

TWaver[®] 3D 机房管理软件

用户手册

(版本 : 1.2)

2010 年 10 月

赛瓦软件


info@servasoftware.com

<http://www.servasoftware.com>

PO Box 8143, Wichita Falls, Texas, USA



版权声明

Copyright © 2010, Serva Software 公司保留所有版权。TWaver、servasoft 、Serva Software 和  servasoft 均为 Serva Software 公司的注册商标。本文中涉及的其他品牌是各自公司的注册商标，在此仅作为标识目的。

修改记录

版本编号	日期	备注
A	2010-6-3	v1.0
B	2010-10-8	v1.2

目 录

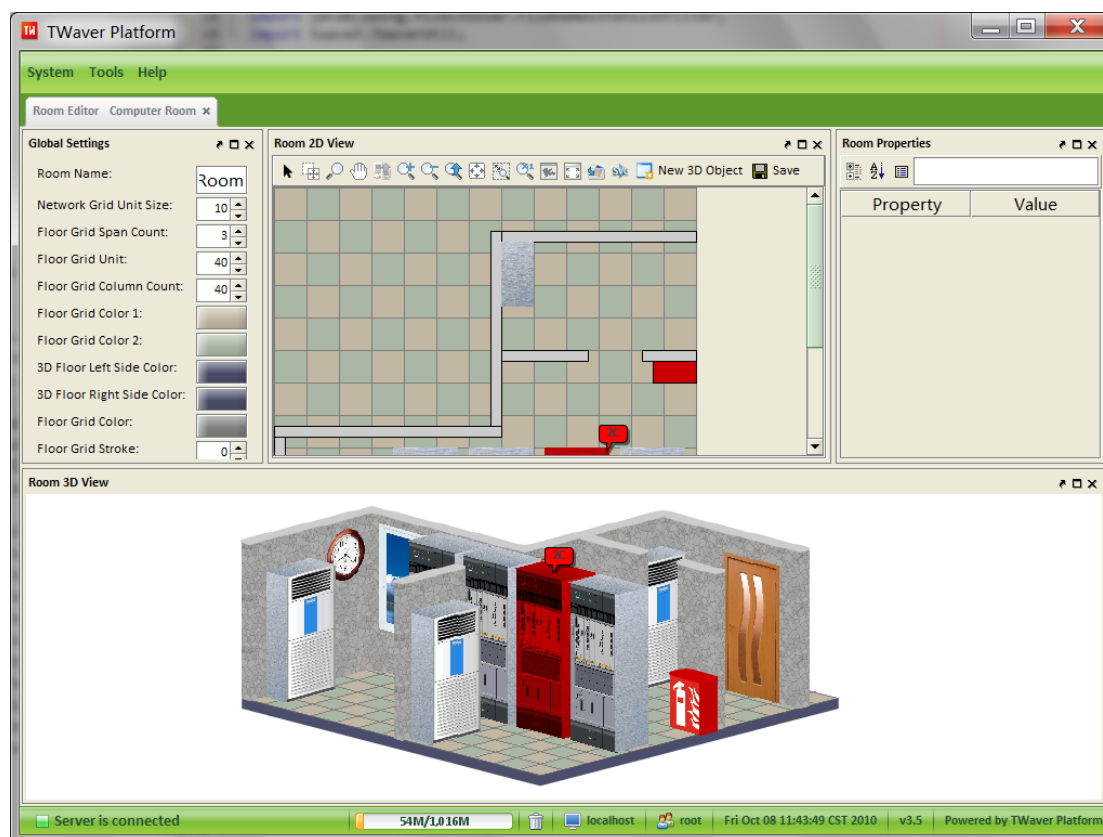
概述.....	1
什么是 TWaver® 3D 机房管理软件	1
产品结构	2
目标用户	2
安装.....	3
安装 MySQL 数据库	3
安装	12
初始化.....	14
常用操作.....	17
创建机房	17
保存机房	18
打开机房	19
创建机房物体	19
设置机房环境参数.....	23
常用编辑操作	25
一个完整的例子.....	28
开发指南.....	35
开发环境	35
类结构.....	35
简单例子	36
创建简单的编辑器.....	38

Element3D 主要属性	43
Network3D 主要属性.....	44
数据的输入与输出	45
更多信息.....	48
技术支持	48
关于赛瓦软件	48

概述

什么是 TWaver® 3D 机房管理软件

TWaver® 3D 机房管理软件（以下简称“3D 机房”）是用于管理电信运营商机房资源的软件平台，是赛瓦软件 TWaver®产品家族中的一员。3D 机房基于 TWaver 产品家族中的 TWaver Java 产品分支进行创建和扩展，是基于 Java 和 Swing 技术的桌面应用程序。3D 机房通过集成设计环境（IDE）对电信行业各种机房资源进行设计、建模、存储，并以三维技术进行呈现和展示。3D 机房既是一个完善的电信机房资源管理工具，也是一个电信资源管理的二次开发平台和组件。



产品结构

TWaver3D 机房是一个结构简单的桌面应用程序，包括数据编辑环境和数据存储两个部分。3D 机房的功能包括 2D 机房设计器、3D 机房呈现试图、数据转换与存储模块。通过 3D 机房，用户可以方便快速的创建电信机房环境，并对各种机房资源进行管理和存储。这些机房数据通过 XML 格式进行输出，并存储在文件系统或者数据库中。这些数据资源遵循 TWaver 和 JavaBeans 相关规范，可以被其他网络管理软件共享（如网管系统、资源管理系统等）。

目标用户

TWaver3D 机房可以为广大电信运营商、设备制造商、软件开发商所使用，增加开发效率，提高软件的用户体验。尤其在以下几种情况下，TWaver3D 机房将会发挥其更大的作用：

- 电信网管软件开发者：将 TWaver3D 机房作为二次开发平台，开发符合自身需要的机房资源管理软件系统；
- 系统集成商：可将 TWaver3D 机房与现有软件系统整合，提供完整的机房资源管理能力；
- TWaver 开发者：将 TWaver3D 机房作为呈现组件，对各种机房资源进行三维展示；
- 电信运营商：直接使用 TWaver3D 机房，管理现有机房资源；

安装

安装 MySQL 数据库

TWaver3D 机房支持将数据直接存放到数据库中，也可以将数据存放在文件系统中。所以，数据库系统并非是 TWaver3D 机房所必须的一部分。如果需要将数据存储到数据库，或需要在文件系统和数据库系统中进行数据转换和迁移，则需要安装数据库系统；如只需要把数据存储到文件系统中，可跳过本章。

TWaver3D 机房目前支持安装 MySQL 5.0 或以上版本。按照以下步骤进行 MySQL 数据库的安装。

第一步：到 www.mysql.com 网站下载 MySQL 5.0 或以上版本；

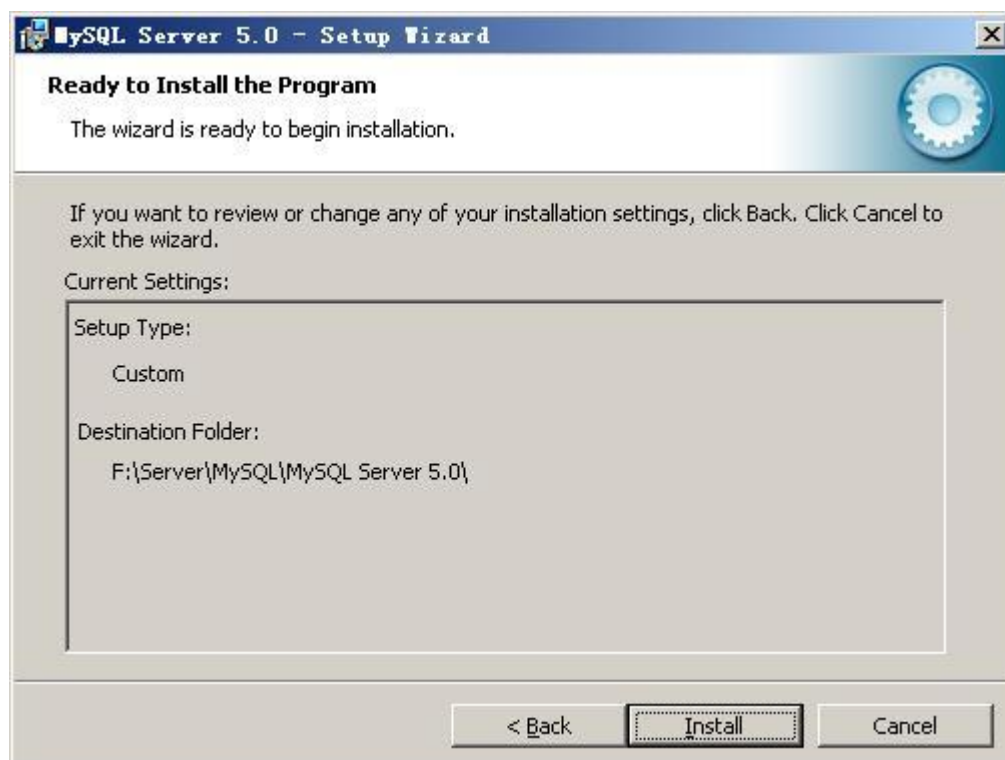
第二步：解压后双击 setup.exe 进行安装。按照以下图示进行选择：



点击“Next”按钮继续；



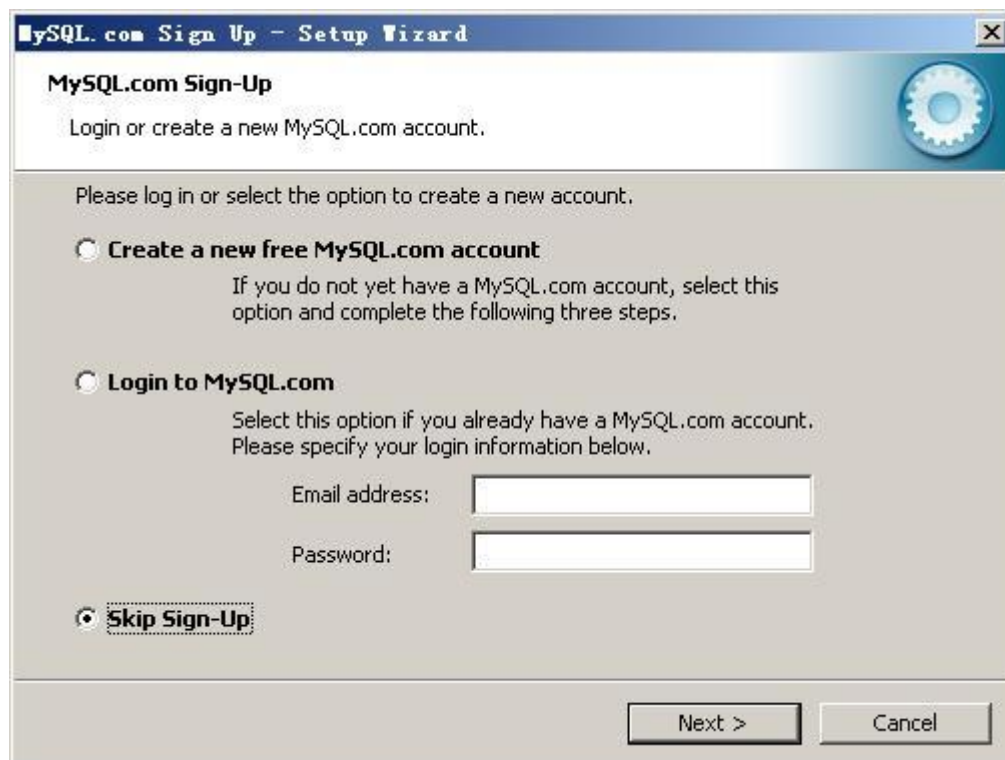
安装类型有 Typical (典型安装)、Complete (完整安装) 和 Custom (自定义安装) 三个选项。选择 Typical 或 Complete 均可。点击 “Next” 按钮继续；



