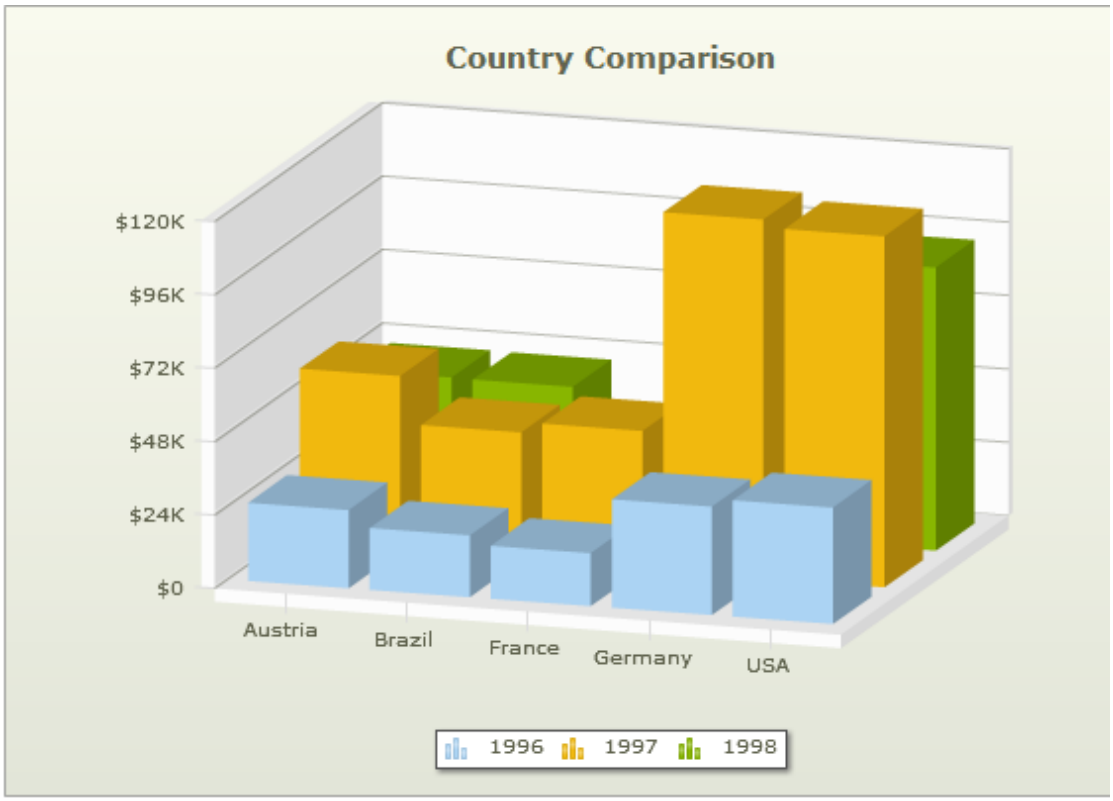


(类型：Column2D)

网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com

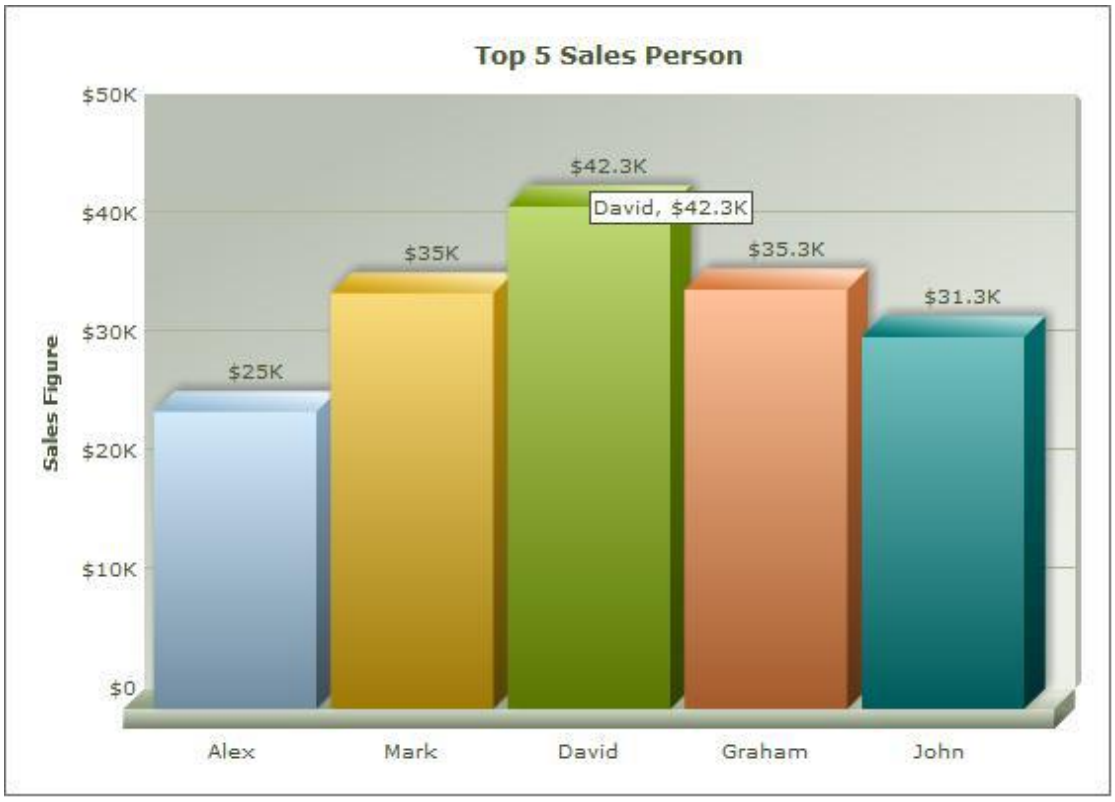


(类型：MSColumn2D)

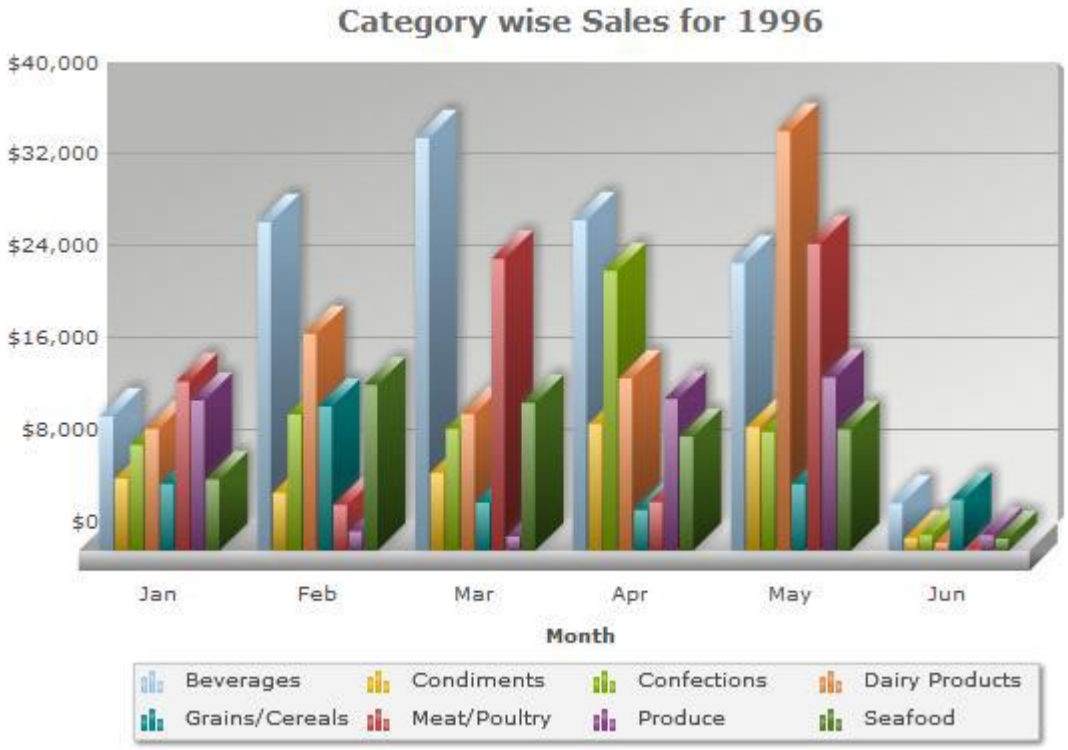


(类型：MSCombi3D)

