

Smart Knight

My Articles: From. 2009-3 To. 2012-3 <http://blog.csdn.net/watson243671>

导航

- 首页
- 新随笔
- 联系
- 聚合XML
- 管理

搜索

搜索

最新评论XML

1. re: Eclipse中设置作者日期等信息
非常有用,推荐

--测试用户
2. re: Eclipse中设置作者日期等信息
测试

--测试用户
3. re: 用Java求两时间点之间日期差的简洁方法
试了试, 大牛的是正确的

--打算打
4. re: 30 IMP-00019: row rejected due to ORACLE error 12899
转
qq 632618767,有问题请教 麻烦 加下

--魏云龙
5. re: maximo 更改数据库[未登录]
这个工具怎么用啊, maximo初学者, 谢谢了

--li

MAXIMO入门培训

MAXIMO入门培训

目录

MAXIMO入门培训... 1

1.MAXIMO使用技术介绍... 2

2.MAXIMO程序结构... 3

MAXIMO程序简介... 3

3.MAXIMO页面开发... 7

界面示例... 8

常用控件... 8

4.MAXIMO工作流开发... 9

MAXIMO工作流模型... 9

工作流涉及的应用程序... 9

工作流组件、标记... 10

工作流实践... 11

5.MAXIMO后台任务开发... 13

界面配置... 13

代码开发... 13

6.MAXIMO数据库配置... 14

界面配置... 14

更新数据库... 14

7.MAXIMO代码开发... 16

MboSet 例子... 16

Mbo例子... 17

FldXX例子... 18

8.Mbo中常用类及说明... 19

add(). 20

appValidate (). 20

canDelete(). 20

checkFieldAccess(). 20

clear(). 20

copy(). 20

delete(). 21

getXXXX(attributeName). 21

getInitialBaseLanguageString(). 21

getInsertCompanySetId(). 21

getThisMboSet(). 21

init(). 21

initialize(). 21

isModified(). 22

isModified(attribute). 22

isNew(). 22

isNull(attributeName). 22

isSelected(). 22

modify(). 23

rollbackToCheckpoint(). 23

setFieldFlag(attributeName, flag, state). 23

setFlag(flag, state). 23

setValue(name, attributeName). 23

setValueNull(attributeName). 24

toBeAdded(). 24

isNew(). 24

toBeUpdated(). 24

toBeDeleted (). 24

toBeSaved (). 25

9.MboSet中常用类及说明... 25

save(). 25

cleanup(). 25

close(). 25

commit(). 25

1.MAXIMO使用技术介绍

RMI

Maximo基于J2EE架构，其中的业务对象都采用RMI的机制实现，这种机制给分布计算的系统设计、编程都带来了极大的方便。

Ajax

Maximo中的逻辑表现和用户交互采用Ajax技术，实现流畅的用户体验。

Semi-Automatic Object/Relationship Mapping

使用半自动对象和关系映射机制提高开发效率和方便开发的维护。

XML

基于xml技术构架系统，Maximo中的应用页面和信息交互都采用xml格式。

事务处理

MAXIMO框架提供自动的事务处理机制。如果需要长事务，需要将相关方法纳入到一个事务队列中。

OR持久化

MAXIMO提供系统级别的OR映射和持久化化机制。我们对持久层的操作只需要继承MAXIMO提供的基类。

缓存机制

MAXIMO提供系统级别的缓存机制，并且缓存会在数据发生变化时自动重新加载。

日志跟踪和异常处理

MAXIMO使用并扩充了Log4j，提供了自己的根，可以根据具体的所属制定合适的日志处理方案。MAXIMO中对异常进行了包装，提供了友好的异常处理机制。

2.MAXIMO程序结构

MAXIMO程序简介

businessobjects.jar (maximo业务相关类)

Lib (类库资源)

maximouiweb.war (webclient前台应用)

mboejb.jar (maximo集成相关的包)

mbojava.jar (maximo一些自测试的包)

mboweb.war (Maximo的启动加载的web应用)

meajmsejb.jar (maximo集成相关的包)

meaweb.war (集成相关的web应用)