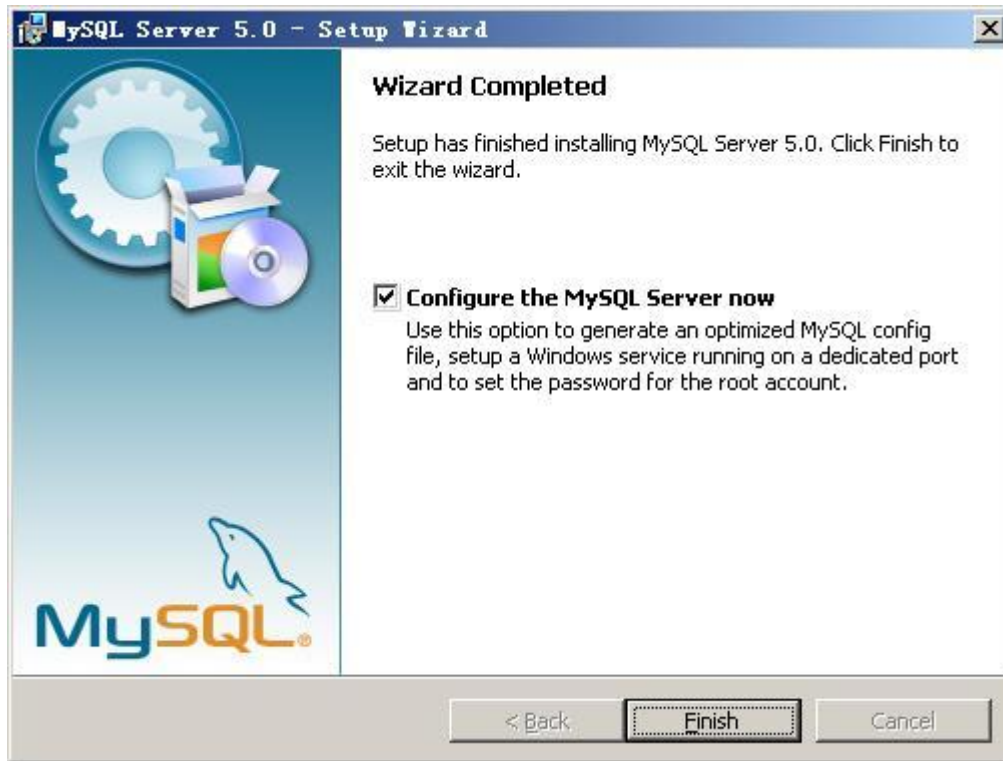
安装前的最后确认。如有误，可点击“Back”按钮返回修正。点击“Install”按钮进行安装；



正在安装中，请稍后，直到出现以下界面：



这里询问是否要注册 mysql.com 的账号。选择“Skip Sign-Up” ,点击“Next”按钮忽略此步继续；



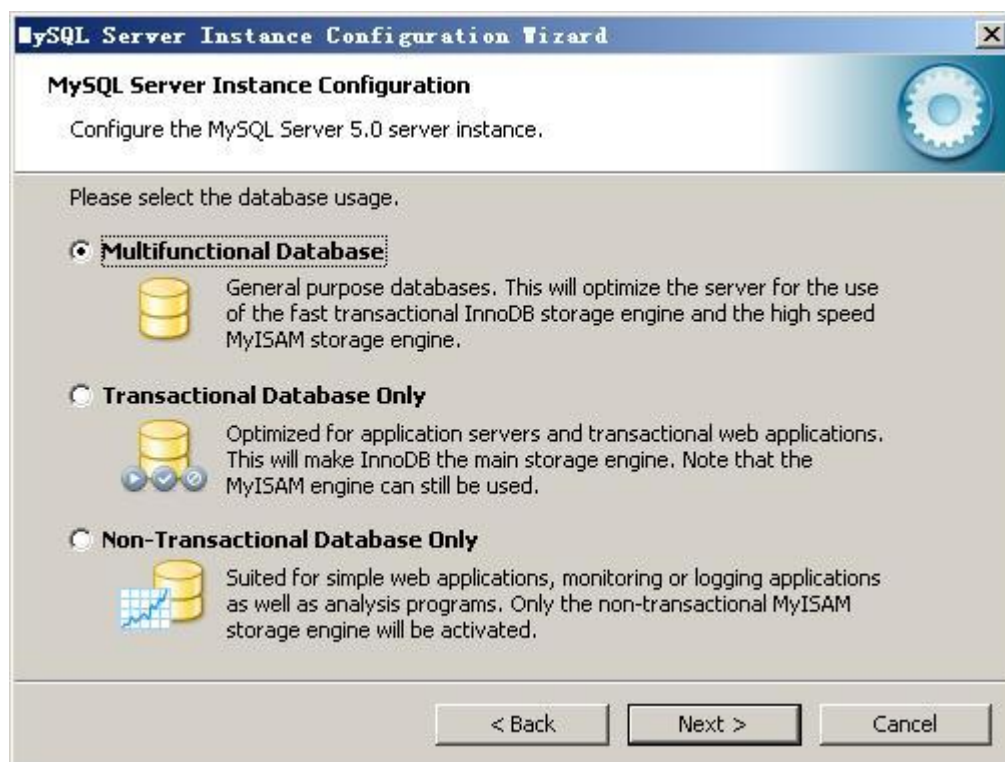
在软件安装结束后，在“Configure the MySQL Server now”前打钩，然后点击“Finish”按钮进行 MySQL 配置；



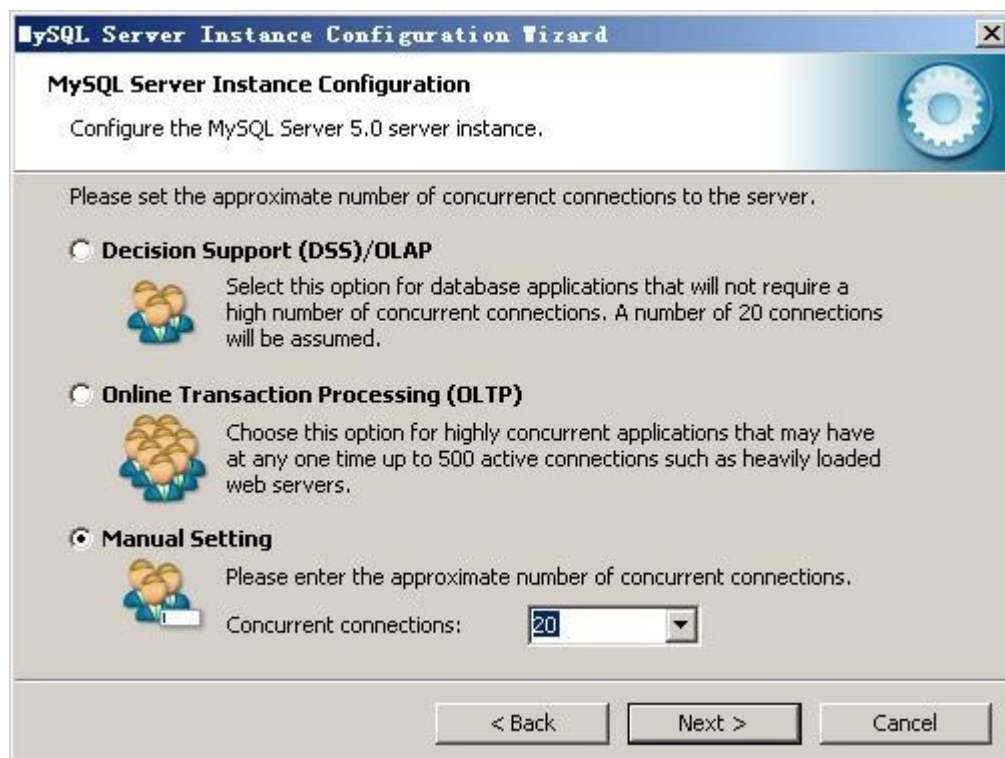
选择 “Detailed Configuration”，点击“ Next ”按钮继续；



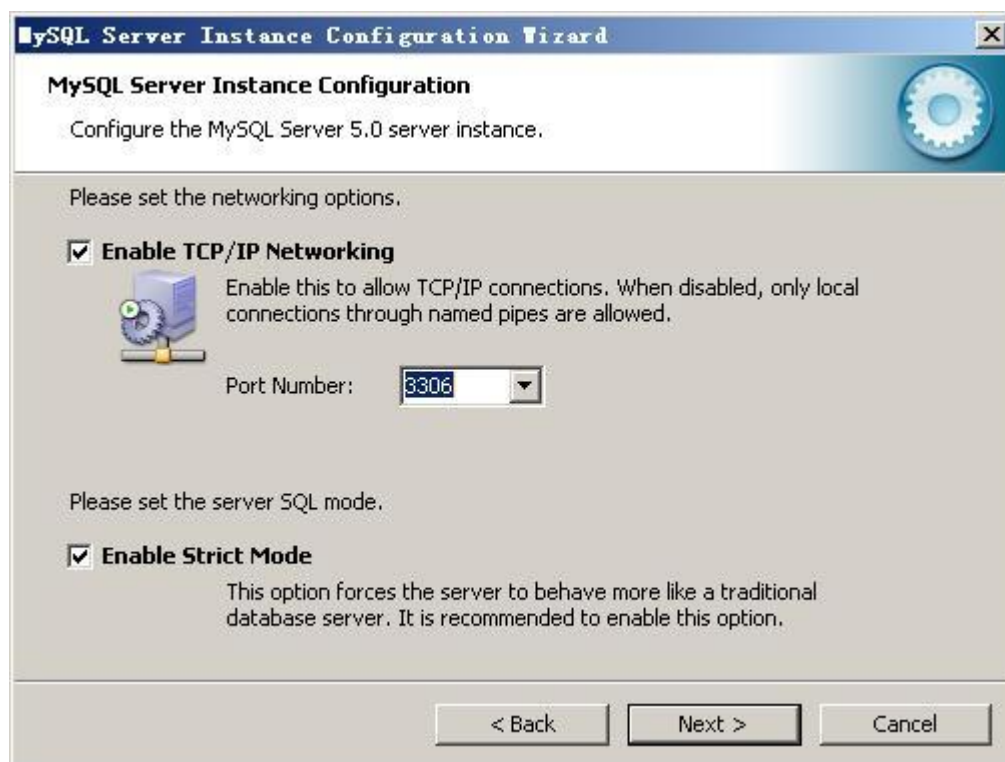
选择 “Server Machine ”，点击“ Next ”按钮继续；



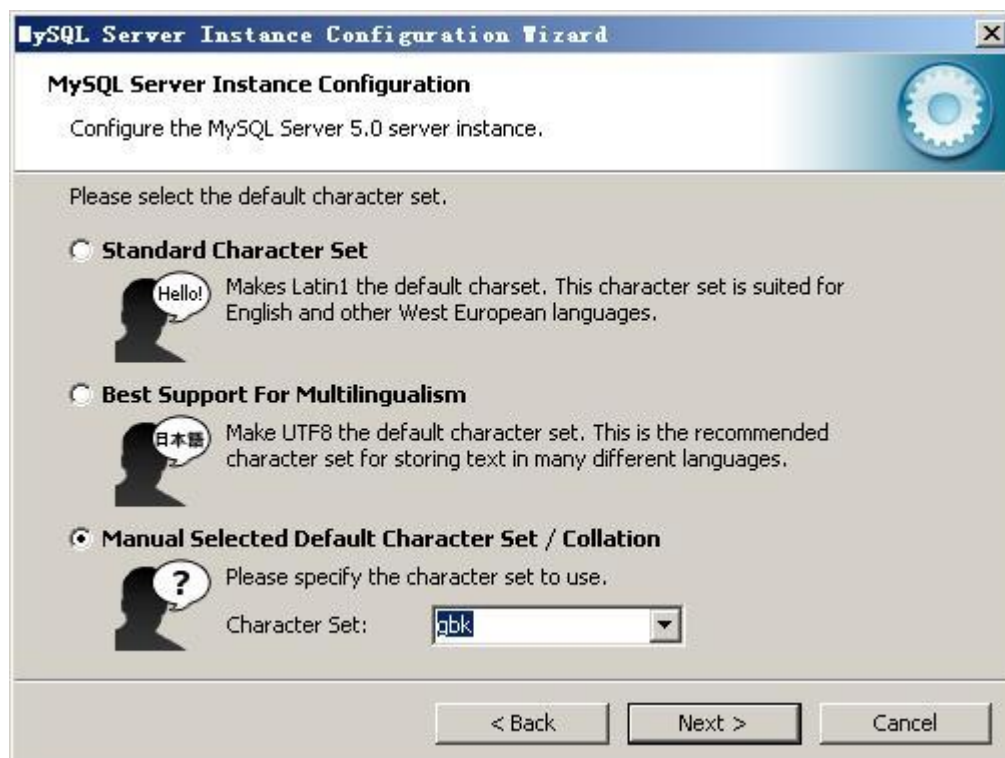
选择 “Multifunctional Database ”，点击“ Next ”按钮继续；



保持默认选择，点击 “Next ”按钮继续；



保持默认端口设置，点击“Next”按钮继续；



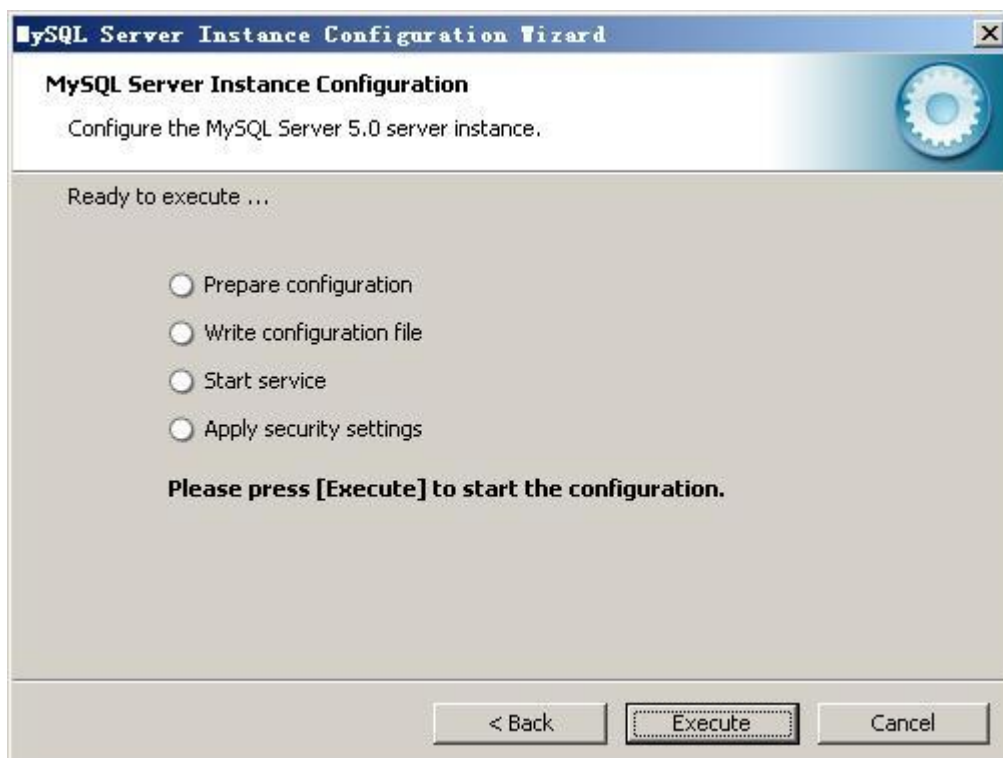
选择 MySQL 编码。选择 “Manual Selected Default Character Set/Collation” 在 Character Set 文本框中输入 “GBK” 或 “UTF8” 点击 “Next” 按钮继续；