网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com



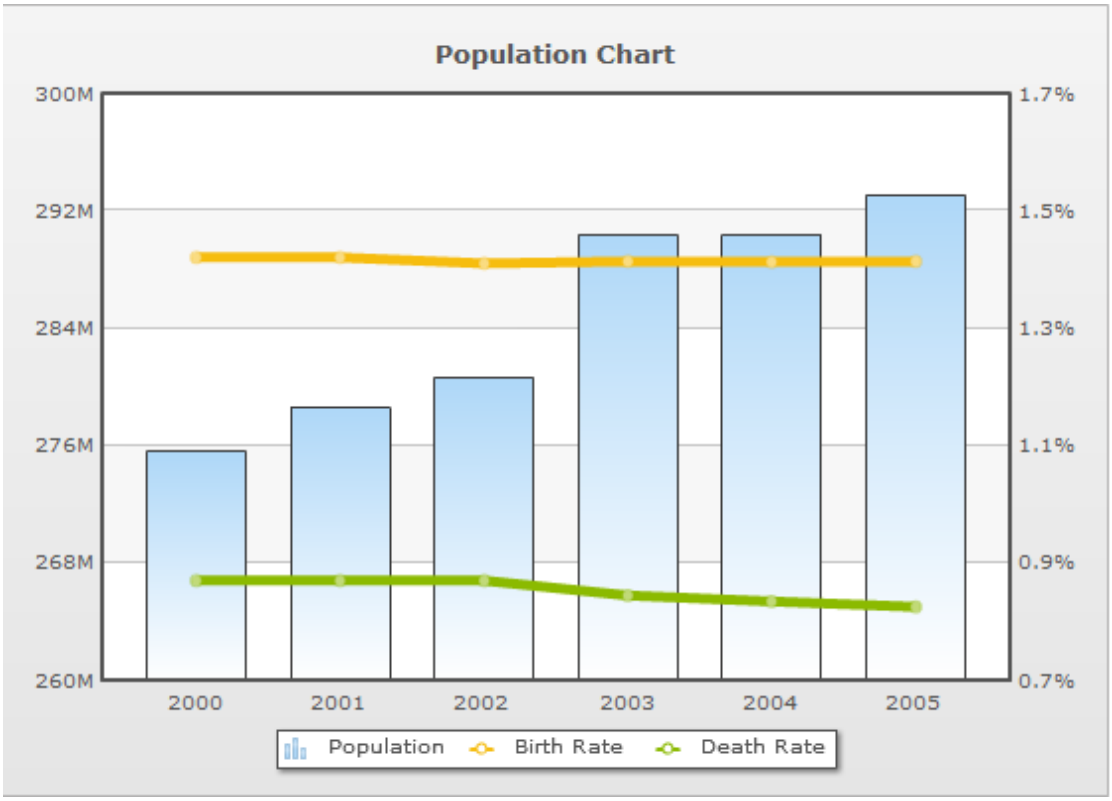
(类型：Column3D)



(类型：MSColumn3D)

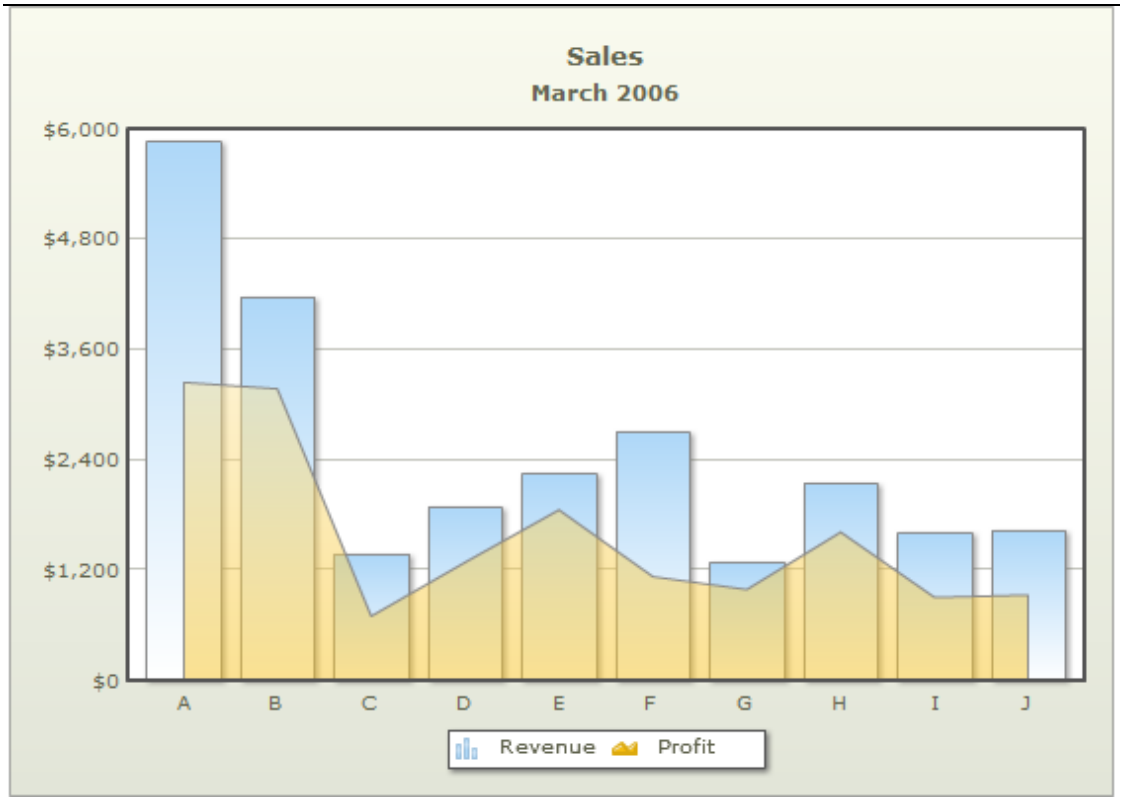
网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com

(类型：MSColumn3DLineDY)

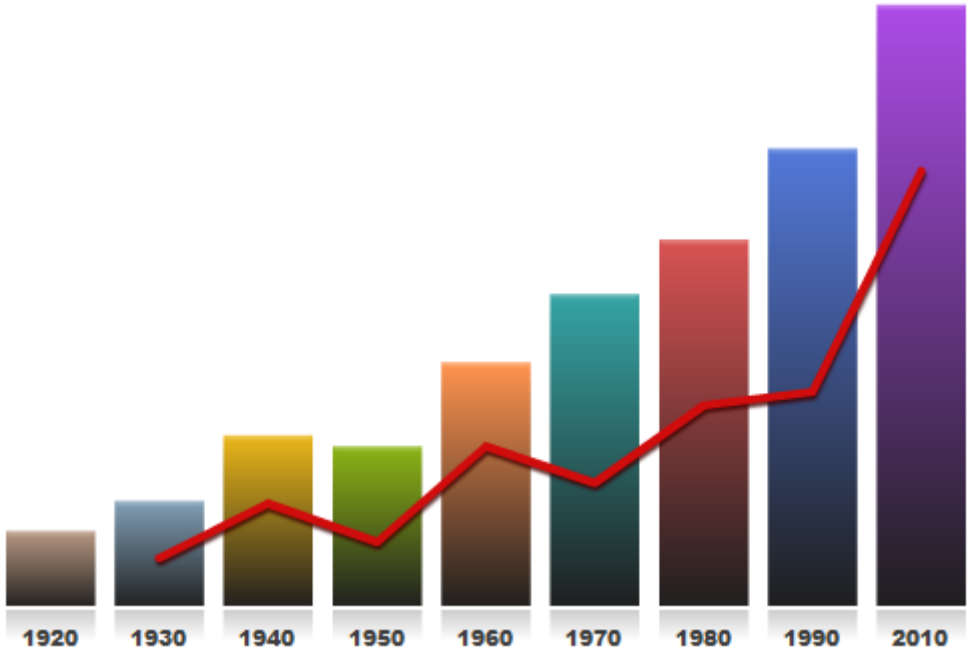


(类型：MSCombiDY2D)