META-INF (相关描述文件)

properties.jar (maximo的配置文件)

resources (maximo相关资源)

表现层

MAXIMO的页面除了用户登录验证的过程外，其余的页面内容均由XML进行提供。这些XML存储在MAXPRESENTATION表中。XML文件由JSP的控制标签组件组成。其中components组件和Javascript中的脚本共同生成需要显示的页面，CSS文件夹定义了通用样式。

控制层

mboweb的部署描述符

```
<servlet>
```

```
<description>Server used to load MAXIMO Business Objects environ-
ment</description>
```

```
<display-name>MAXIMO Startup Servlet</display-name>
```

```
<servlet-name>MAXIMOSTartupServlet</servlet-name>
```

```
<servlet-class>psdi.servlet.MAXIMOSTartupServlet</servlet-class>
```

```
<load-on-startup>2</load-on-startup>
```

```
</servlet>
```

MAXIMOSTartupServlet的主要功能是读取MAXIMO的配置文件maximo.properties, 并将其以Properties对象的形式传递给MXServer的start方法。MAXIMOSTartupServlet调用MXServer.start(properties)主要是建立MXServer的实例, 并初始化与其相关的信息。

MXServer是MAXIMO的系统服务和应用服务的运行期间的上下文。

A. 系统服务主要包括数据连接池的建立和管理、对MAXIMO缓存的实现、调用MaximoDD从数据库读取并加载所有数据字典信息、实例化系统服务和应用服务的对象、实现所有远程应用服务的灵活定位、注册RMI、提供MaxSession的线程管理等。

B. 应用服务主要是与业务相关的MBO.

程序主要包含在:

psdi.server.*: 定义了和maximo系统有关的类, 如数据库的连接管理, session的管理, 线程的管理, 数据对象内容的加载, 实例化系统服务和应用服务的对象, mbo和mboset的访问等等。

Maximouiweb的部署描述符

```
<servlet>
```

```
<servlet-name>webclient</servlet-name>
```

```
<servlet-class>psdi.webclient.servlet.WebClientServlet</servlet-
class>
```

```
<init-param>
```

```
<!-- The character encoding the servlet will use for all http requests
and
```

```
request responses. -->
```

```
<param-name>char_encoding</param-name>
```

```
<param-value>UTF-8</param-value>
```

```
</init-param>
```

```
</servlet>
```

```
<servlet-mapping>
```

```
<servlet-name>webclient</servlet-name>
```

```
<url-pattern>/ui/*</url-pattern>
```

```
</servlet-mapping>
```

WebClientServlet是其最核心的一个servlet, 所有和/ui/*匹配的URL都会映射到这个servlet,该servlet负责定义页面的内容类型, 并调用WebClientSession进行相关处理。

业务层与持久层

Bussinessobjects的目录结构。Bussinessobjects主要包括系统级服务和业务级服务两个方面的内容。

psdi.app.*: 定义了业务对象Mbo、MboSet（一组Mbo的集合，如一个列表），主要是描述和应用程序有关的所有业务逻辑。

psdi.server.*: 定义了和maximo系统有关的类，如数据库的连接管理，session的管理，线程的管理，数据对象内容的加载，实例化系统服务和应用服务的对象，mbo和mboset的访问等等。

psdi.util.*: 定义了maximo很多有用的类，如克隆app，删除app，httpsession的管理，加密解密，客户化应用生成等等。

psdi.workflow.*: 定义了工作流所需要的逻辑处理。

psdi.mbo.*: 定义了maximo核心的持久层的操作,实现ORMaping.