勾选所有设置，点击 “Next” 按钮继续；



输入 root 账号密码。在两个文本框中输入 “public ”, 点击 “Next ”按钮继续；



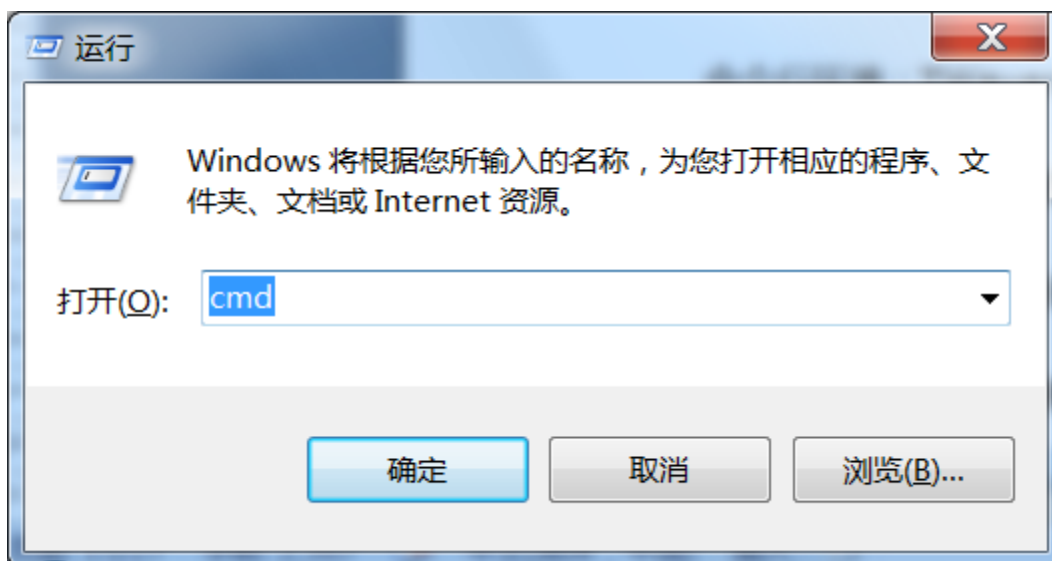
点击 “Execute ”按钮进行设置；

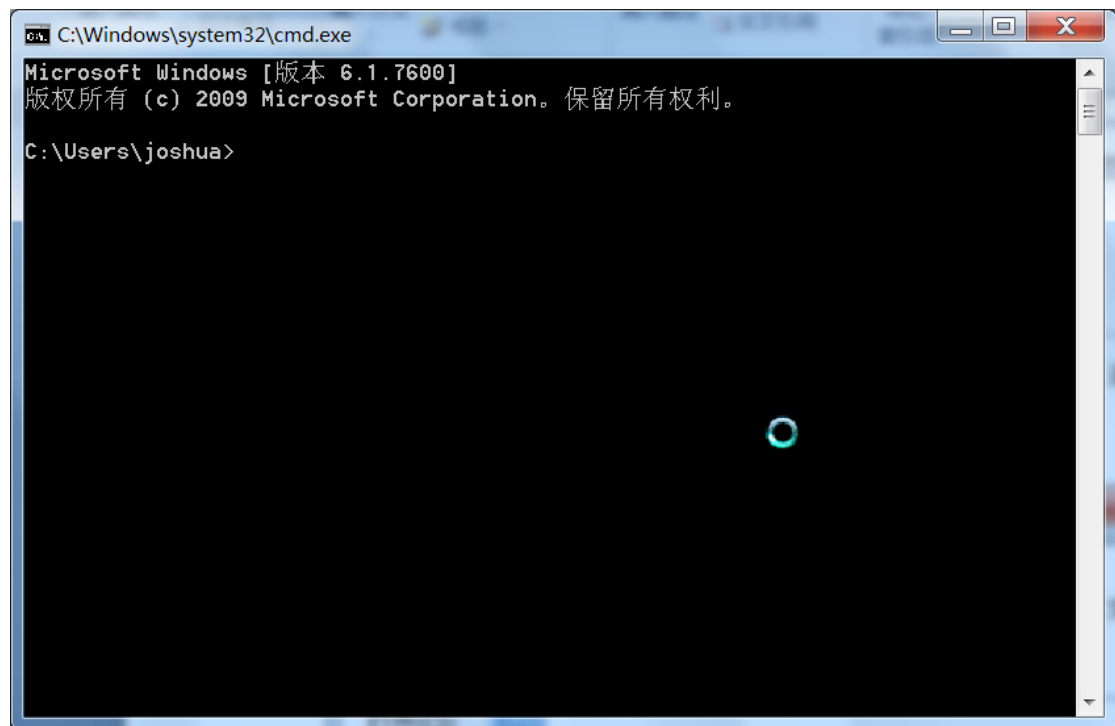


MySQL 安装结束。

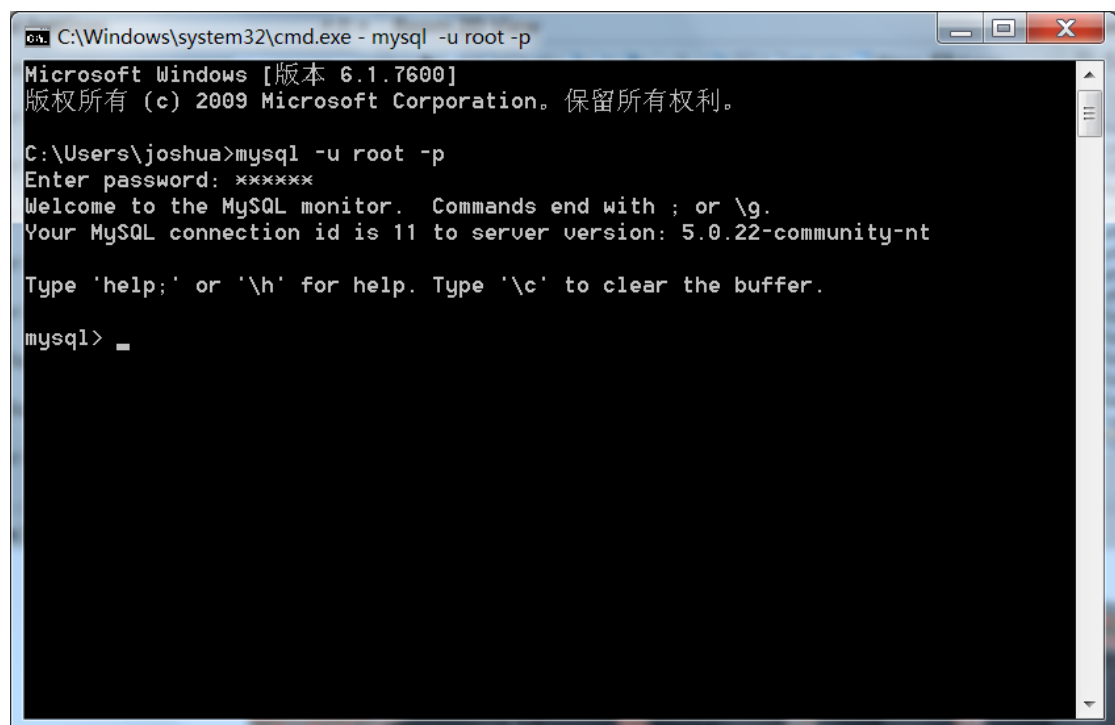
安装

第一步：在 Windows 中，点击“开始”->“运行”，输入 CMD 回车，进入 DOS 命令行环境；





第二步：输入 `mysql -u root -p` 回车，输入 `public`，进入 MySQL；



第三步：输入 `create database twaver_platform` 创建数据库；

第四步：输入 `exit` 退出 MySQL 环境；

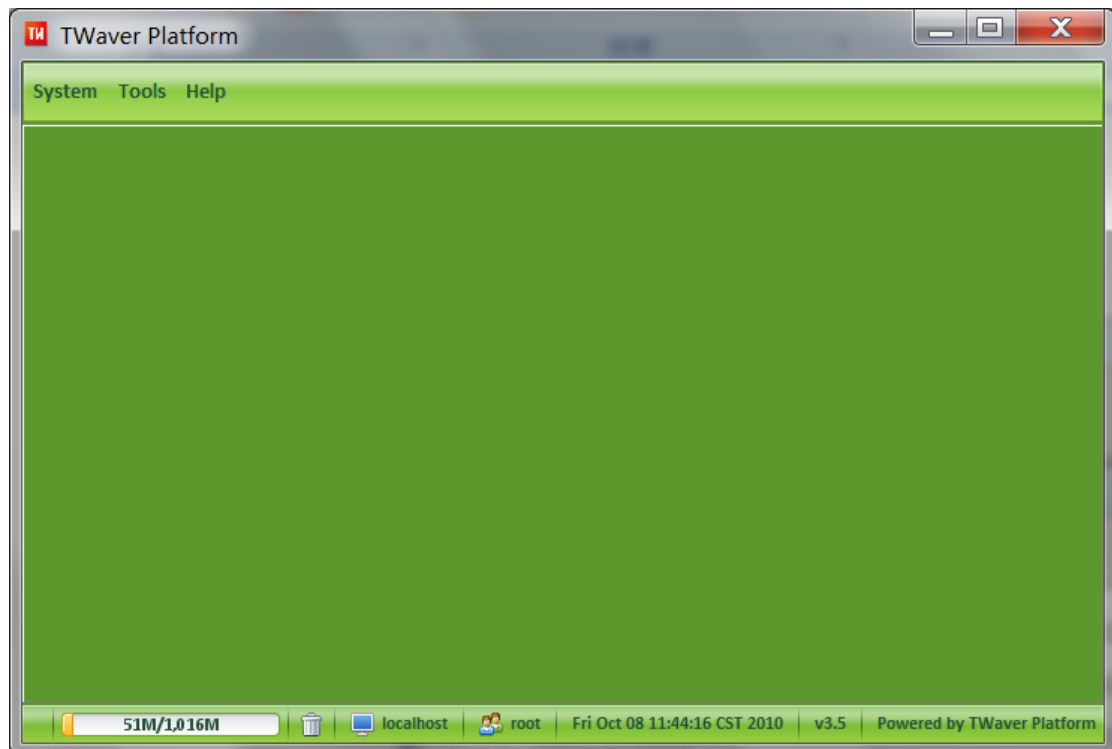
第五步：在 TWaver3D 机房的软件 zip 压缩文件中找到 database.sql 文件，并解压到硬盘，例如 c:\根目录；

第六步 输入 `mysqldump -u root -p twaver_platform < c:\database.sql` 命令将数据库脚本导入。回车后输入数据库密码 public；

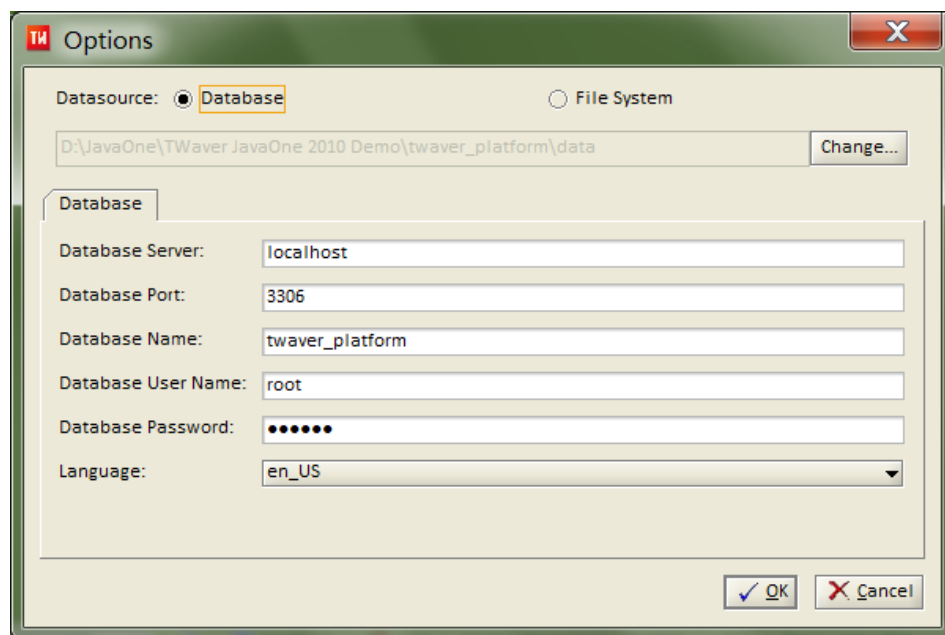
第七步：解压 TWaver3D 机房 zip 软件包，双击 run.bat 启动系统；