网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com

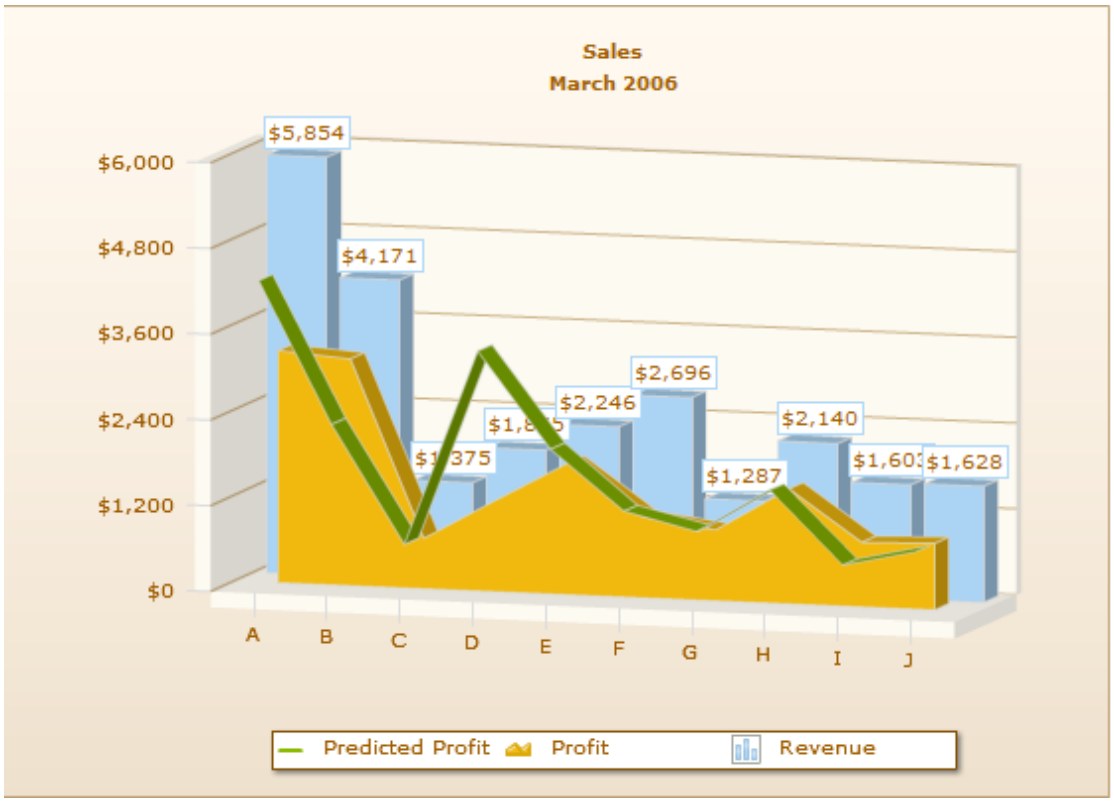


(类型：MSCombi2D)



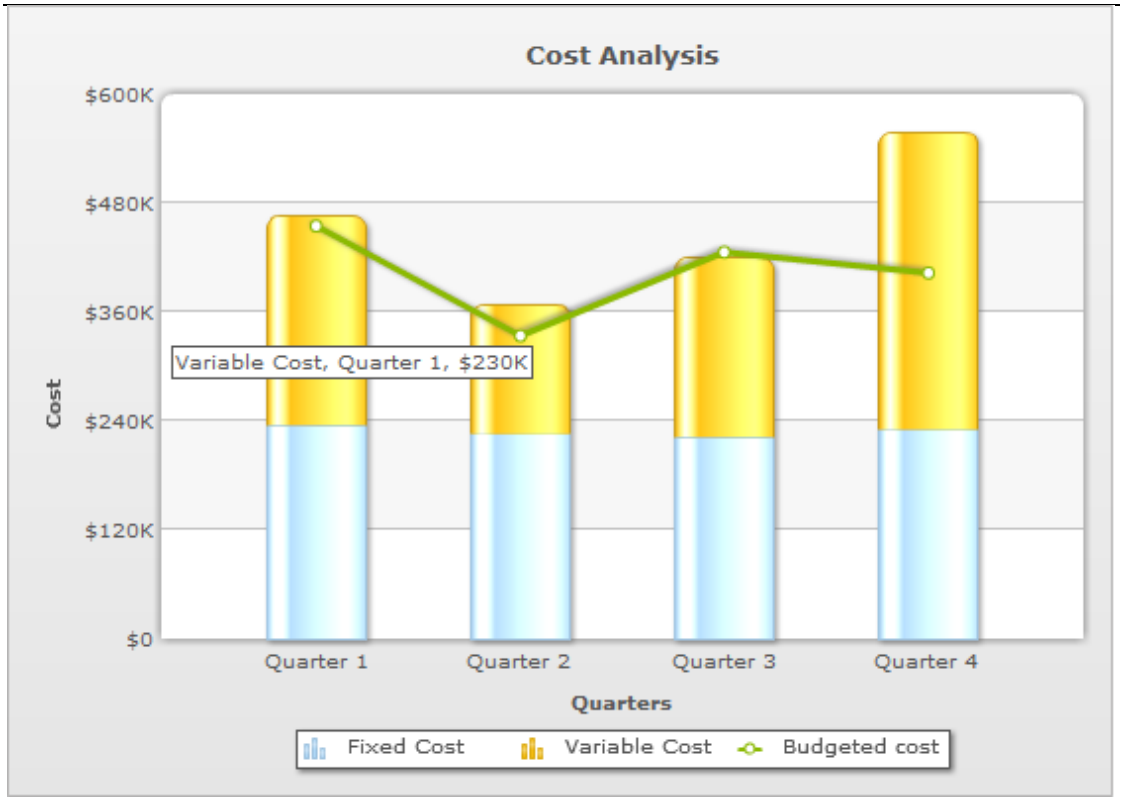
(类型：MSCombi2D)

网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com

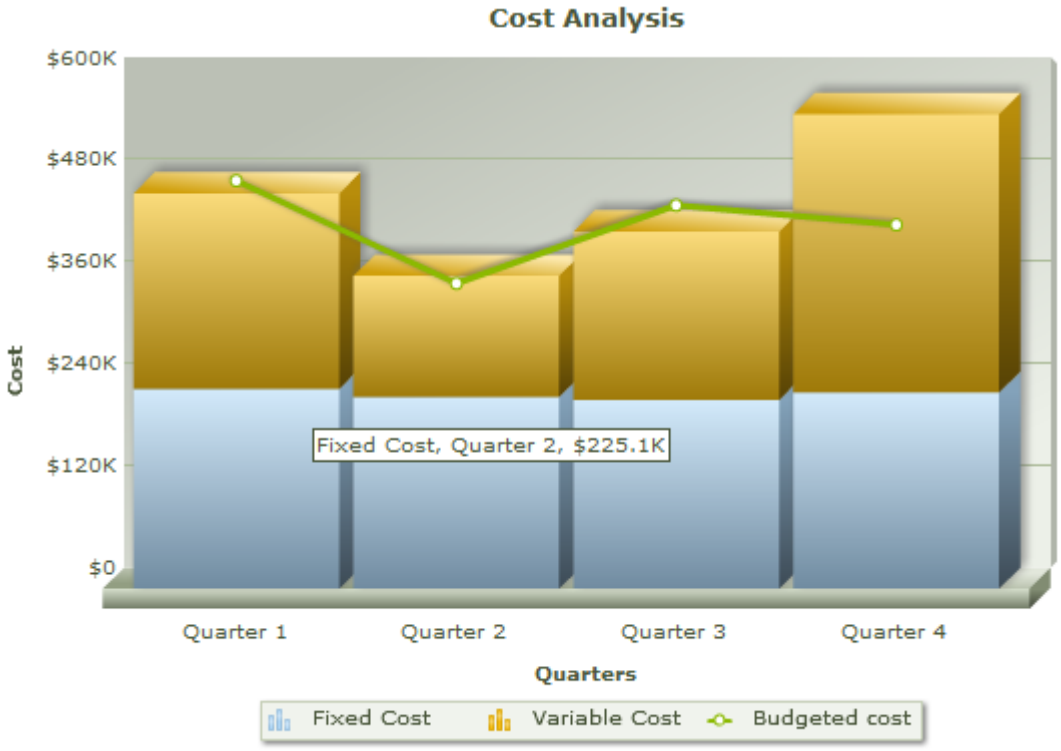


(类型：MSCombi3D)

(类型：MSColumnLine3D)