3.MAXIMO页面开发

“应用程序设计器”简单易用，技术用户和非技术用户均无需编辑任何代码即可配置 Maximo 应用程序。“应用程序设计器”具有拖放功能，进一步简化了编辑应用程序的过程。

可以使用“应用程序设计器”来完成一些常见修改操作，包括移动字段和部分、创建新字段、表和标签页。其他高级功能有创建或复制应用程序、定义签名选项、编辑工具栏和“选择操作”菜单。

界面示例

常用控件

4.MAXIMO工作流开发

您可以使用“工作流设计器”应用程序为记录创建一系列路径，以完成整个过程。工作流过程可被认为是一幅图，用于交互指引记录或用户执行一系列步骤。在“工作流设计器”的图形界面中，展示了记录可能通过的路径。

MAXIMO工作流模型

工作流定义——过程模型

活动

转换条件

相关数据

角色

需要激活的内部/外部程序

工作流涉及的应用程序

Actions 操作

Communication Templates 通信模板

Escalations 上报、自动触发

Inbox/Assignments Setup 收件箱设置

People 人员

Person Groups 人员组

Roles 角色

Workflow Application 工作流应用

Workflow Designer 工作流设计器

Workflow Inbox 工作流收件箱

工作流组件、标记

“开始”节点表示“工作流”过程的开始。当创建新过程时，“工作流”会在画布上放置一个“开始”节点。任何过程都只能有一个起点。

“停止”节点标志着工作流结束，并且记录脱离工作流控制的点。当创建新过程时，“工作流”会在画布上放置一个“停止”节点。您可以在画布上根据需要放置其他“停止”节点。

“任务”节点使用户可以指引记录的路径。您必须至少有一个连接从“任务”节点引出来。当您的业务规则需要对邮箱任务分配做出肯定或否定的用户响应时，请使用“任务”节点。

“条件”节点会根据记录中信息自动指向记录。从“条件”节点出来的连接，必须有一个正向连接和一个反向连接。当记录退出“条件”节点时，所用连接由节点内的 SQL 表达式表示，该表达式能够解析真（正连接）或假（负连接）。

“手动输入”节点允许用户指引记录的路径。当您希望用户从菜单中选择下一步时，请使用“手动输入”节点。

“子过程”节点表示嵌入在另一个“工作流”过程中的完整“工作流”过程。除了正向线之外，“子过程”还可以引出一个反向线。当“子过程”遇到停止节点时，它会沿结束之前走过的那条路线返回到主过程。这样，“子过程”可以将造成终止的逻辑原因带回主过程。

“交互”节点必须有一个或多个进入该节点的操作线，但只能有一个退出该节点的操作线。通过提供定义好的选项，您可以使用交互节点将终端用户引导至特定问题的解决路径。这些定义好的选项可以指导用户走过指定好的路径，并可在任何给定的会话中管理与记录之间的关系。“手动输入”节点通常用在交互节点之前。

您可以使用 **“等待”**节点来创建对某个操作的特定反应。您可以在“等待”节点中定义任何操作来触发指定的反应。当“工作流”在活动过程中遇到“等待”节点时，过程将一直在该节点处暂停，直到发生任何指定的事件。当指定事件发生时，该事件将通知节点，同时过程将从单一退出点退出该节点，继续执行。

工作流实践

利用Maximo的demo库中的工作流实例进行相关实践，查看过程APPWFWO，修改此过程，并在工单中测试该流程。

任务节点：

标题：是显示在图标下的文字；

描述：是对任务节点的说明；

应用程序：是该流程应用的应用程序；

时限：是规定该任务需完成的时间长度，超过该时限，则该任务回分配给“升级角色”。

任务分配：为该任务分配角色；

为角色指定表达式：角色下的表达式可以限制只有在满足该表达式的情况下，任务才会分配到该角色，即同一个任务可以表达式来确定分配对象。该选项对控制转发很有用。

执行接受操作，两个选项：接受任务的所有人都接受才算任务接受和只要任何一个人接受就算该任务接受。

任务节点可以最多有两个输出操作：一个正向操作和一个反向操作。正向操作对应接受该任务，反向操作对应拒绝该任务。

任务节点的输入操作可以有多个。

5.MAXIMO后台任务开发

界面配置

新建后台任务：

配置相关参数：

代码开发

代码如下：

```
public class CrontaskTest extends SimpleCronTask{  
    public void init() throws MXException{  
        super.init();  
        //参数初始化  
    }  
    public void cronAction() {  
        //执行相应逻辑  
    }  
}
```