初始化

启动 TWaver3D 机房后，进入程序主页面，如下图：



点击主菜单 Tools——Options，打开系统设置界面：

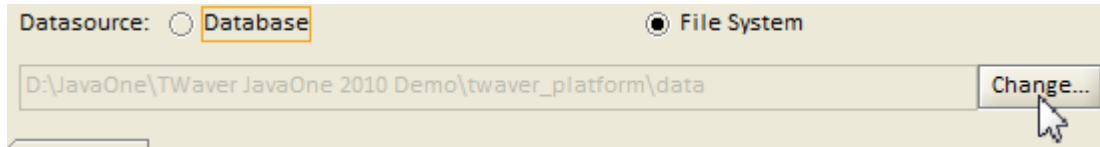


首先要设置数据保存的方式。TWaver3D 机房提供两种数据保存方法：数据库和文件系统。如果选择 Database，则所有数据的存取将直接针对数据库进行；如果选择 File System，则需要指定一个具体的文件目录，TWaver3D 机房会将数据分门别类存储到该目录的具体子目录中。

如果选择数据库模式，则确保设置以下数值：

- Datasource：选择 Database 一项；
- Database Server：数据库服务器地址。输入 localhost（本机）；
- Database Port：数据库服务器端口。输入 MySQL 默认端口 3306；
- Database Name：数据库名称。输入 twaver_platform；
- Database User Name：数据库用户名。输入 root；
- Database Password：数据库密码。输入 public；
- Language：可根据需要选择图形界面使用的语言；

如果选择文件模式，则确保设置好文件存储的目录位置。点击 Change 按钮，选择一个具体目录。



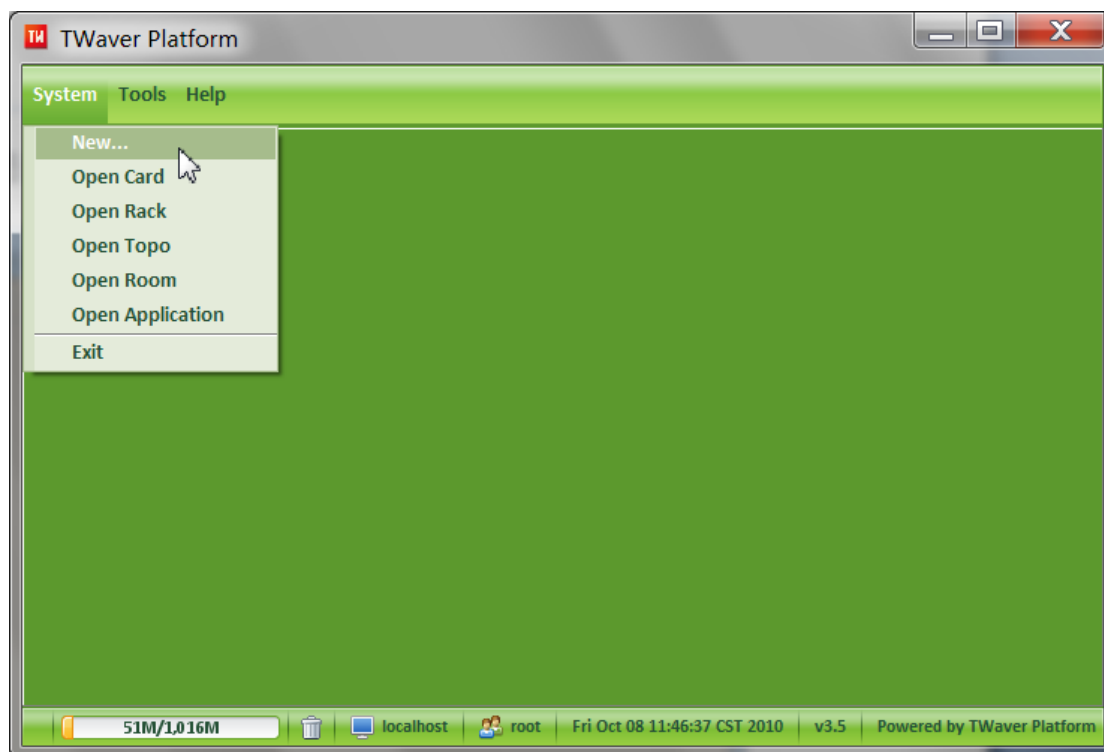
点击 OK 保存设置。

至此，TWaver3D 机房已经安装设置完毕，可以使用。以上设置也可以在以后使用过程中随时更改。所有设置的具体参数会被保存在用户计算机本地目录中。

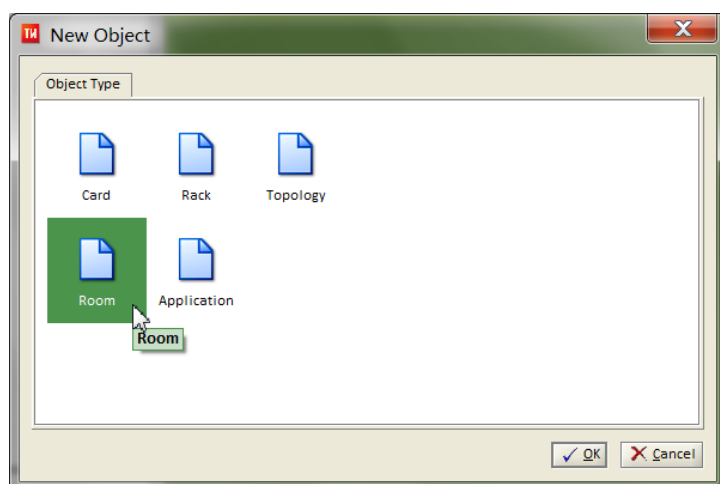
常用操作

创建机房

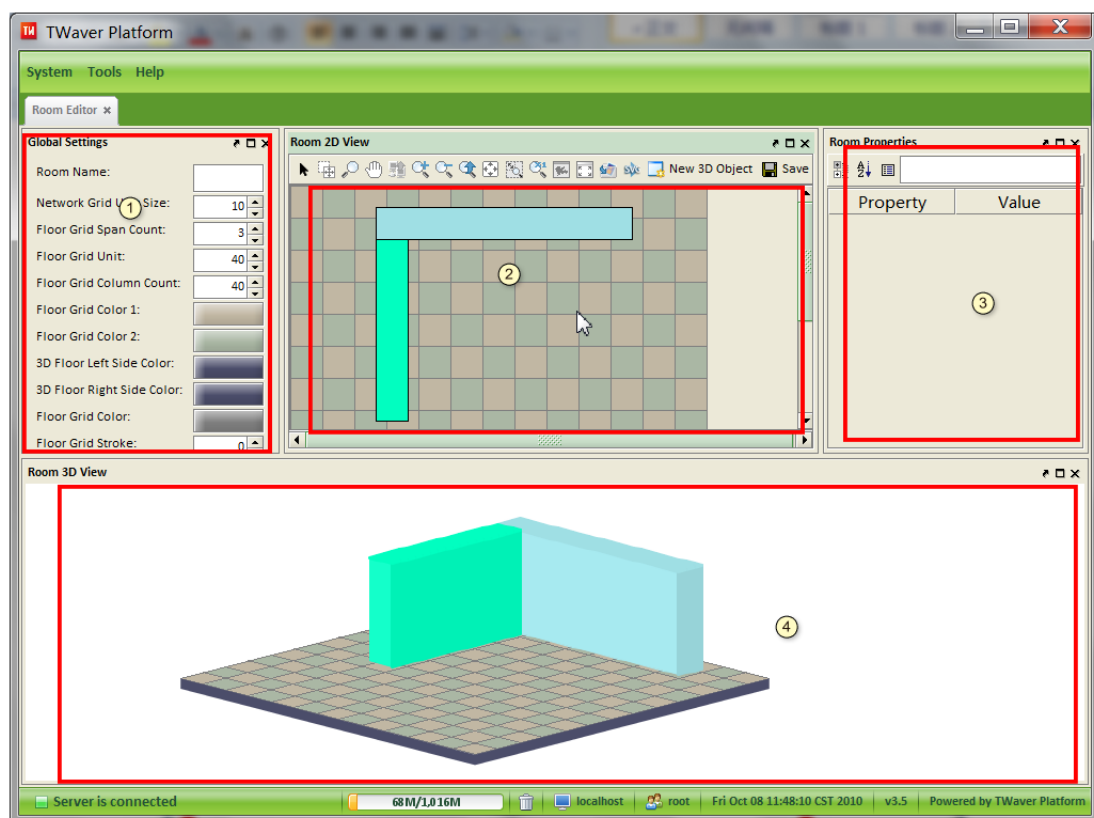
在 TWaver3D 机房中，点击 System——New 菜单：



在对话框中选择“Room”，点击 OK 按钮，创建新的机房对象。



机房集成编辑环境如下图所示。



其中各个面板作用如下：

1. 机房环境参数设置。包括房间大小、填充颜色等；
2. 机房 2D 设计视图。可以在这里创建对象、移动对象、缩放、参数设置等；
3. 对象属性表。可以对选中对象进行各种参数设置；
4. 3D 设计视图。对 2D 设计视图中的数据进行 3D 视图自动转换和呈现。

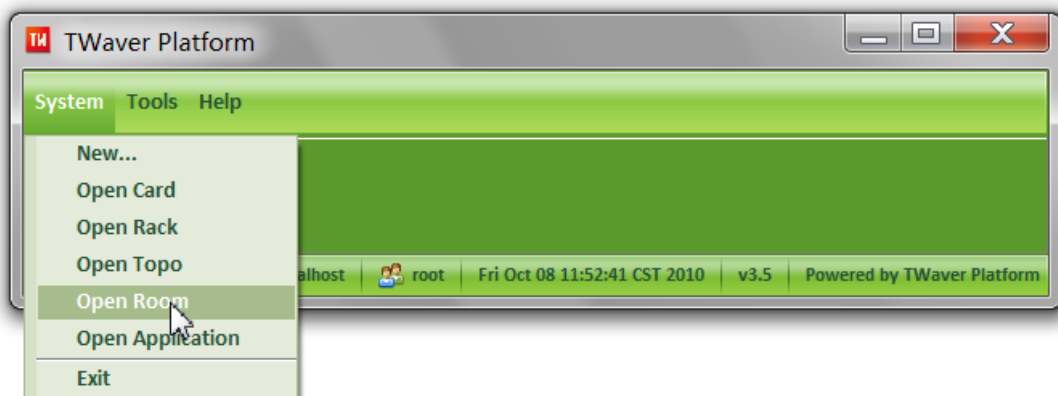
保存机房

在 TWaver3D 机房中，点击 2D 设计视图中工具条的“Save”按钮，可以将当前机房保存到数据库或者文件系统中，这取决于如何设置保存选项。

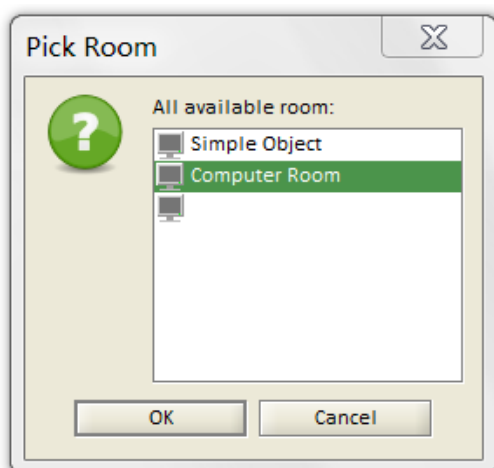


打开机房

一旦机房被保存到数据库或者文件系统中,即可在任意时间打开机房数据进行修改。点击主菜单中的 System——Open Room 菜单项:



在弹出的对话框的列表中,选中要编辑的机房,点击“确定”按钮即可打开机房:

