



Spectrum Software, Inc.
11445 Johns Creek Pkwy.
Suite 300
Duluth, GA – 30097
www.spectrumscm.com

Subject: **SpectrumSCM API Concepts and Usage**

Issue Date: **Jan 20th, 2003**

From: **William C. Brown**
corey@spectrumsoftware.net
(770)813-4952

1.0 Introduction: SpectrumSCM is a process driven Source Configuration Management System that can be used to manage the life cycle of any electronic asset. Users of the system define workflows in the tool that correspond to the processes that are already in place within their organizations. By default, workflows defined in SpectrumSCM are considered ad-hoc, which means that work items can be assigned from any user defined phase into any other user defined phase¹. The responsibility of moving work items from one phase to another falls on the shoulders of users that have been assigned that particular responsibility² (Project Managers, etc...)

The purpose of the SpectrumSCM API (application programming interface) is to allow users of the system to construct automated workflow and external system interfaces by implementing a well defined set of event triggers and interfaces. An automated workflow system relieves the burden of making particular users or team leaders responsible for work item progression and adds the ability to automate decision making processes, including the automation of higher level approvals for work items. The API also allows for the definition of interfaces to external systems. External system integration allows issue tracking numbers and content from external systems to be easily integrated into and out of the SpectrumSCM system.

Custom programs that implement the SpectrumSCM APIs are known as *plugins* in SpectrumSCM. *Plugins* are easily compiled and can easily be added to a running SpectrumSCM server through an XML interface. Plugins are loaded dynamically into the server at run time. Dynamic loading also allows the plugins to be changed by the developer and reloaded without impacting the server. The SpectrumSCM plugin configuration allows for one or more plugins to be actively configured into the system. Individual plugins can be turned on or off by simply modifying the plugins XML definition file.

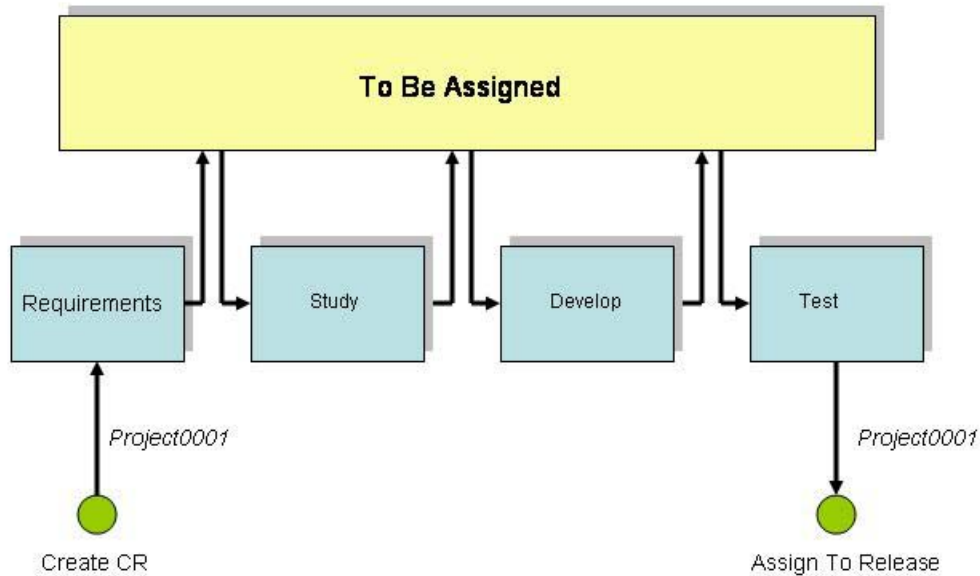
2.0 Manual vs. Automated Workflow: By default, the SpectrumSCM system provides for a manual only workflow process. This decision was made because of the complexities and ambiguities involved with trying to support an

¹ Only users with the proper permissions model can actually assign workitems.

² See the SpectrumSCM User Guide chapter 5 on User Management.

automated workflow system right out of the box. Every organization has a different definition of what a workflow system should accomplish and how it should work. Our goal was to be able to handle the common denominator of all workflow systems and then allow for complete customization through server extensions. SpectrumSCM has accomplished this goal with their base system and their API extensions.