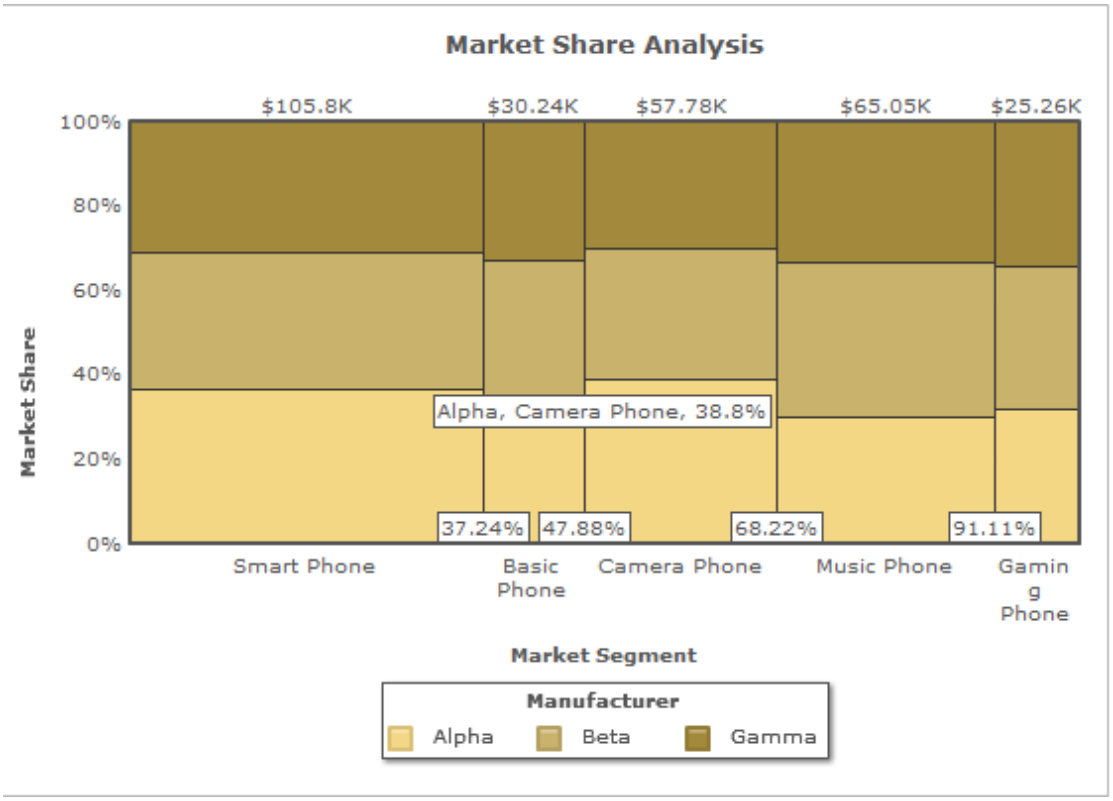


(类型：StackedColumn2DLine)

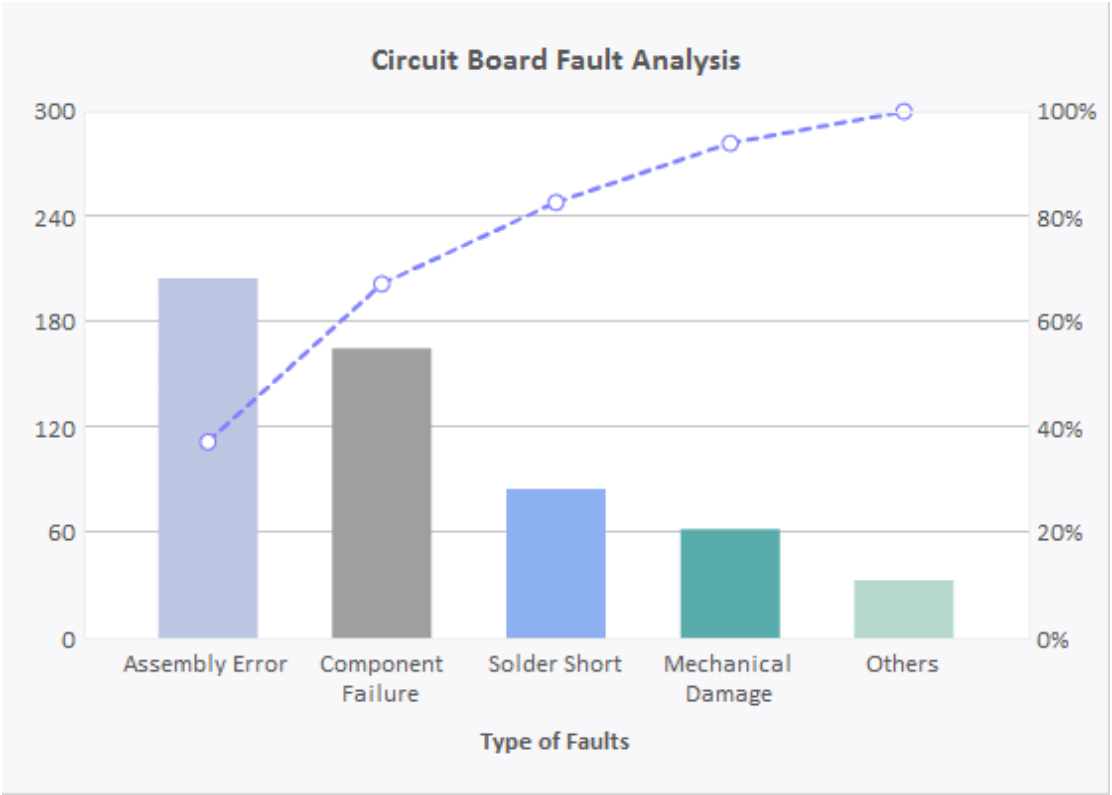


(类型：StackedColumn3DLine)