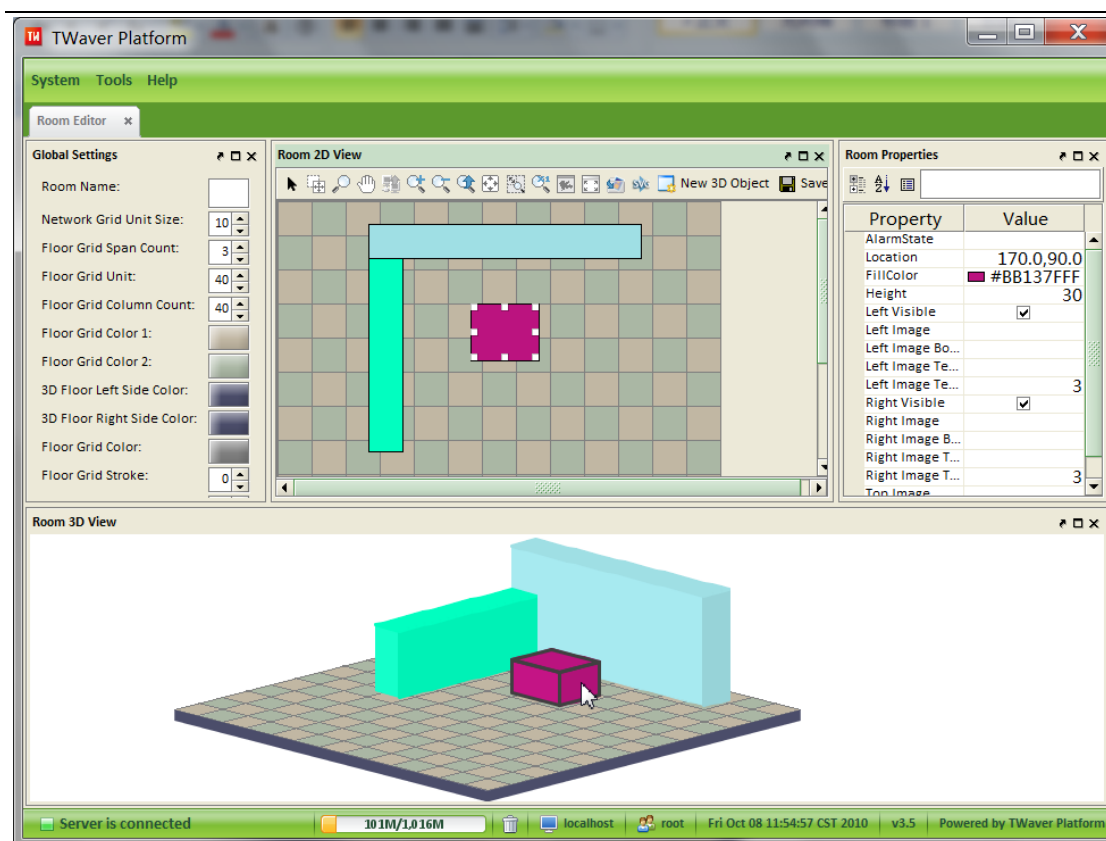


创建机房物体

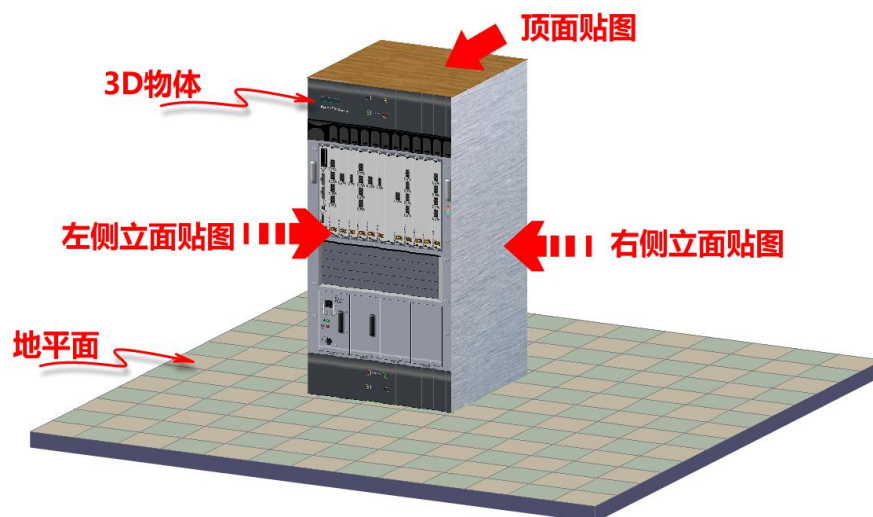
点击工具栏的“New”按钮,可以创建一个新的 3D 机房物体。



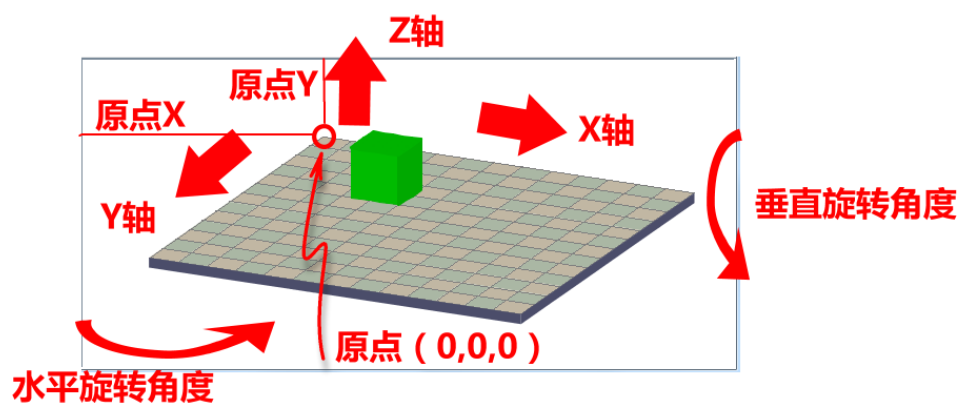
新建的机房物体会在 2D 和 3D 视图中显示出来。同时,新建的物体也会自动被选中,其相关属性会在右侧的属性表中罗列出来。



3D 机房环境中，每个物体的主要属性如下图所示：



3D 机房环境还提供了不同角度的旋转、缩放等，图解如下：



在 3D 机房中，物体的呈现遵循以下原则：

- 每一个物体都是立方体。3D 机房不支持其他形状的物体，例如球体、圆柱体等；
- 每一个物体在水平、垂直两个方向上旋转角度不超过 90 度。也就是说，每个物体只能呈现其左、右、顶三个面；
- 每个物体必须接触在水平面上，而不能悬空；

3D 机房物体的主要属性和含义在下表中罗列：

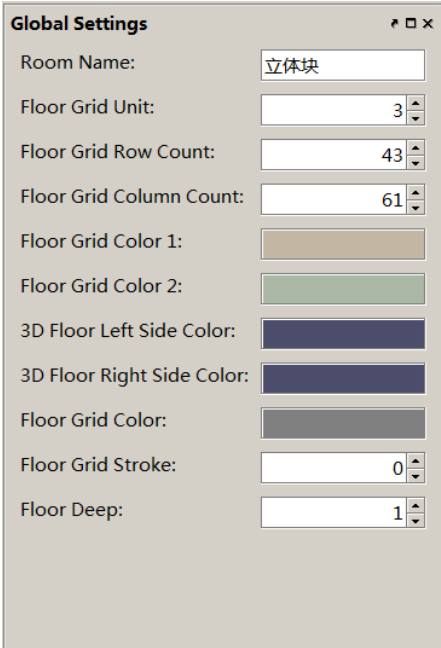
属性	描述	举例
AlarmState	告警状态，包含当前物体上的所有告警信息。	 带有告警信息的 3D 物体
Location	物体的坐标位置	
Height	物体的 3D 高度	 具有不同高度的物体

Left Visible	<p>物体的左侧立面否可见</p>	 <p>左侧立面不可见的物体</p>
Left Image	<p>物体的左侧立面贴图图片</p>	 <p>设置左侧立面贴图的物体</p>
Left Image Bounds	<p>物体的左侧立面贴图的具体边界</p>	 <p>左侧贴图设置边界为 [10,10,30,30] , 开始坐标为 10,10 , 宽高分别为 30 和 30</p>
Left Image Texture	<p>物体左侧立面贴图是否使用纹理模式</p>	 <p>设置金属拉丝纹理图片</p>
Left Image Texture Scale	<p>物体左侧立面贴图是否使用纹理的放大系数</p>	 <p>问题放大系数设置为 3 和</p>

		10 的区别
Right Visible	物体的右侧立面否可见	同上
Right Image	物体的右侧立面贴图图片	同上
Right Image Bounds	物体的右侧立面贴图的具体边界	同上
Right Image Texture	物体右侧立面贴图是否使用纹理模式	同上
Right Image Texture Scale	物体右侧立面贴图是否使用纹理的放大系数	同上
Top Image	物体顶面贴图图片	 <p>设置顶面图片的物体</p>
Top Visible	物体顶面是否可见	 <p>顶面不可见的物体</p>

设置机房环境参数

机房环境参数包括机房名称以及机房地平面参数。在左侧编辑面板上可以直接进行编辑：

A screenshot of the 'Global Settings' dialog box. It contains several configuration options for a 3D floor plan. The 'Room Name' is set to '立体块'. 'Floor Grid Unit' is 3, 'Floor Grid Row Count' is 43, and 'Floor Grid Column Count' is 61. There are color pickers for 'Floor Grid Color 1' (tan), 'Floor Grid Color 2' (green), '3D Floor Left Side Color' (dark blue), '3D Floor Right Side Color' (dark blue), and 'Floor Grid Color' (grey). 'Floor Grid Stroke' is 0 and 'Floor Deep' is 1. Each numeric field has up and down arrow buttons.

Room Name:	立体块
Floor Grid Unit:	3
Floor Grid Row Count:	43
Floor Grid Column Count:	61
Floor Grid Color 1:	
Floor Grid Color 2:	
3D Floor Left Side Color:	
3D Floor Right Side Color:	
Floor Grid Color:	
Floor Grid Stroke:	0
Floor Deep:	1

具体参数含义如下：

参数	描述
Room Name	机房名称
Floor Grid Unit	地平面网格单位。也就是地平面每个方块的尺寸有多少个最小单位数量。默认值是 3
Floor Grid Row Count	地平面方块行数量
Floor Grid Column Count	地平面方块列数量
Floor Grid Color 1	地平面方块颜色一
Floor Grid Color 2	地平面方块颜色二
3D Floor Left Side Color	地平面左侧立面颜色
3D Floor Right Side Color	地平面右侧立面颜色
Floor Grid Color	地平面网格线颜色
Floor Grid Stroke	地平面网格线宽度，默认为 0，单像素

Floor Deep

宽


地平面的厚度（高度），默认为 1 个单

位

常用编辑操作

编辑环境中，2D 视图和 3D 视图直接联动。2D 视图中可以控制尺寸、位置等操作，3D 视图中可以进行缩放、旋转、选择等操作。

2D 视图常用操作方法如下：

操作	描述
选择	<p>通过鼠标可以直接在 2D 视图上点击物体进行选择。被选中的物体会显示 8 个小方块。同时，属性表也会列出选中物体的所有属性。</p> 
修改尺寸	<p>选中物体后，通过鼠标对 8 个小方块可以直接对物体的尺寸进行修改。</p>  <p>此外，也可以用鼠标选中物体后，按 Shift+方向键来修改物体的尺寸。</p>
移动物体	<p>用鼠标选择物体后，对物体进行拖拽，可以显示出要移动的新位置；松开鼠标，可以将物体移动到虚线显示的位置。</p>  <p>此外，也可以用鼠标选中物体后，点击方向键对物体进行移动。</p>
放大视图	<p>点击工具栏的放大按钮，可以对整个视图进行放大。</p>

	
缩小视图	<p>点击工具栏的缩小按钮，可以对整个视图进行缩小。</p> 
缩放回退	<p>点击工具栏的 Zoom Back 按钮，可以回退到上次缩放比例。</p> 
充满缩放	<p>点击工具栏的 Zoom To Overview 按钮，可以将视图拉伸缩放充满当前整个可见区域。</p> 
矩形缩放	<p>点击工具栏的按钮，可以在视图上拖拽一个矩形区域，并将其放大至整个画布</p> 
缩放重置	<p>点击工具栏的“Zoom Reset”按钮，可以将当前视图缩放重置为 1:1。</p> 
鹰眼视图	<p>点击工具栏的“Overview Window”按钮，可以显示或隐藏鹰眼视图</p> 
全屏显示	<p>点击工具栏的“Full Screen”按钮，可以进入或退出全屏模式。</p> 
导出图片	<p>点击工具栏的“Export Image”按钮，可以把视图导出到位图图片中。</p> 

	<p>在弹出对话框中，可以指定图片导出的区域。</p> 
导出 SVG	<p>点击工具栏的“Export SVG”按钮，可以把视图导出到一个 SVG 文件中。SVG 是一个 W3C 标准的矢量图形文件格式。</p> 
拷贝粘贴	<p>选中物体后，按 Control+C 组合键进行物体拷贝，按 Control+V 组合键进行物体粘贴。</p>
删除对象	<p>拖动鼠标选择要删除的对象，点击“Del”键可以删除所选中的对象。在确认对话框中选择“是”。</p> 

3D 视图常用操作方法如下：

操作	描述
视图缩放	在 3D 视图上，用鼠标滑动滚轮，可以对视图进行缩放。鼠标指向的位置，会在缩放过程中，始终保持在鼠标的位置
视图平移	用鼠标在 3D 视图上进行拖拽，可以直接对视图进行上下左右平移。
物体选择	点击鼠标可以选择鼠标所在位置的物体。如果鼠标同时点多个物体，则被点中的物体都会被选中。
视图旋转	按住 Control 键的同时，用鼠标拖拽视图，可以对视图进行水平、垂

