JS多级命名空间

JavaScript不像C#或Java，有专门的namespace和package语法支持，当JS复杂到一定程度，尤其是引用大量的第三方JS框架和类库之后，命名冲突就会成为一个严重的问题，因此使用JS自己的变通方式建立命名空间很重要。

最简单的实现命名空间的方法，代码示例如下：

//这是伪代码,:   
var com;       
if(!com) com = {}; //第一级域名       
com.ModuleClass = {}; //第二级域名       
com.ModuleClass.函数名1=function(){       
    函数体；       
}     
com.ModuleClass.函数名2=function(){       
    函数体；       
}

还可以建立一个注册多级命名空间的机制：  
//1、命名空间注册工具类       
var Namespace = new Object();       
               
Namespace.register = function(path){       
    var arr = path.split(".");       
    var ns = "";       
    for(var i=0;i<arr.length;i++){       
        if(i>0) ns += ".";       
        ns += arr[i];       
        eval("if(typeof(" + ns + ") == 'undefined') " + ns + " = new Object();");       
    }       
}       
       
//2、注册命名空间 com.boohee.ui       
Namespace.register("com.boohee.ui");       
       
//3、使用命名空间       
com.boohee.ui.TreeGrid = function(){       
    this.sayHello = function(name){       
        alert("Hello " + name);       
    }       
}       
       
var t = new com.boohee.ui.TreeGrid();       
t.sayHello("uid");

===========================================================================

/\*\*\* 构造命名空间 \*\*\*/  
/\*  
\* if (typeof(window.Com) == "undefined")//也可以使用：if (typeof(Com) == "undefined")  
{  
window.Com = {};//也可以使用：window.Com = new Object();  
}

\*/  
//一级：BrcLib  
if(typeof(BrcLib) == "undefined")  
{  
    BrcLib = new Object();  
}  
//二级：BrcLib.JavaScript  
if(typeof(BrcLib.JavaScript)=="undefined")  
{  
    BrcLib.JavaScript = {};  
}  
//三级：BrcLib.JavaScript.File  
if(typeof(BrcLib.JavaScript.File)=="undefined")  
{  
    BrcLib.JavaScript.File = {};  
}  
//三级：BrcLib.JavaScript.Message  
if(typeof(BrcLib.JavaScript.Message)=="undefined")  
{  
    BrcLib.JavaScript.File = {};  
}

//静态方法  
BrcLib.JavaScript.Message.ShowMessage = function(ms)  
{  
alert(ms);  
}

//必须使用new后的方法  
BrcLib.JavaScript.Message.prototype.formatString = function(format)  
{  
//code  
}