网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com

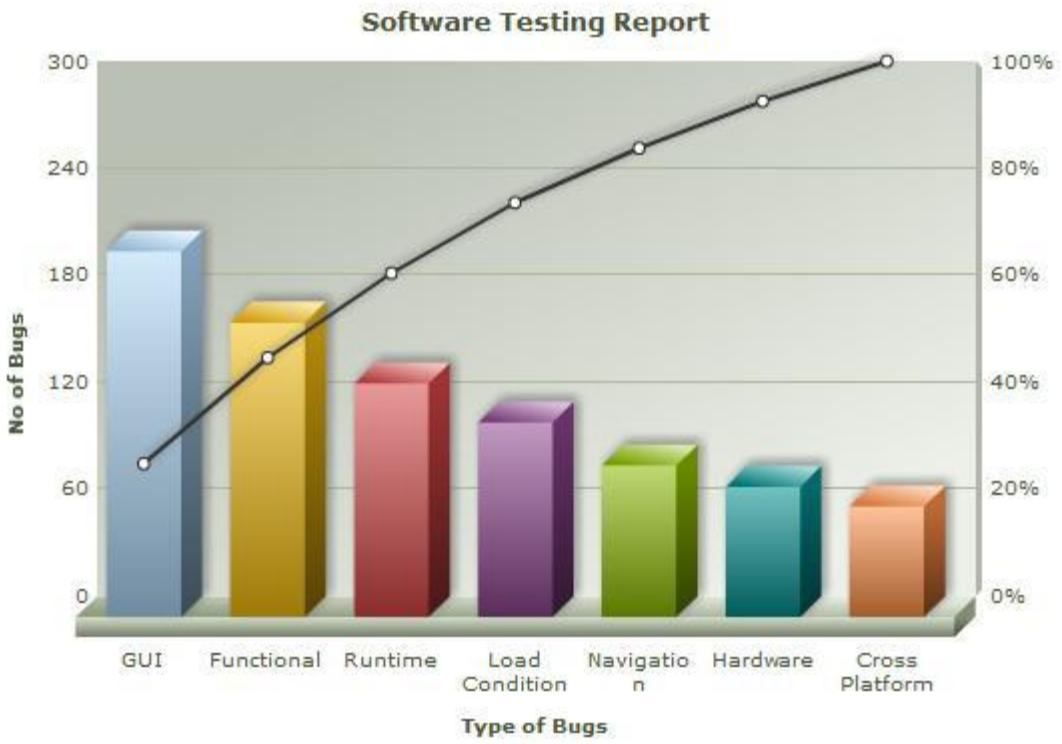


（类型：Marimekko）

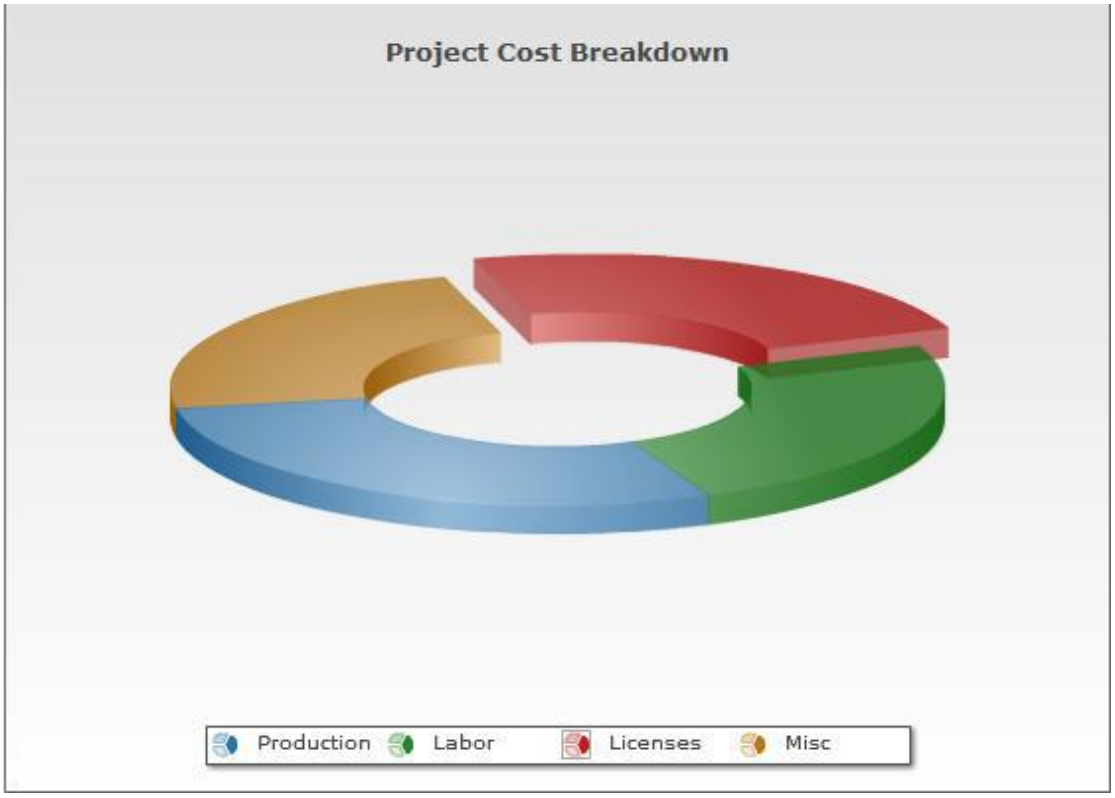


网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com

(类型：Pareto2D)



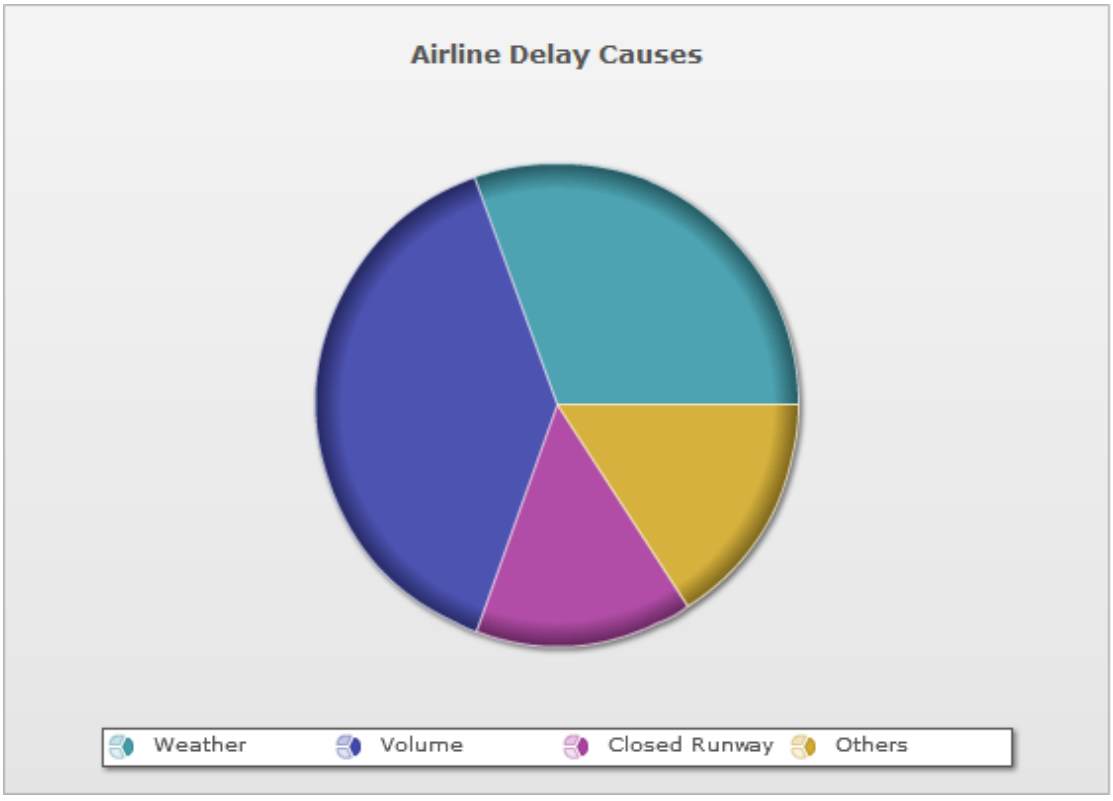
(类型：Pareto3D)



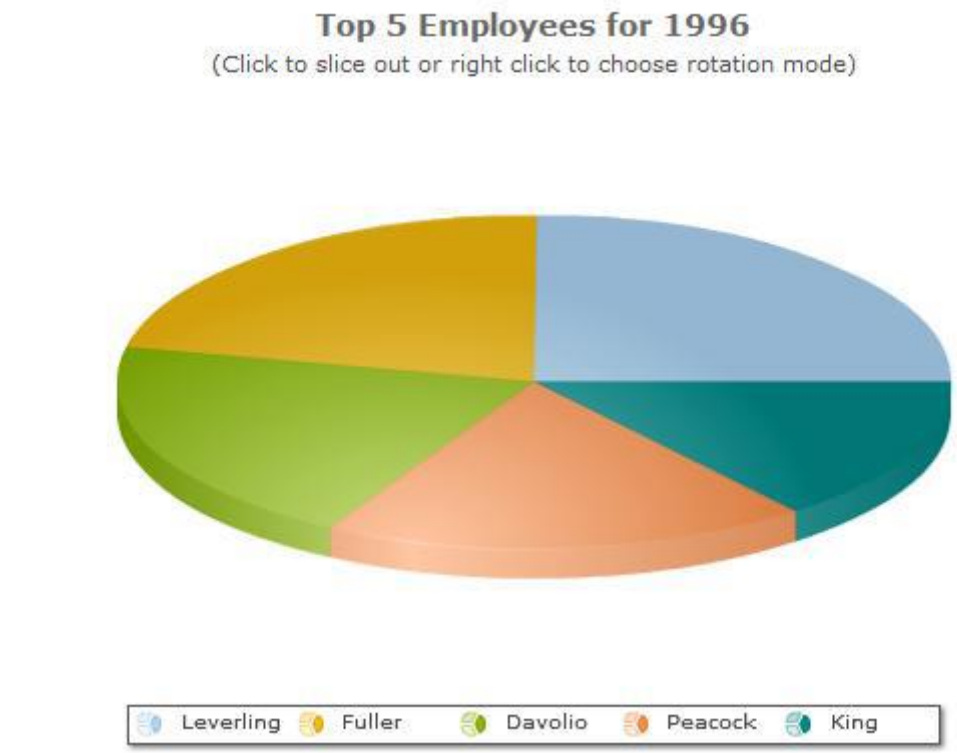
(类型：Doughnut3D)