直两个方向旋转。

注意：水平旋转和垂直旋转最大角度为 90 度。

重置缩放

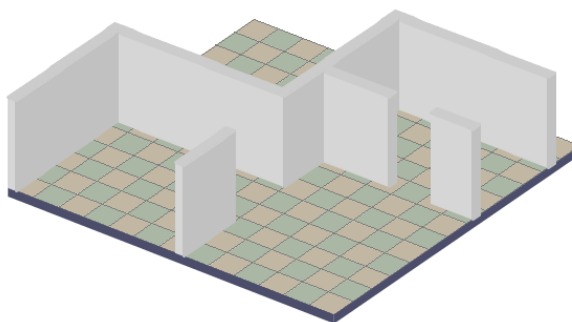
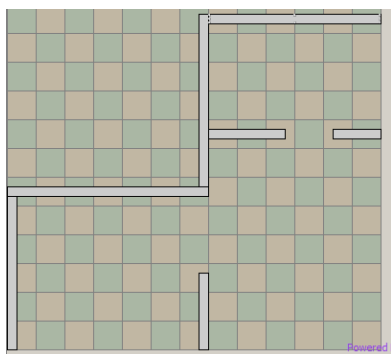
双击 3D 视图可以回到 1:1 的原始缩放比例。

一个完整的例子

本节通过编辑一个完整的例子来展示如何创建一个完整的 3D 机房。

第一步：创建房间墙体布局

点击 New 按钮创建节点，通过 Shift+箭头调整节点尺寸，形成一个单位的墙体厚度，并首尾相接，拼凑成如下形状布局。

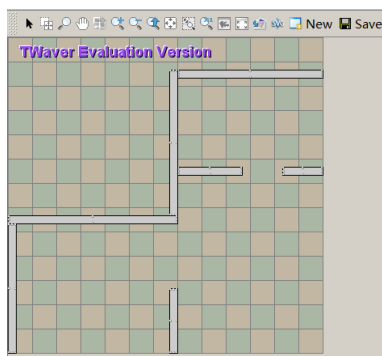


第二步：设置墙体纹理图片

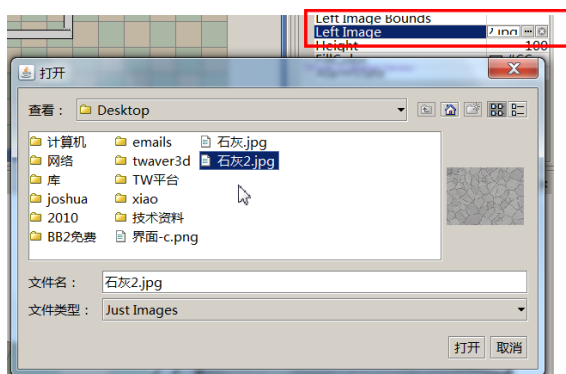
准备一张自己喜欢的墙体纹理作为素材，存放在磁盘上（如下图）。



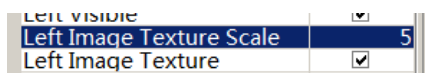
在 2D 视图上按键 Control+A 选择所有物体。



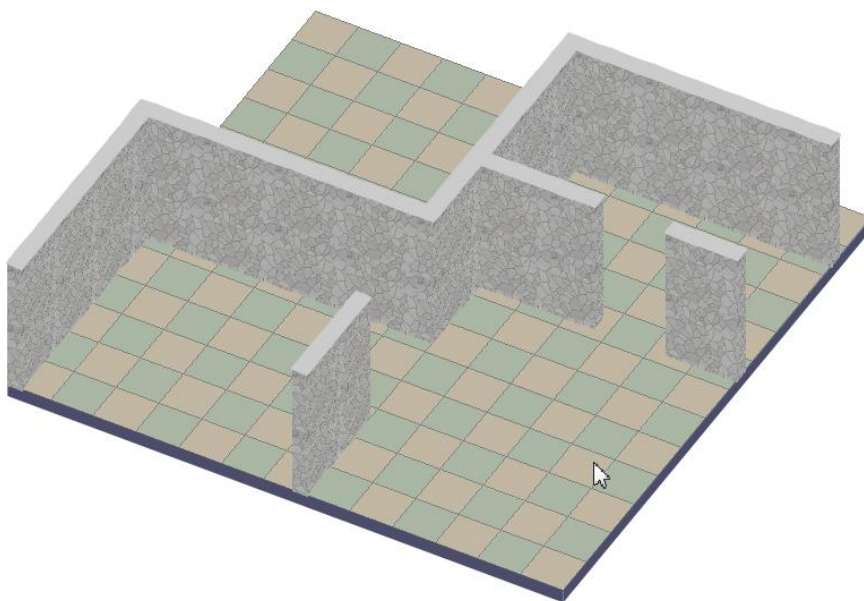
在右侧属性表中，设置其左侧立面的图片为该纹理图片。



同时将左侧里面设置为纹理模式，将纹理放大系数设置为 5。

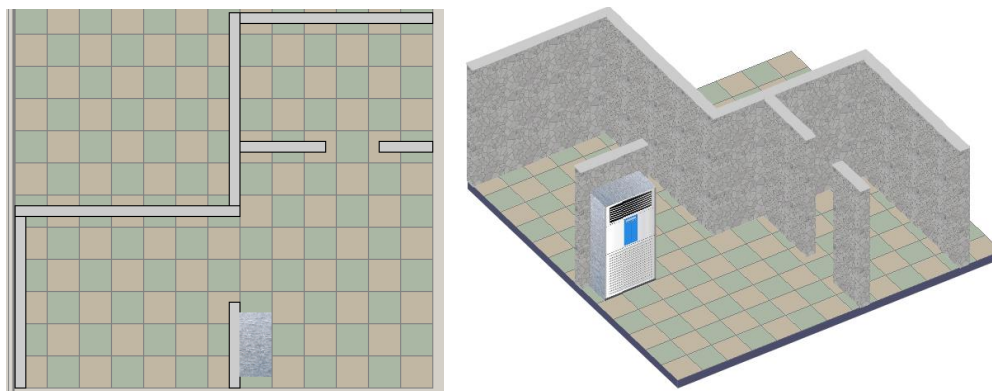


对右侧里面设置同样图片、纹理以及纹理放大视图。最终 3D 效果如下图：

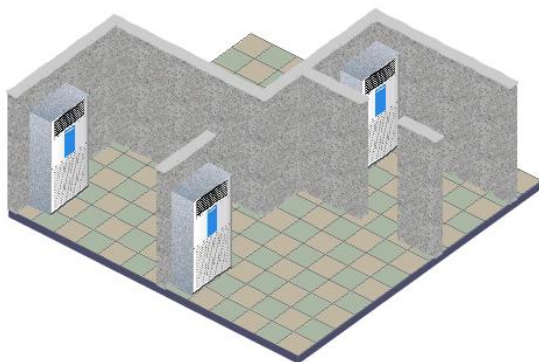


第三步：创建空调对象

选中所有墙体（Control+A）并在属性表中设置高度为 150。在底部墙位置放置一个新对象，高度 120 并设置其 top、left 立面设置金属拉丝纹理图片，右侧立面设置一个空调面板图片。显示效果如下图：

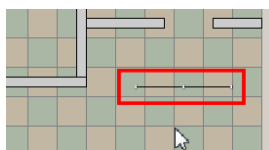


在 2D 视图上，选中空调对象，按 Control+C 后连续按 Control+V 快捷键，创建更多空调对象，并放置在合适的位置：

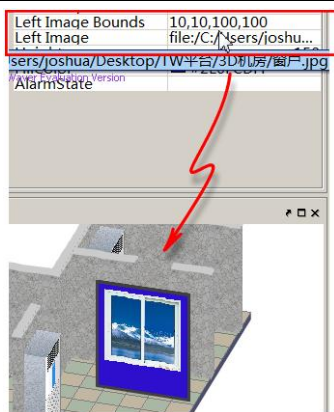


第四步：创建窗户和挂画

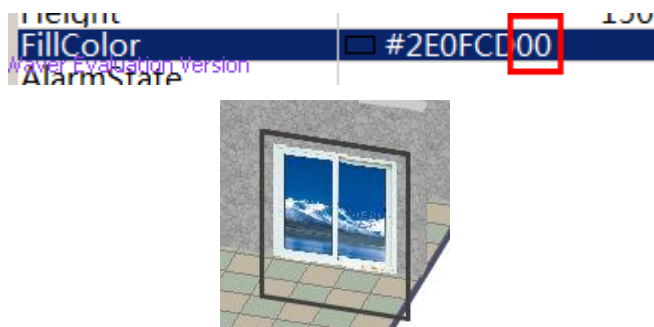
要在现有墙体上创建窗户、门、挂画等对象，可以创建一个物体并设置其厚度为 0，然后设置其侧面贴图，并强制指定贴图的边界区域，最后将对象靠在墙体表面的位置即可。



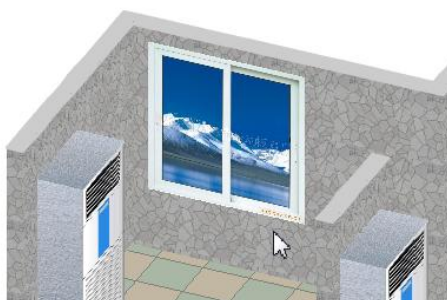
创建物体，调整厚度为 0



设置左侧图片，并指定其边界为（10,10,100,100）



设定物体填充色为透明（修改颜色最后两位数字为 00 即可）



移动物体到墙体位置

重复以上步骤，设置门、挂画等物体，进一步美化机房环境。

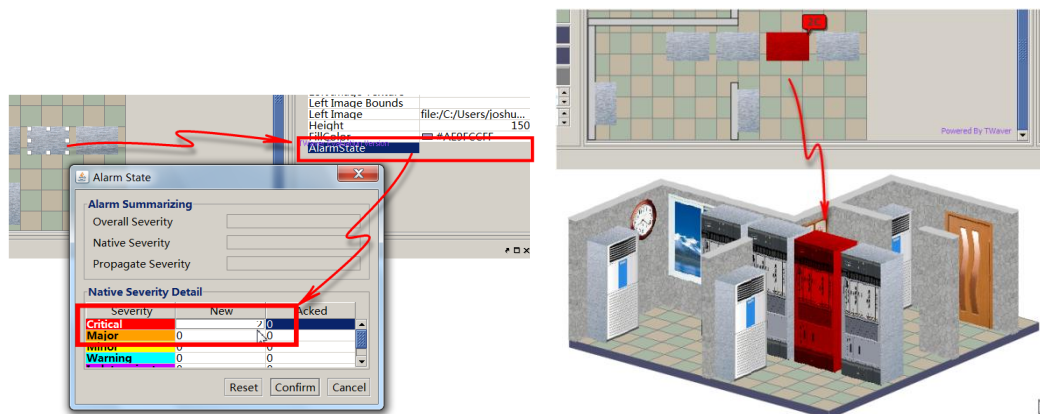


第五步：创建电信设备

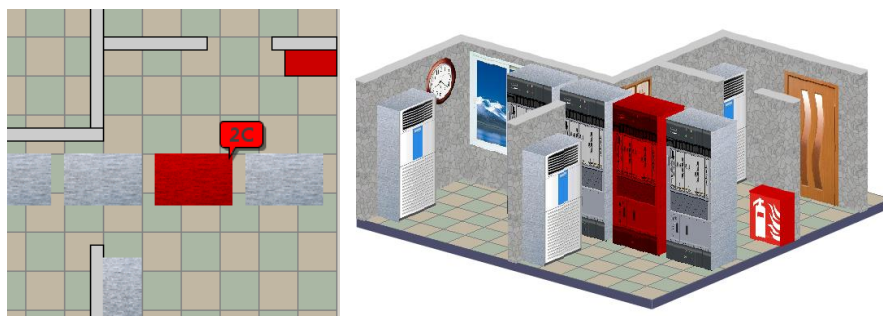
在房间空白处放置新物体，并设置其左右面板为设备面板图。创建更多设备并调整其位置。