6.MAXIMO数据库配置

界面配置

包括新建对象、创建对象属性、索引和关系

更新数据库

第一步，打开管理方式

第二步，应用配置变更，开始配置数据库

7.MAXIMO代码开发

MboSet 例子

MboSet是Maximo中持久层的核心类，其核心的超类psdi.mbo.MboSet。

应用程序对象与MboSet类直接关联，在创建对象时必须指定与对象关联的MboSet类。

MboSet类必须继承psdi.mbo.MboSet类，实现psdi.mbo.MboSetRemote接口；

(1) 必须定义构造方法；

(2) 实现getMboInstance(MboSet arg0)方法，在该方法中返回Mbo类的实例

```
package maxdemo.lesson1;

import java.rmi.RemoteException;

import psdi.mbo.Mbo;

import psdi.mbo.MboServerInterface;

import psdi.mbo.MboSet;

import psdi.util.MXException;

public class CustDemoMboSet extends MboSet implements CustDemoMboSetRemote{

    public CustDemoMboSet(MboServerInterface arg0) throws RemoteException {

        super(arg0);

    }

    private static final long serialVersionUID = 3918526761961603733L;

    protected Mbo getMboInstance(MboSet arg0) throws MXException, RemoteException {

        return new CustDemoMbo(arg0);

    }

}
```

Mbo例子

定义：Mbo是Maximo中持久层的核心类，用来处理与数据库的所有操作，例如：新增、修改、删除等。其核心的超类是psdi.mbo.Mbo。

```
package maxdemo.lesson1;

import java.rmi.RemoteException;

import psdi.mbo.Mbo;

import psdi.mbo.MboSet;

import psdi.util.MXException;

public class CustDemoMbo extends Mbo implements CustDemoMboRemote{
```



```

public CustDemoMbo(MboSet arg0) throws RemoteException {
    super(arg0);
}

public void add() throws MXException, RemoteException {
    super.add();
    //新增逻辑
}

public void delete() throws MXException, RemoteException {
    super.delete();
    //删除逻辑
}

public void modify() throws MXException, RemoteException {
    super.modify();
    //修改逻辑
}

private static final long serialVersionUID = 9053122793999687843L;
}

```

FldXX例子

FldXXXXXX 类是对字段的填写进行增强的类，其功能如下：1) 构造函数 FldXXXXX(MboValue mbovalue)通过传入的mbovalue的属性名构造系统查询语句对查找进行支持。2) action()和vaildate () 方法在查找带值返回后根据业务逻辑的需要进行后续动作。其中validate 在点击带值返回时执行而action方法在记录保存时执行。

```

package maxdemo.lesson1;

import java.rmi.RemoteException;

import psdi.mbo.MAXTableDomain;

import psdi.mbo.MboSetRemote;

import psdi.mbo.MboValue;

import psdi.util.MXException;

public class FldDemoMonth extends MAXTableDomain{

    public FldDemoMonth(MboValue arg0) {
        super(arg0);
    }

    //一个示例

    String thisAttr = getMboValue().getAttributeName();

    setRelationship("LOCATIONS", "LOCATION=" + thisAttr);

    setListCriteria("1=1");

    String[] targetKeys = { thisAttr };

```

```
String[] sourceKeys = { "LOCATION" };

setLookupKeyMapInOrder(targetKeys, sourceKeys);

}

public MboSetRemote getList() throws RemoteException, MXException {

return super.getList();

}

public void action() throws MXException, RemoteException {

super.action();

//字段动作逻辑

}

public void validate()

throws MXException, RemoteException{

super.validate()

//字段验证逻辑

}

public void init() throws MXException, RemoteException {

super.init();

//字段始化逻辑

}

}
```