网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com

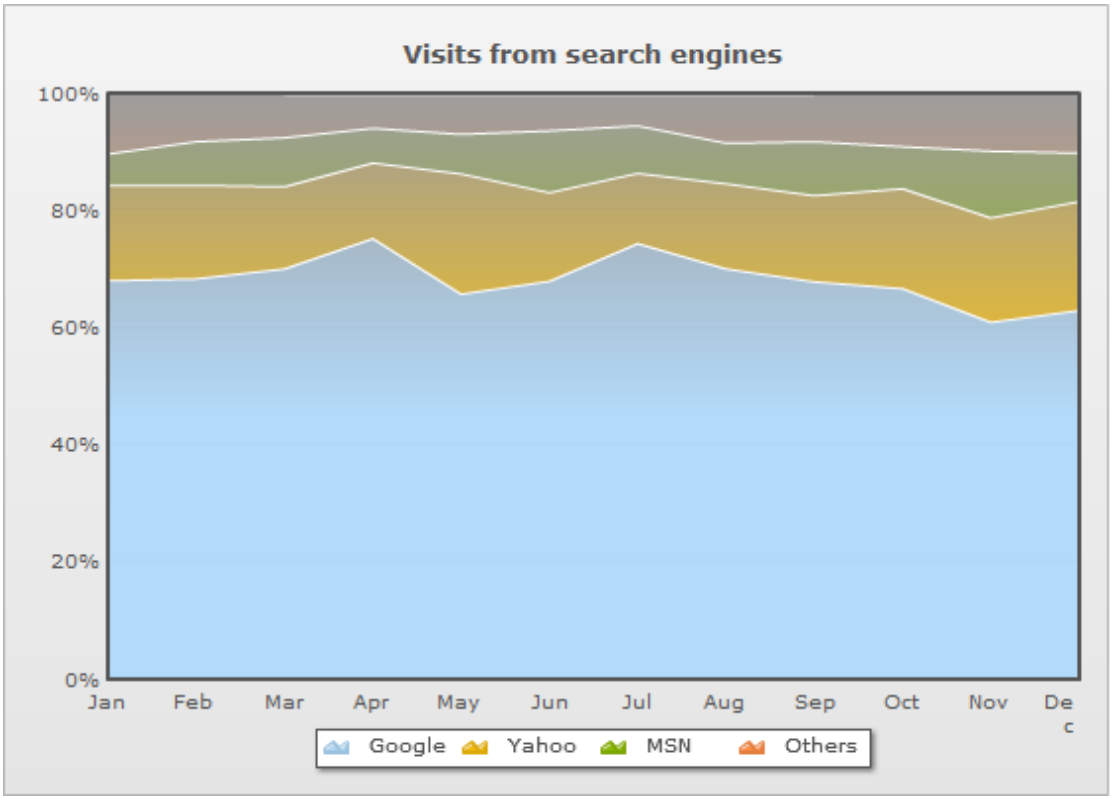
(类型：Doughnut2D)



(类型：Pie2D)



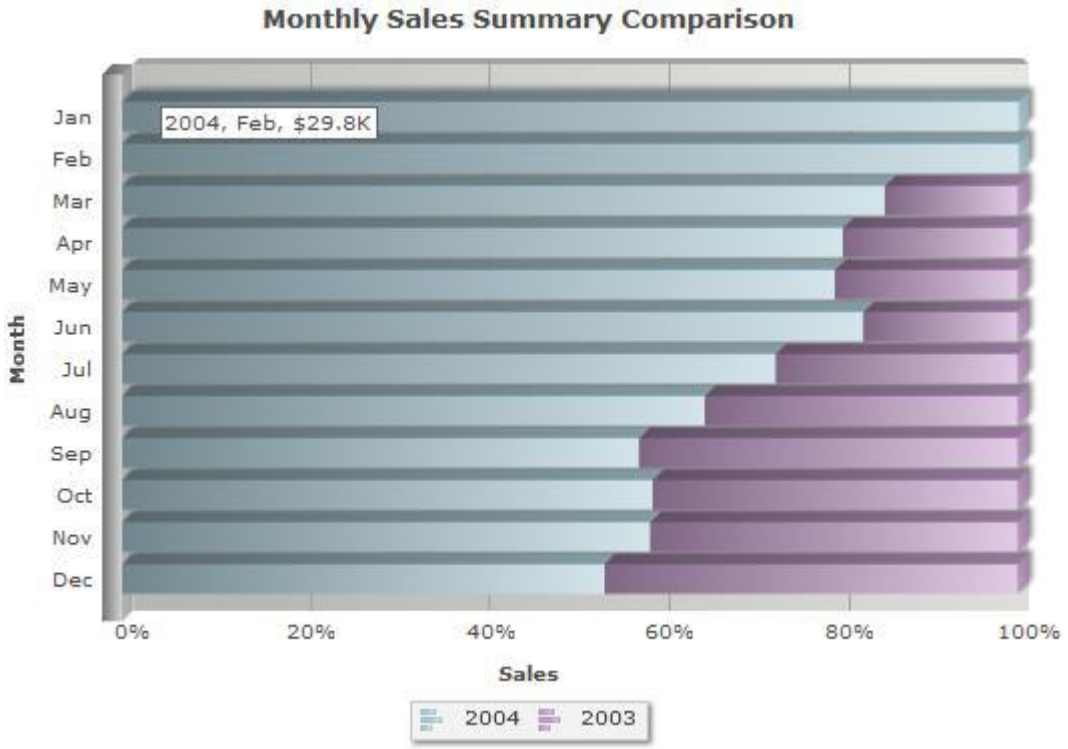
(类型：Pie3D)



网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com

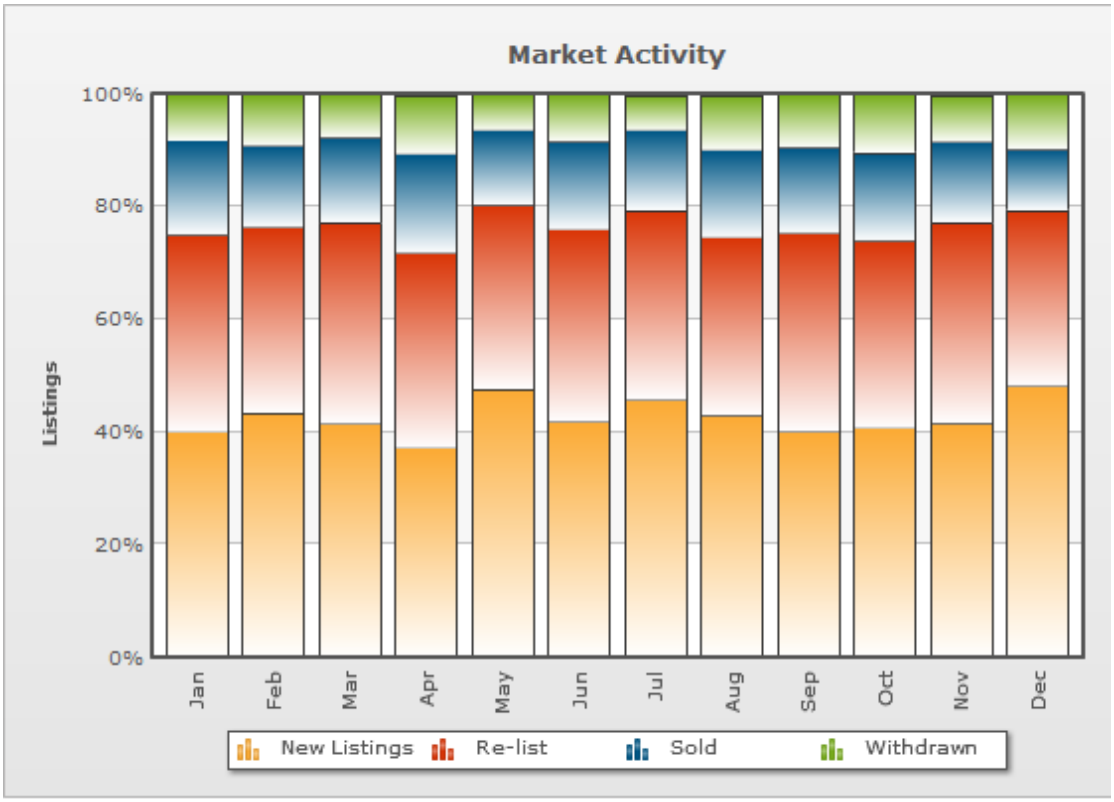
(类型：StackedArea2D)

(类型：StackedBar2D)