可以在属性表中设置设备的告警状态。



添加墙角物体，设置填充色为暗红色，设置图片为灭火器消防栓。

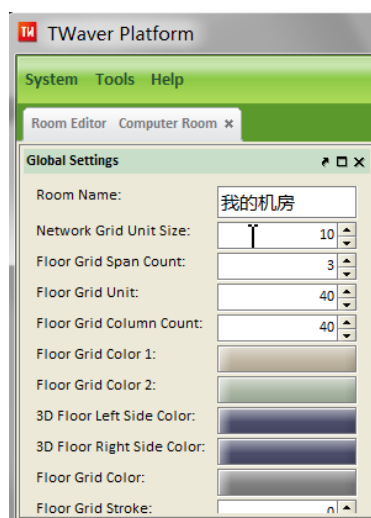


一个简单的机房环境就基本完成了。可以双击 2D 或 3D 视图的标题条进行最大化显示：

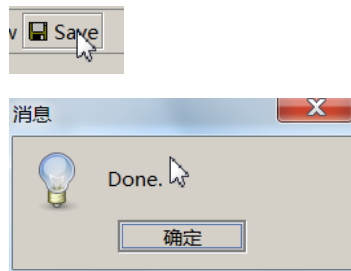


第六步：保存机房

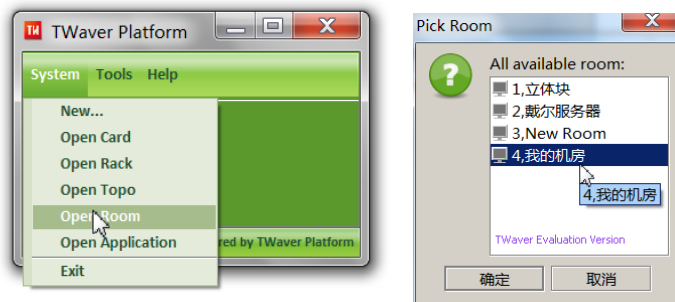
在机房环境变量编辑面板上设置机房名称等信息。



点击工具条上的 Save 按钮保存当前机房：



可以在主菜单中再次打开机房，进行进一步编辑修改。



通过以上步骤可以看到，TWaver3D 机房可以很方便的通过可视化的方式创建机房环境并用 3D 方式进行呈现。

开发指南

TWaver3D 机房不但可以单独作为工具使用，而且还提供了独立的组件，为开发者提供二次开发的能力。

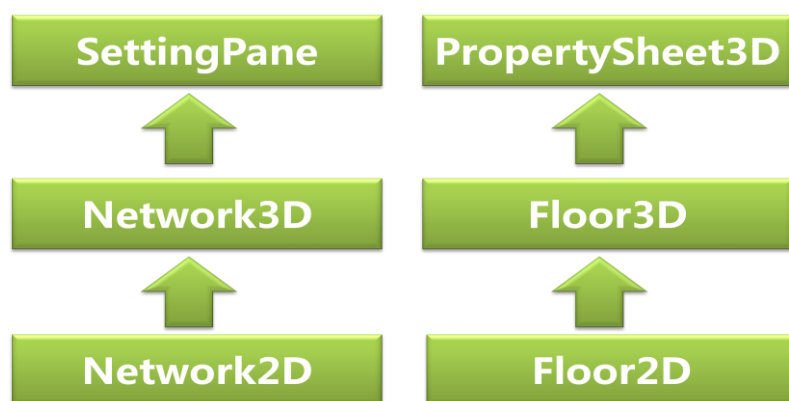
开发环境

使用 TWaver 机房组件进行二次开发，需要以下开发环境：

- JDK 6+ ；
- Java IDE (Netbeans 或 Eclipse) ；
- TWaver Java 3.5+ ；

类结构

TWaver 机房二次开发涉及到的主要类及其关系：



其中，各个类的主要用途如下：

类	用途
Network2D	是 TWaver Java 中 TNetwork 的一个扩展。作为 3D 视图的 2D 呈现、编辑和数据驱动。

Floor2D	和 Network2D 配合 ,用于绘制 2D 视图中的地板平面。
Network3D	3D 视图 ,以 3D 的方式呈现 2D 视图中的数据 ,并提供缩放、旋转等交互。Network3D 必须依赖于 Network2D 而存在。
Floor3D	和 Network3D 配合 ,用于在 3D 视图中绘制地板平面的 3D 视图。它依赖于 Floor2D 存在。
SittingPane	一个封装好的 3D 机房属性编辑面板。包括 3D 视图中的各种参数设置 ,可以用于需要数据编辑的场合中。对于只需要 3D 数据呈现的情况 ,可以不使用该类。
PropertySheet3D	一个封装好的 3D 数据属性表。当一个或多个 3D 数据被选中后 ,所有 3D 相关的属性会自动罗列在这个属性表中 ,方便查看和修改。该类可以用于需要数据编辑的场合中。对于只需要 3D 数据呈现的情况 ,可以不使用该类。

简单例子

下面的代码可以创建一个简单的 3D 机房视图。该程序只显示了一个 3D 视图 ,并显示了一个 3D 立方体物体。3D 视图可以通过鼠标的滚轮、拖拽等方式进行缩放、平移等操作。

代码如下 :

```
import java.awt.Color;
import javax.swing.JFrame;
import javax.swing.SwingUtilities;
import twaver.TDataBox;
import twaver.TWaverConst;
```



```
import twaver.TWaverUtil;

public class SimpleTest extends JFrame {

    private TDataBox box = new TDataBox();
    private Network2D network2d = new Network2D(box);
    private Floor2D roomFloor2D = new Floor2D(network2d);
    private Network3D view3d = new Network3D(network2d, roomFloor2D, 300, 100);

    public SimpleTest() {
        this.setTitle("TWaver 3D");
        this.setSize(900, 600);
        TWaverUtil.centerWindow(this);
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        view3d.setScale(1.5);

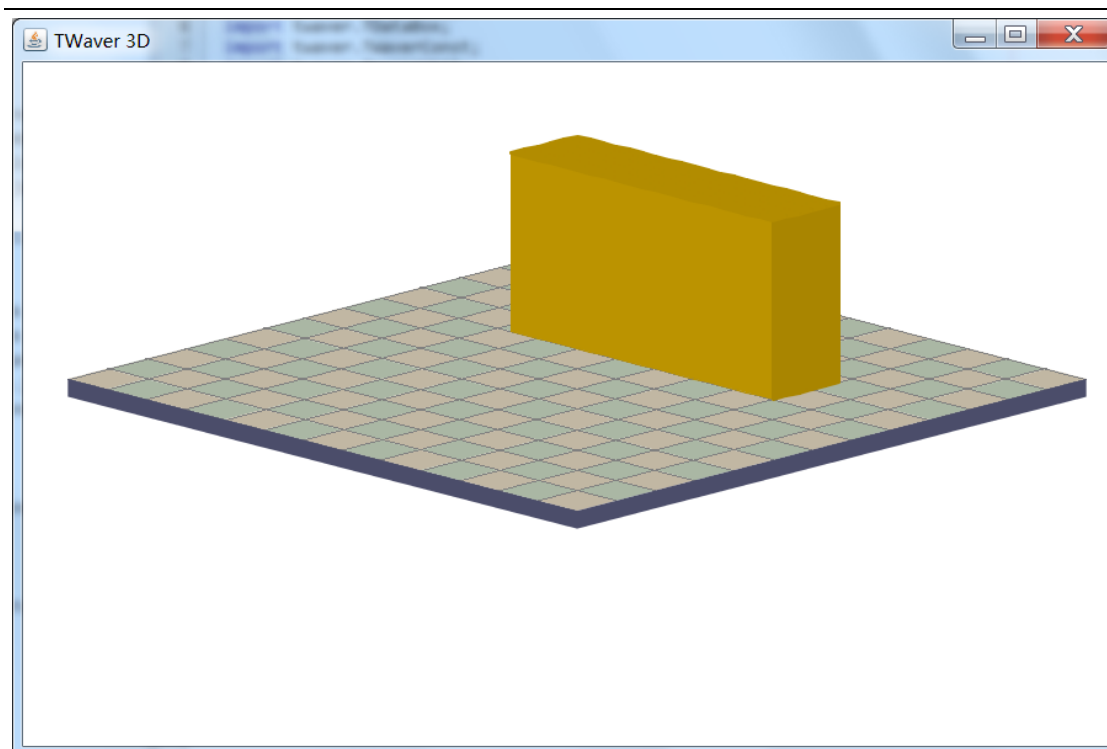
        this.add(view3d);

        //add a 3d object
        Element3D element = new Element3D();
        element.setLocation(100, 100);
        element.setSize(200, 50);
        element.setDeep(100);
        element.setFillColor(Color.orange.darker());
        network2d.getDataBox().addElement(element);
    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(new Runnable() {

            public void run() {
                TWaverUtil.setLocale(TWaverConst.EN_US);
                SimpleTest test = new SimpleTest();
                test.setVisible(true);
            }
        });
    }
}
```

运行该代码，效果如下图：



创建简单的编辑器

上述例子显示了一个非常简单的、只读的 3D 视图。本节将介绍如何创建一个简单的编辑器程序，可以对数据进行动态的添加、编辑等操作。其实在上述例子中已经创建了 Network2D 视图，只是没有显示出来而已。下面的例子将 Network2D 视图显示出来，并增加了 SettingPane 参数编辑视图、PropertySheet3D 属性表等组件，共同构成了简单的编辑器程序。

代码如下：

```
import java.awt.Color;
import java.awt.Insets;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.JButton;
import javax.swing.JComponent;
import javax.swing.JFrame;
import javax.swing.JSplitPane;
import javax.swing.SwingUtilities;
import twaver.TDataBox;
```

```

import twaver.TWaverConst;
import twaver.TWaverUtil;
import twaver.table.TPropertySheetPane;

public class Test extends JFrame {

    private TDataBox box = new TDataBox();
    private Network2D network2d = new Network2D(box);
    private Floor2D roomFloor2D = new Floor2D(network2d);
    private Network3D view3d = new Network3D(network2d, roomFloor2D, 300, 50);
    private SettingPane settingPane = new SettingPane(roomFloor2D, view3d, 2);
    private PropertySheet3D sheet = new PropertySheet3D(box);
    private TPropertySheetPane sheetPane = new TPropertySheetPane(sheet);

    public Test() {
        init();
    }

    private void init() {
        this.setTitle("TWaver 3D Room");
        this.setSize(1000, 700);
        TWaverUtil.centerWindow(this);
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        JSplitPane split1 = createSplit(sheetPane, network2d, JSplitPane.HORIZONTAL_SPLIT, 300);
        JSplitPane split2 = createSplit(settingPane, view3d, JSplitPane.HORIZONTAL_SPLIT, 350);
        JSplitPane split = createSplit(split1, split2, JSplitPane.VERTICAL_SPLIT, 350);

        this.add(split);

        JButton createNodeButton = new JButton("New");
        createNodeButton.setMargin(new Insets(0, 0, 0, 0));
        createNodeButton.addActionListener(new ActionListener() {

            public void actionPerformed(ActionEvent e) {
                Element3D element = new Element3D();
                element.setLocation(100, 100);
                element.setDeep(50);
                element.setFillColor(Color.green.darker());
                network2d.getDataBox().addElement(element);
            }
        });
        network2d.getToolBar().add(createNodeButton);
    }
}