8.Mbo中常用类及说明

Mbo是maximo的业务对象，它是一个没有默认构造函数的类，所以我们需要在实现的XXXMbo中写自己的以MboSet为参的构造函数，他们的类型要是相对应的。在相对应的MboSet中，XXXMboSet需要返回一个此处的相对应的Mbo对象。

add()

说明：该方法是在要把一个新的Mbo加到MboSet的集合里的时候调用。在实际使用的时候可以在上面做字段值初始化和只读属性控制等的操作。

现在这个方法里面没有实现任何内容，也就是说默认的情况下，它不做任何操作，可以根据实际情况复写这个方法做相应操作。

appValidate ()

说明：该方法是保存之前验证的方法，可以根据自己的验证规则复写这个方法。

使用方法：保存之前自动调用此方法验证，可以复写此方法来完成自己的规则，

canDelete()

说明：判断对象是否可以被删除，这个方法也没有实现任何内容，必须要复写，因为该方法没有返回值，所以如果不能删除的话必须在所写的方法里抛出异常。

使用方法：该方法是在删除操作(delete)之前自动调用

checkFieldAccess()

说明：该方法为检查整个对象的是否只读属性，如果不允许访问某个字段，则抛出MXAccessException提示用户。

使用方法：直接调用此方法即可，如：this. checkFieldAccess()。

clear()

说明：清空所有涉及到的Mbo。

使用方法：直接调用此方法即可，如：this. clear ()。

copy()

说明：复制方法，在MboSet里新增一个Mbo,并把当前Mbo的值复制给新建的Mbo。

delete()

说明：把一个对象通过普通的安全检查后删除它，只有在一个对象没有被标识为“NODELETE”，属于当前对象的集合没有被标识为“NODELETE”，而且必须通过canDelete方法里的所有检查才能被删除。

使用方法：直接调用此方法就可以了。

getXXXX(attributeName)

说明：在当前对象中得到由属性name标识的字段值，其中“XXXX”为java的基本数据类型。attributeName为字段的属性名，如果不是当前对象则要包含“对象名.”。

getInitialBaseLanguageString()

说明：得到首选的基本语言字符串，如果没有指定，则返回一个空字符串。

getInsertCompanySetId()

说明：得到默认的company的ID，以便在一个新的mbo中使用。

getThisMboSet()

说明：得到拥有此Mbo的MboSet。

init()

说明：当初初始化MboValue化后构造Mbo的时候调用此方法，可以复写此方法，完成用户个性化的初始化操作。

使用方法：实际中可以复写此方法

initialize()

说明：初始化Mbo，触发init事件，调用可被复写的init()方法。

使用方法：实际中它是调用了init()方法。

isModified()

说明：数据是否修改过？只要有一个或者一个以上的属性值更改过，就那么这个对象就是更改过的。如果更改过，返回true，否则返回false。

使用方法：直接调用此方法，例如：this.isModified()。

isModified(attribute)

说明：判断指定属性的字段值是否更改过，只有当数据通过setValue方法改变的时候，那么这个字段的值就是改变过的，如果更改过，返回true，否则返回false。参数attribute为要判断的字段属性名。

使用方法：直接调用此方法，例如：this.isModified("name")。

isNew()

说明：判断Mbo是否是新建的，如果是新建则返回true，否则返回false。

使用方法：直接调用此方法(this.isNew())，一种典型的使用方法是在标记删除的时候判断，若是新建的，直接删除，否则只是做标记。

isNull(attributeName)

说明：判断指定属性名的字段值是否为null，如果为null则返回true，否则返回false。其中attributeName为要判断的属性的字段名，如果要判断的属性不是当前对象的属性，则要在属性名上加“对象名.”。

使用方法：直接调用此方法：boolean isnull = this.isNull ("name")。

isSelected()