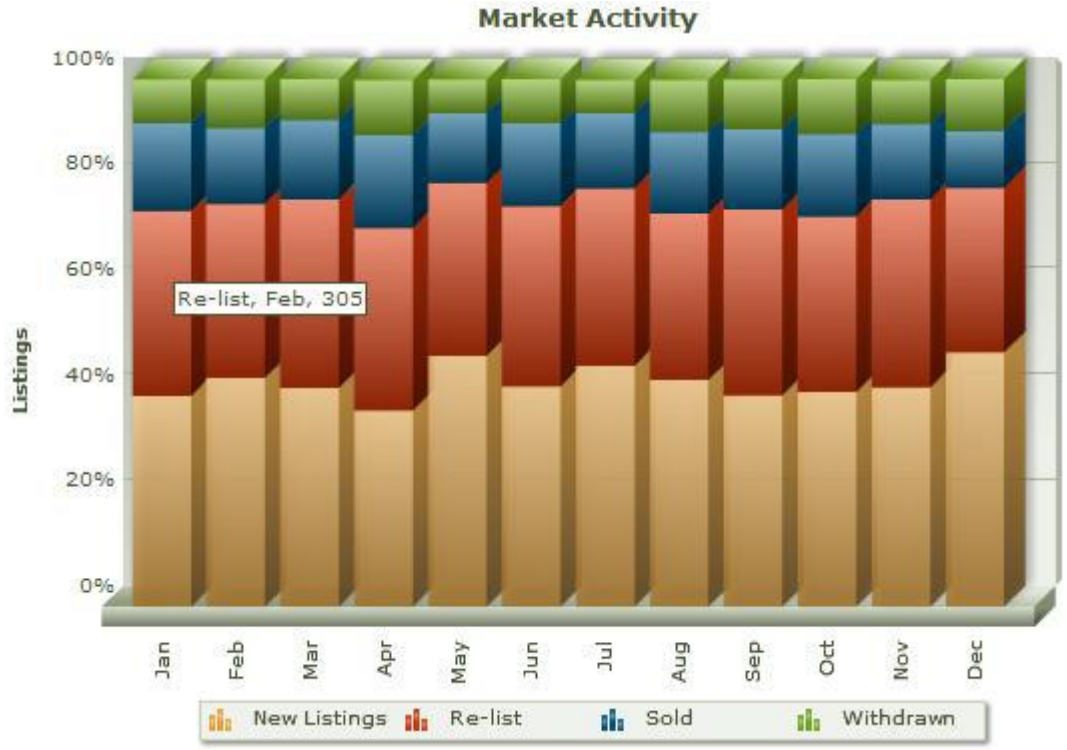


(类型：StackedBar3D)

网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com



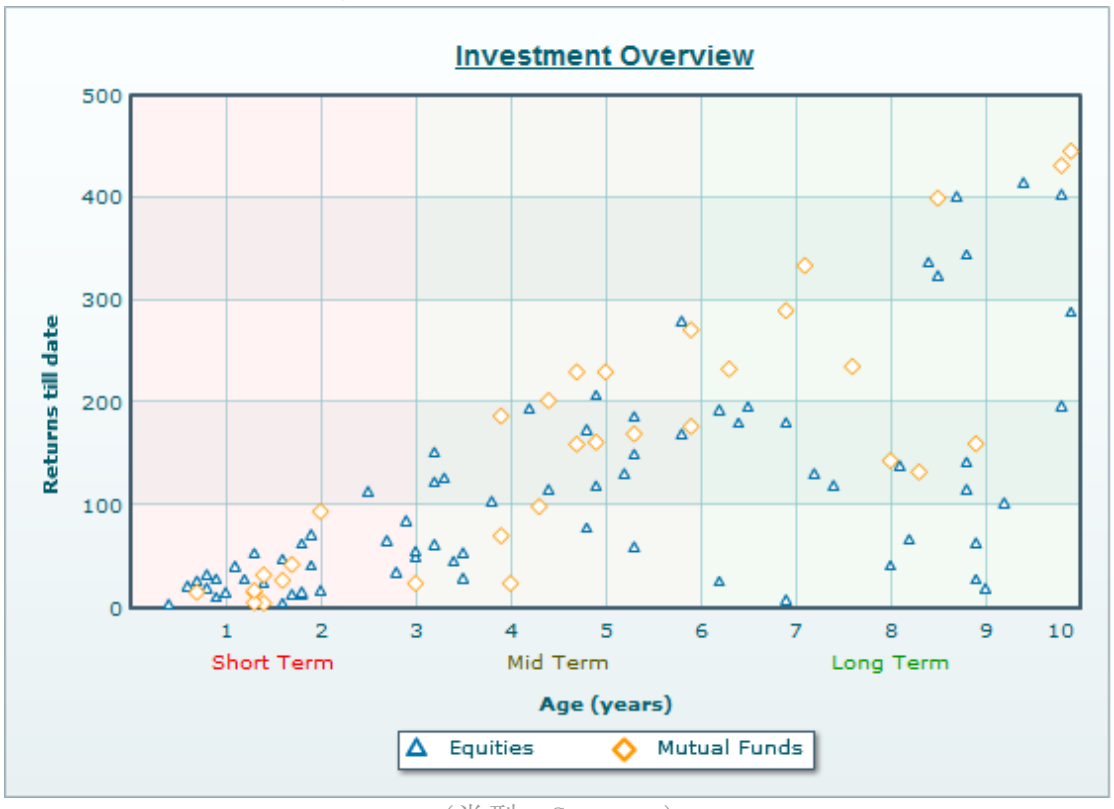
(类型：StackedColumn2D)



(类型：StackedColumn3D)

网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com

(类型：StackedColumn3DLineDY)



(类型：Scatter)

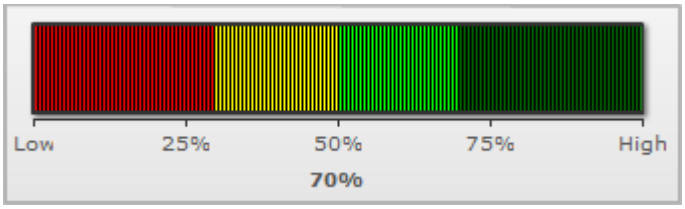
网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com

(类型：ZoomLine)

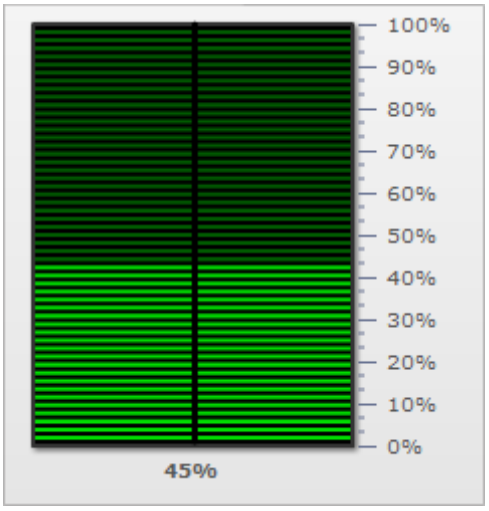
(类型：AngularGauge)

网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com

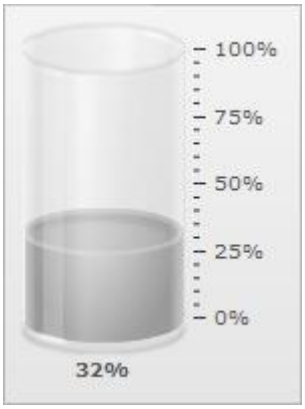
(类型：HLinearGauge)



(类型：HLED)

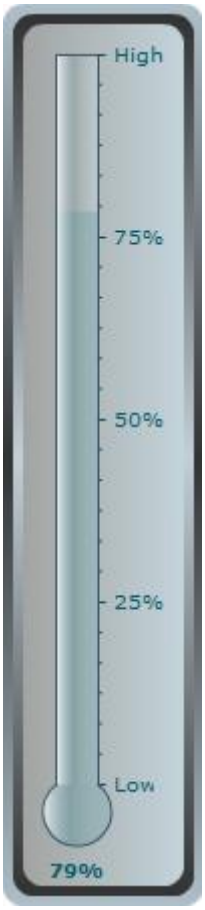


(类型：VLED)

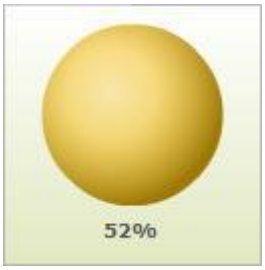


(类型：Cylinder)