```

```
private JSplitPane createSplit(JComponent one, JComponent two, int direction, int dividerLocation) {
    JSplitPane split = new JSplitPane(direction);

    split.setLeftComponent(one);
    split.setTopComponent(one);

    split.setRightComponent(two);
    split.setBottomComponent(two);

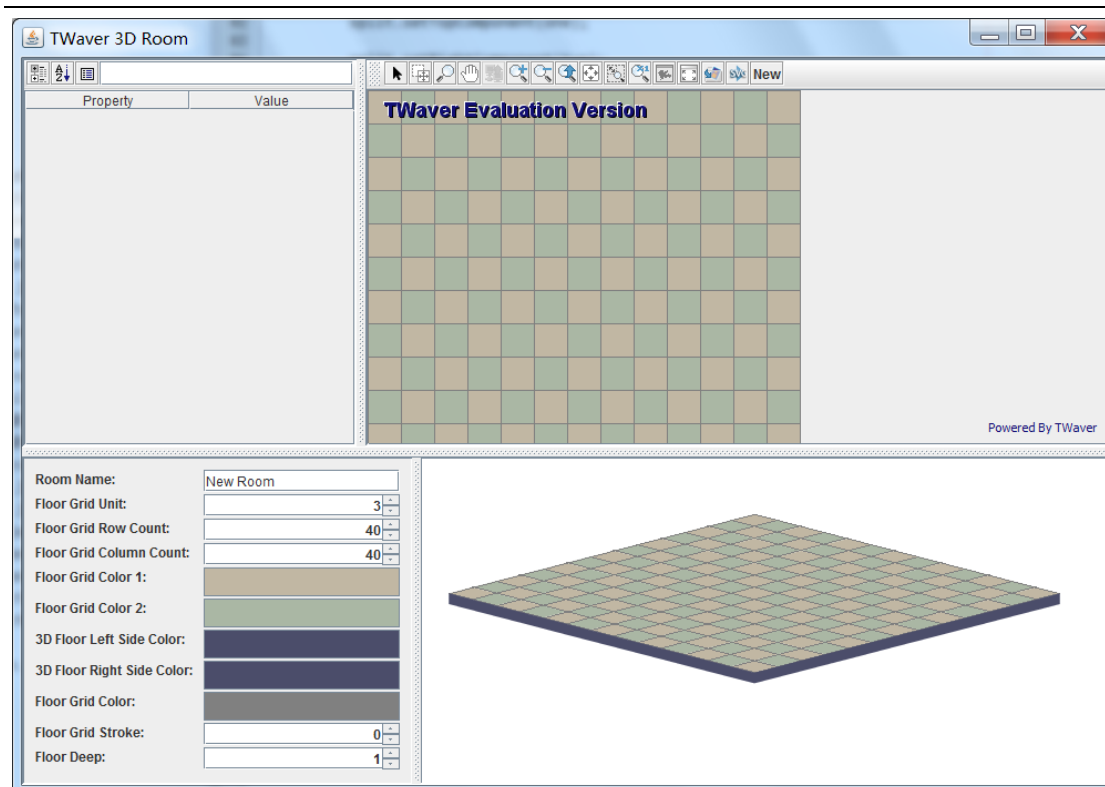
    split.setContinuousLayout(true);
    split.setDividerLocation(dividerLocation);

    return split;
}

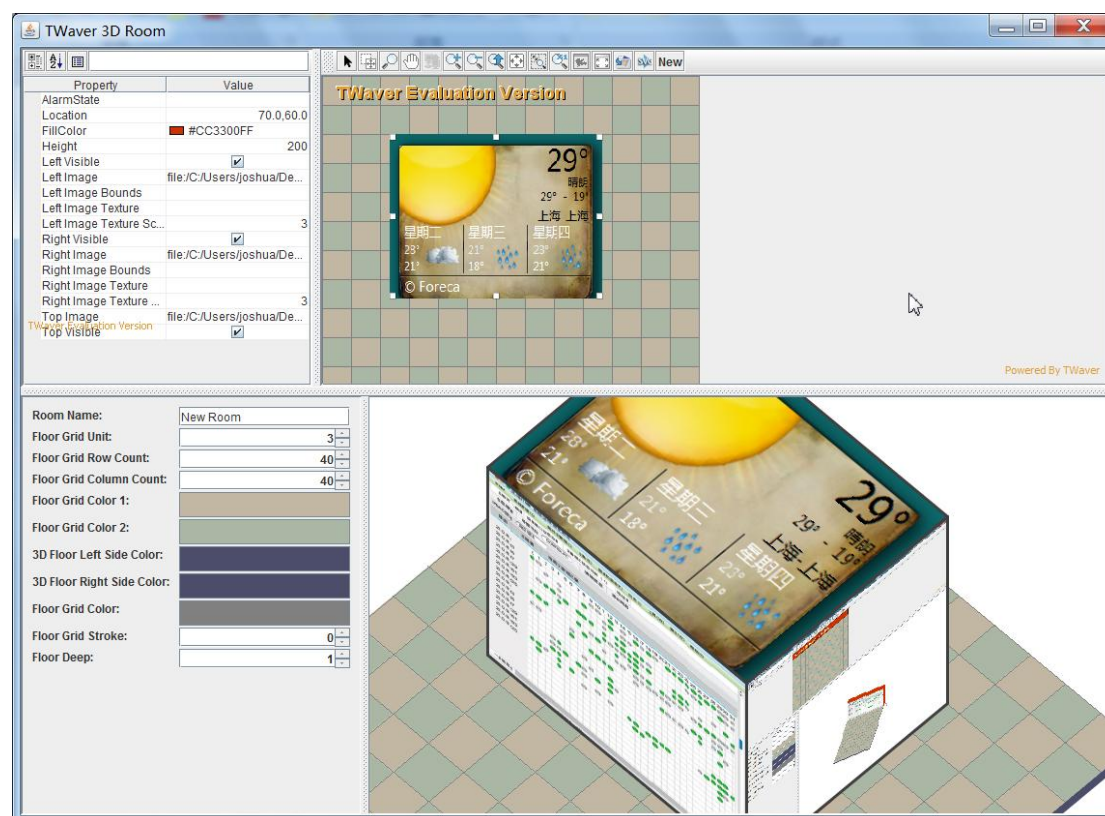
public static void main(String[] args) {
    SwingUtilities.invokeLater(new Runnable() {

        public void run() {
            TWaverUtil.setLocale(TWaverConst.EN_US);
            Test test = new Test();
            test.setVisible(true);
        }
    });
}
```

上述代码在 Network2D 的工具条上添加了一个按钮，点击按钮后，创建一个 Element3D 对象，并将其设置在面板上。程序运行效果如下图：



点击“new”按钮创建一个3D物体，并在左侧属性表中设置其上、左、右三面的图片贴图。效果如下：



保存至文件

要将编辑器中的数据保存至文件，可以在工具条上添加一个按钮，点击后将 DataBox 中的数据输出到 XML 字符串，然后将字符串保存至文件。

```
String xml = network3d.toXML(true);
JFileChooser chooser = new JFileChooser();
chooser.setFileFilter(filter);
int answer = chooser.showSaveDialog(EditorDemo.this);
if (answer == JFileChooser.APPROVE_OPTION) {
    String fileName = chooser.getSelectedFile().getAbsolutePath();
    if (fileName != null) {
        if (!fileName.toUpperCase().endsWith(".XML")) {
            fileName += ".xml";
        }
        OutputStream out = new FileOutputStream(fileName);
        out.write(xml.getBytes());
        out.close();
    }
}
```

从文件打开

对于保存的数据，可以写一个按钮，选择对应的文件，通过 DataBox 的 parse 方法进行解析，打开数据文件。

```
JFileChooser chooser = new JFileChooser();
chooser.setFileFilter(filter);
int answer = chooser.showOpenDialog(EditorDemo.this);
if (answer == JFileChooser.APPROVE_OPTION) {
    File file = chooser.getSelectedFile();
    if (file.exists()) {
        network2d.getDataBox().clear();
        network3d.clearObject3D();
        InputStream input = new FileInputStream(file);
        network2d.getDataBox().parse(input);
    }
}
```

Element3D 主要属性

通过以上例子可以发现，创建一个 3D 视图非常简单。在 3D 视图中创建一个 3D 数据的主要代码只有如下几行：

```
Element3D element = new Element3D();
element.setLocation(100, 100);
element.setDeep(50);
element.setFillColor(Color.green.darker());
network2d.getDataBox().addElement(element);
```

Element3D 实际上是一个普通的 TWaver Element 对象，它从 TWaver 预定义的 Element——ResizableNode 继承而来，并封装了 3D 物体的基本属性，例如左侧图片、右侧图片、顶部图片等等。通过设置和修改这些属性，即可实现对 3D 物体的外观定制。

Element3D 类的主要属性罗列如下。注意：所有 setLeft***均作用域左侧里面，而右侧立面均有对应的 setRight***方法，其功能是对右侧立面进行相应的功能设置。

- setFillColor：设置填充颜色；
- setDeep：设置 3D 物体的厚度（也就是三维中的高度）；
- setPaintLeft：是否左侧立面可见。当设置 false，整个左侧立面将消失；
- setLeftImageUrl：设置左侧立面的贴图。和 TWaver Java 中的惯例一样，给定图片的 URL 字符串即可。注意：默认的贴图方法，是将图片拉伸并铺满整个左侧立面。如需要边距或自定义边界，可以使用 setLeftImageBounds 方法进行设置；
- setLeftImageBounds：设置左侧立面的固定边界。该函数可以强制指定左侧立面图片的起始坐标位置和图片的宽高；

- `setLeftImageTexture` : 指定左侧立面贴图是否使用纹理模式。如果设置 `true` , 则贴图会被作为纹理进行全立面贴图。注意 : 一旦启用纹理 , `ImageBounds` 参数将不再起作用 ;
- `setLeftImageTextureScale` : 设置左侧立面纹理贴图的放大比例。默认值为 3。当该值越大 , 纹理的单位贴图尺寸也更大 ;
- `isPaintTop` : 是否顶面可见。当设置 `false` , 物体的顶面将会消失 ;
- `setTopImageUrl` : 物体的顶面贴图。注意 : 一旦设置顶面贴图 , 则 `Network2D` 中物体的图片将使用该顶面贴图进行显示。

Network3D 主要属性

`Network3D` 类是用来呈现 3D 场景的组件。注意它并非从 `TWaver Java` 的 `TNetwork` 或上面提到的 `TNetwork2D` 继承而来 , 而是直接从 `Swing` 的 `JPanel` 扩展而来。`Network3D` 必须依赖于 `Network2D` 而存在 , 从而从 `Network2D` 获得数据。此外 , `Network3D` 还必须设置一个 `Floor2D` 的类 , 用来绘制 3D 的场景地平面。`Network3D` 还提供了构造函数可以直接设置其 3D 坐标原点的屏幕坐标位置。

`Network3D` 的构造函数 :

```
//给定 Network2D 和 Floor2D 进行构造
public Network3D(Network2D network2d, Floor2D floor2d)
//给定 Network2D 和 Floor2D 进行构造 , 同时指定 3D 场景的坐标原点的位置。
public Network3D(Network2D network2d, Floor2D floor2d, int originX, int originY)
```

`Network3D` 的主要函数和用法如下 :

- `getNetwork2D` : 获得与 `Network3D` 绑定的 `Network2D` 对象。

- `setScale` : 设置 3D 场景的放大系数。设置 1 为原始比例。也可以通过调用 `zoomIn` 或 `zoomOut` 函数进行单次的缩放。
- `zoomIn/zoomOut` : 放大或缩小。每次缩放会用当前缩放因数乘以 (放大) 或除以 (缩小) `scale step` 数值 (可以通过 `getScaleStep` 函数获得, 其默认值是 1.2)。
- `setScaleStep` : 设置单次缩放变化因数。
- `setOrigin` : 设置 3D 坐标原点的 x 和 y 数值。
- `setHorizontalAngle` : 设置 3D 场景水平旋转角度。必须为 0-90 度。
- `setVerticalAngle` : 设置 3D 场景垂直旋转角度。必须为 0-90 度。
- `getObjectAt` : 获得给定坐标的 3D 物体。如果多个物体存在该点, 则返回所有物体。
- `toXML` : 将 3D 场景所有数据输出到 XML 中。

数据的输入与输出

默认输出方法

在 3D 视图程序中, 3D 机房数据使用和 TWaver Java 提供的同样的对象 XML 序列化方法进行数据的序列化和反序列化, 从而实现数据的输入与输出。通过 `TDataBox.output()` 方法和 `TDataBox.parse()` 方法完成数据的导出和输入。不过, 为了方便开发者使用, `Network3D` 类提供了 `toXML` 函数可以直接输出 XML 字符串:

```
String xml = this.network3d.toXML(true);
```

其中的 `boolean` 值是用来控制是否对各立面贴图进行 BASE64 编码输出。

自定义格式输出

如果要进行进一步细致的输出参数控制，也可以不适用 toXML 函数，而直接使用 TDataBox 的 output 方法进行。以下代码显示了如何将内存中的 3D 数据导出到 XML 格式的字符串中并进行输出，同时对输出的各种参数进行控制，例如是否输出 ID、是否输出告警信息等：

```
ByteArrayOutputStream out = new ByteArrayOutputStream();
setting.setWithElementId(false);
setting.setWithUserProperty(true);
setting.setOutputStream(out);
box.output(setting);
String xml = out.toString("utf8");
```

以上代码可以对数据进行 XML 序列化输出。

为何使用 BASE64 图片编码输出

3D 物体中的各个立面贴图将以 URL 字符串的方式输出到 XML 中。当 XML 被传输到其他机器或应用环境中，如果图片的 URL 资源位置发生变化，则无法读到这些图片。解决这一问题的一个方法是：在数据输出过程中，使用 BASE64 编码对所有图片资源进行编码并序列化输出。这样，XML 中就包含了所有的图片资源，无论 XML 被传输到哪里，都可以无外部依赖的打开。

使用 Network3D 的 toXML 函数可以直接进行图片的 BASE64 编码输出。实际上，该函数也是使用了 TDataBox 的输出参数控制机制，对用到的图片进行遍历和编码设置。以下代码展示了其具体实现方法：

```
//use base64 image encoding output setting.
DataBoxOutputSetting setting = new DataBoxOutputSetting();
Map images = new HashMap();
Iterator it = this.view2d.getDataBox().iterator();
while (it.hasNext()) {
```

```
Element element = (Element) it.next();
if (element instanceof RoomNode) {
    RoomNode roomNode = (RoomNode) element;
    this.addImage(images, roomNode.getLeftImageUrl());
    this.addImage(images, roomNode.getRightImageUrl());
    this.addImage(images, roomNode.getTopImageUrl());
}
}
setting.setImages(images);
```

经过以上代码的预处理之后，再使用这个 setting 对象对 box 进行序列化，即可实现 BASE64 的图片编码存储。

更多信息

技术支持

如果您需要产品技术支持与帮助，您可以：

- 拨打电话到：(86) 021-64398788 转 TWaver 技术支持；
- 发送传真到：(86) 021-64395374；
- 发送电子邮件到：info@servasoftware.com；
- 加入 TWaver MSN 在线技术支持讨论组：twavercn@hotmail.com
- 浏览官方网站 <http://www.servasoftware.com>
- TWaver 中文社区：twaver.servasoft.com
- TWaver 中文论坛：twaver.servasoft.com/forum

Serva Software 技术支持小组将为您提供专业、细致的技术服务。

关于赛瓦软件

Serva Software(赛瓦软件)隶属于 Serva Group(赛瓦集团)。Serva Group 于 1977 年成立于美国 Texas 州 Wichita Falls。Serva Software 致力于通信行业软件和制造业信息化软件的研究和开发，为电信运营商、设备制造商、独立软件提供商和各类制造企业提供解决方案和服务。Serva Software 是国际电信管理论坛(TMF)成员、Sun 公司 Java 合作伙伴、DNV ISO 9001 认证企业。

赛瓦软件(上海)有限公司是 Serva Software 于 2004 年在中国成立的全资子公司，软件认证企业、高新技术企业，主要面向中国及整个亚太地区提供产品技术支持与服务。