说明：判断当前Mbo是否已经标记为选中，如果是，返回true，否则返回false。

使用方法：

modify()

说明：当Mbo的属性值改变的时候调用的方法，默认的时候不做任何事情，我们可以复写此方法来完成一些特殊的操作，如改变更改日期，更改人等，如果操作过程中程序抛出MXException，则改变的记录回滚。

使用方法：复写此方法，完成需要的操作。

rollbackToCheckpoint()

说明：回滚到上一个还原点。

使用方法：

setFieldFlag(attributeName, flag, state)

说明：设置当前Mbo的名字为attributeName的字段的状态，其中attributeName为要设置的字段的名字，为String型，flag是要设置的状态标识，它是个long型，在psdi.mbo.MboConstants这个接口中有这个标志的常量，state为要设置字段的状态，它是boolean型。

使用方法：例如，要设置字段为只读——`Mbo.SetFieldFlag("name",MboConstants.READONLY,true);`

setFlag(flag, state)

说明：设置Mbo的flags设置值，通常设置NOUPDATE(是否可以更新)，NODELETE(是否可以删除)，而READONLY(是否只读)是设置所有的字段是否只读。

使用方法：如，要设置不能更新——`setFlag(MboConstants.NOUPDATE,true);`

setValue(name, attributeName)

说明：设置Mbo的某个字段的值，通常用在一些字段写默认值等，其中attributeName (String型)为字段的名称，value(java基本数据类型)为要设置的值；

使用方法：如——`setValue("username","maxadmin");`是把属性名为“username”的字段值设为“maxadmin”。

setValueNull(attributeName)

说明：设置Mbo的某个字段为空(null)，其中attributeName (String型)为字段的名称。

使用方法：如——`setValueNull ("username");`是把属性名为“username”的字段值设为null。

toBeAdded()

说明：检查Mbo是不是待增加到数据库中的，返回boolean，如果是待增加的，则返回true，否则返回false。

使用方法：直接调用此方法即可。

isNew()

说明：检查Mbo是不是新建的(通过add()或者insert()方法增加的)，返回boolean，如果是新建的，则返回true，否则返回false。

使用方法：直接调用此方法即可。

toBeUpdated()

说明：检查Mbo是不是待更改的，返回boolean，如果是待更改的，则返回true，否则返回false。

使用方法：直接调用此方法即可。

toBeDeleted ()

说明：检查Mbo是不是待删除的，返回boolean，如果是待删除的，则返回true，否则返回false。

使用方法：直接调用此方法即可。

toBeSaved ()

说明：检查Mbo是不是需要保存的，检查的时候同时检查所关联的对象，返回boolean，如果是需要保存的，则返回true，否则返回false。

使用方法：直接调用此方法即可。

9.MboSet中常用类及说明

MboSet相当于一个集合，对集合中的Mbo进行管理。

save()

说明：保存该对象和与该对象相关联的对象，子类可以根据自己的逻辑来包含适当的对关联对象更新，然后调用super.save()。

使用方法：当保存的时候调用此方法，实际中经常复写此方法完成自己的操作

cleanup()

说明：该方法是重置Qbe和结果集，然后再关闭游标。

close()

说明：关闭对象集合、释放数据库资源，关闭之后所有的方法都还可以调用，不过像next、previous等方法只能得到集合关闭前接收到的东西。

使用方法：在实际中如果对一个比较小的对象集合操作，可以先接收所有的Mbo，然后用该方法释放他们。

commit()

说明：把上一次commit或者rollback之后的改变提交到数据库，这个方法可以被框架自动调用，而不必在程序里直接调用。

posted on 2012-04-14 15:31 [hailiang](#) 阅读(3114) [评论\(0\)](#) [编辑](#) [收藏](#)

[新用户注册](#) [刷新评论列表](#)

只有注册用户[登录](#)后才能发表评论。

Powered by:

BlogJava

Copyright © hailiang