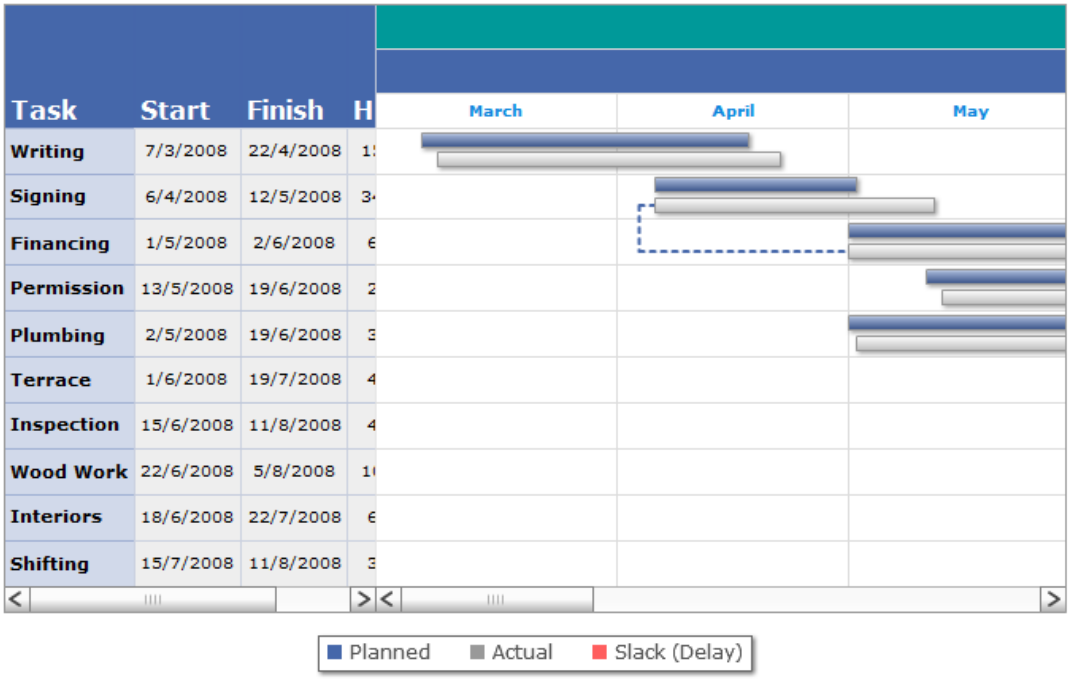
网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com



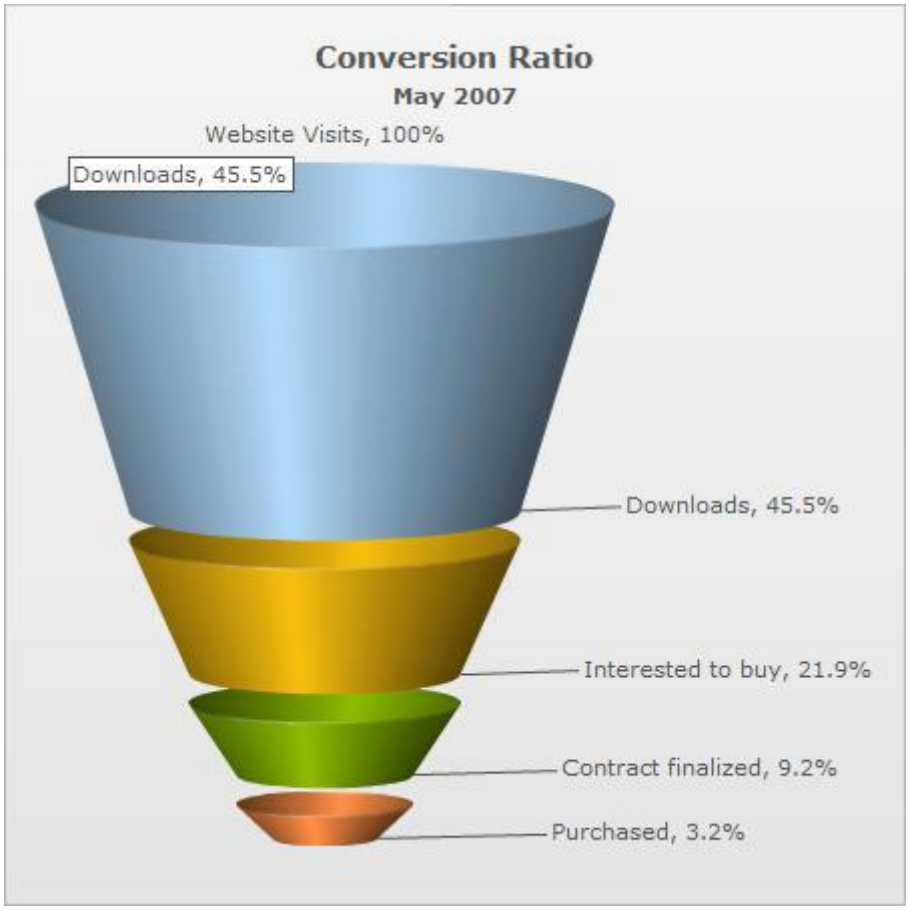
(类型：Thermometer)



(类型：Bulb)

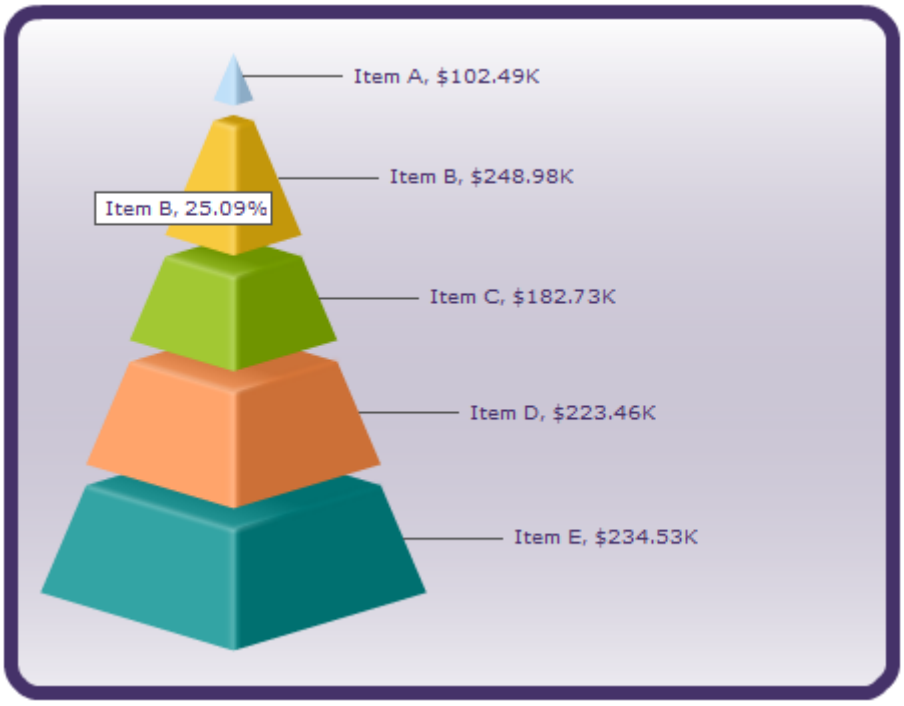


(类型：Gantt)

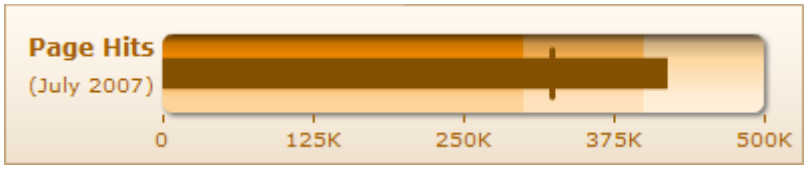


网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com

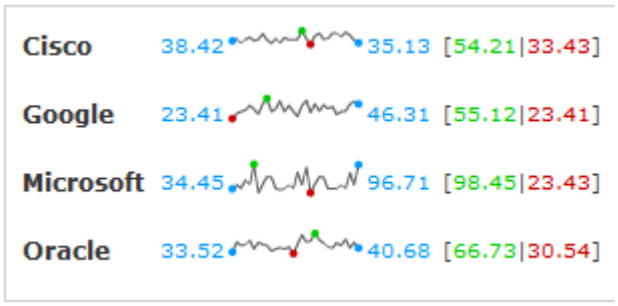
(类型：Funnel)



(类型：Pyramid)

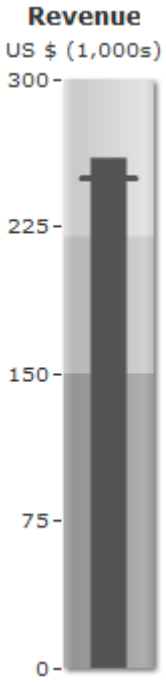


(类型：HBullet)

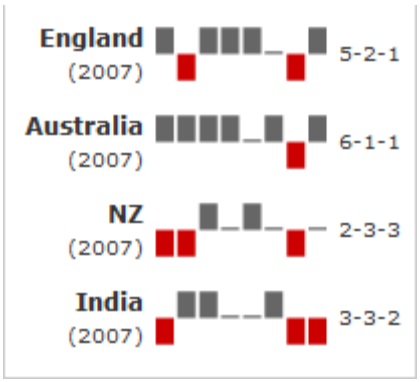


(类型：SparkLine)

网址：http://www.97521.com
电话：(028)85145708，85143878，85147698
电邮：xiefangjian@163.com



(类型：VBullet)



(类型：SparkWinLoss)