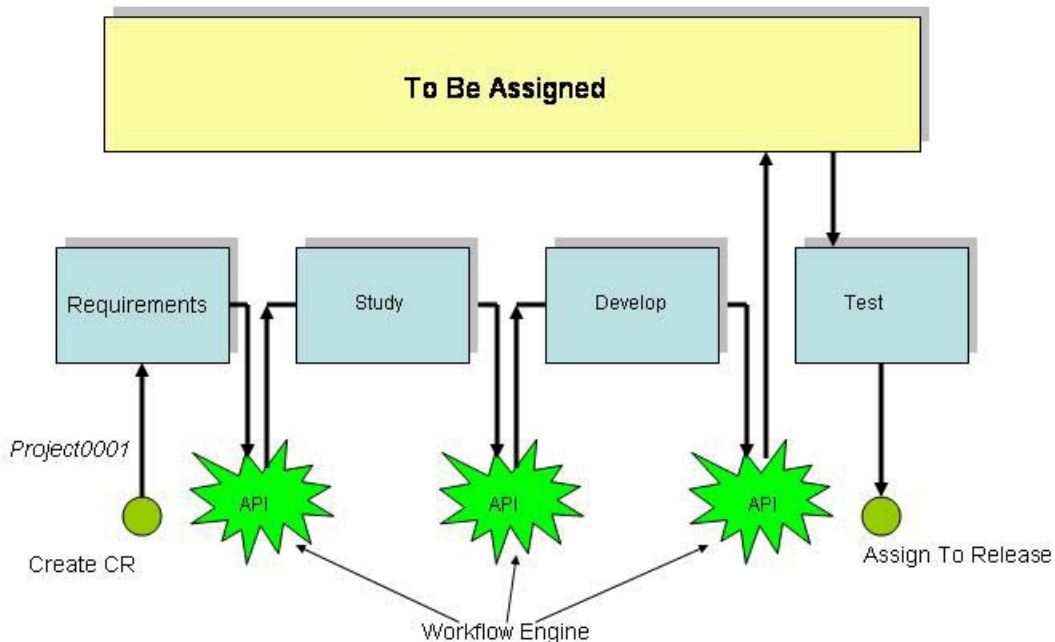
3.0 A Typical Manual Process Workflow: The SpectrumSCM system allows the user to create completely customizable workflows. With the addition of the API, users can use automation to tailor their business process workflow to be cognizant of overall business practices. This allows for external business rules to be executed as part of the transition of an issue or Change request. The following workflow diagram is an example of a trivial software development workflow.



In this example, there are four user defined life-cycle phases *Requirements*, *Study*, *Develop* and *Test*. In the SpectrumSCM system, Change Requests can be created and immediately assigned to any phase within the defined life-cycle. In this example, the CR (*Project0001*) was initially created and assigned to the *Requirements* phase. As Change Requests are manually progressed from phase to phase by the end user, they are not immediately progressed into the next user defined phase. Each Change Request is first

progressed into the **TBA³** super phase. The TBA super phase is where all manually progressed Change Requests end up, unless they are directly assigned into another user defined state by a user with the proper permissions. When Change requests are promoted into the TBA phase, e-mail notifications are sent to the appropriate users so that process decisions about where the CR should go next can be made. The appropriate user would then manually assign the CR into the next appropriate phase.

4.0 An Automated Workflow: In an automated workflow, progression decisions for manually progressed Change Requests are delegated to an automated workflow engine. The engine, built using the SpectrumSCM API, can apply business process decisions to each individual CR and assign them to the next responsible user, or place the CR in a holding pattern until certain business rules have been satisfied. For instance, a CR may not be eligible for promotion from the *Develop* phase into the *Test* phase until some form of code review has been performed. This diagram depicts the flow of control:



In this workflow, Change Requests are manually progressed out of each phase. When the Change Request enters the TBA super phase, the user defined automated workflow engine is immediately activated and decides where the Change Request should go next, and to whom it should be

³ TBA = To Be Assigned

assigned. In this example, when a CR is progressed out of the **Develop** phase it is not immediately progressed into the **Test** phase. But rather it is left in the TBA super phase by the workflow engine until certain business rules have executed (code review?).

5.0 API activation points: The SpectrumSCM API provides for four separate activation points. The activation points can be used to enable a single automated workflow system, or can be used separately to implement independent workflow functionality. Plugins that are associated with the activation points are run in separate threads of execution. Running the plugins in separate threads guarantees that the basic responsibilities of the SpectrumSCM server are never blocked.

5.1 System Startup and Shutdown: There are two activations points specifically designed to work with system startup and shutdown. When the SpectrumSCM server is started or stopped, all registered plugins are searched to see which ones implement the *SystemListener* interface. Each plugin that implements this interface is called from a separate thread of execution. It is possible that, given the length of execution of any particular startup or shutdown transaction, several plugin transactions may run concurrently.

5.1.1 Example Usage: Use the startup and shutdown activation points to connect the SpectrumSCM server to external trouble tracking systems or other external business systems.

5.2 Change Request Transition: Change Request transitions define the third plugin activation point. When a Change Request transitions from a user defined life-cycle phase into the TBA super phase, all plugins that implement the *ChangeRequestListener* interface are executed in separate threads. The change request transition activation point is one of the most useful plugin activation points. Fully automated workflow systems will use this activation point as an event trigger for applying custom change request routing logic and other business processes.

5.2.1 Example Usage: Use the change request transition activation point to implement an business rule sensitive automated workflow system.

5.3 Change Request Creation: Change Request creation defines the last plugin activation point. When a new Change Request is manually created in the SpectrumSCM system, this plugin activation point will be called in a separate thread of execution.

5.3.1 Example Usage: Use the change request creation activation point to communicate change request creations to downstream project management systems.

6.0 Implementing the Interfaces: In order to create a user defined automated workflow system, API users must first implement one or both of the interfaces defined by the API described below.

6.1 The *SystemListener* Interface: This interface defines the methods that are used by the startup and shutdown activation points. Users can implement this interface to create long running processes that are integrated directly into the SpectrumSCM server itself. For example, this interface could be implemented in order to create an active interface with an external system. The following code block illustrates how to implement the *SystemListener* interface:

```
import scm.pub.interfaces.SystemListener;
public class WorkflowEngine implements SystemListener {
    public void startUp() {...}
    public void shutDown() {...}
}
```

6.2 The *ChangeRequestListener* Interface: This interface defines the methods that are called during Change Request creation and transition. The *ChangeRequestListener* interface defines the following two methods:

- `changeRequestTransition(String project, ChangeRequest_d cr_d)`
- `changeRequestCreated(String project, ChangeRequest_d cr_d)`

These two methods are passed the name of the project as a String and the data structure `ChangeRequest_d`. The `ChangeRequest_d` data structure contains all of the current information for the CR involved in the creation or transition. Users would need to write code similar to the following snippet in order to implement the *ChangeRequestListener* interface:

```
import scm.pub.interfaces.ChangeRequestListener;
import scm.pub.transport.ChangeRequest_d;

public class WorkflowEngine implements ChangeRequestListener {
    public void changeRequestTransition(String project, ChangeRequest_d cr_d) {
        ...
    }

    public void changeRequestCreated(String project, ChangeRequest_d cr_d) {
        ...
    }
}
```

7.0 Interacting with the System: There are three first class objects defined in the SpectrumSCM API that can be used to interact with a running SpectrumSCM

server. These objects implement the *Proxy* design pattern⁴ as described in the GOF (Gang of Four) design patterns book (see the footnote). Each object is a proxy or stand in for the corresponding persistent object located in the SpectrumSCM server.

- **ChangeRequest:** The ChangeRequest object is a proxy object for a live ChangeRequest object located in the SpectrumSCM server. Calling the getInfo() method on this object will result in all of the latest information for this particular Change Request to be returned. This object can be used to promote the Change Request into another phase or to add history elements and other notes directly to the Change Request.
- **Project:** The Project object, just like the ChangeRequest object, is a proxy object for the live Project object located in the SpectrumSCM server. This object defines methods that allow the caller to extract project related information directly from the server. This object also contains methods that allow for the creation of new ChangeRequests.
- **ScmSystem:** The ScmSystem object, just like the other objects, is a proxy for the actual ScmSystem object. This object implements both the *Proxy* design pattern as well as the *Singleton* design pattern. The getInstance() method is used to retrieve **the one and only instance** of this object. The ScmSystem object contains methods that allow the caller to retrieve a list of all active projects in the system as well as a list of all registered system users. The object also contains an interface into an E-mail system, which allows the caller to send E-mail messages to interested parties.

A workflow engine can be designed to use these objects transiently for short term operations, or the objects, once constructed, can be stored at a higher scope level for use at a later time. The decisions for the design of the workflow engine are left up to the implementer.

7.1 Transport Objects: Transport objects are used as simple data structures to pass large amounts of information into and out of the proxy objects. The following is a list of all of the Transport Objects defined in the SpectrumSCM API.

- **AttributeMap_d:** An AttributeMap_d object is returned from the method Project.getProjectChangeRequestAttributes(). The AttributeMap_d object is a mapping of Change Request attribute names to a set of attribute values.

⁴ Gamma, Helm, Johnson, Vlissides: Design Patterns, Elements of Reusable Object-Oriented Software. 1995 ISBN: 0-201-63361-2

- **ChangeRequest_d:** An ChangeRequest_d object is returned from the method ChangeRequest.getInfo(). This data structure contains all of the current and historical information for the given Change Request.
- **ChangeRequestCreator_d:** This object is specifically used to create a new Change Request in the system. The contents of this object describe who the new Change Request will be assigned to, in what phase and on which Generic (branch).
- **ChangeRequestHistory_d:** The ChangeRequestHistory_d object is actually a sub-object that is returned as part of the ChangeRequest_d object. It contains historical information about the Change Request.
- **User_d:** An User_d object is returned from the method ScmSystem.getUserInfo() as well as Project.getUserInfo(). In the case of calling the getUserInfo() method on the Project object, more information about the users current category assignments are returned.

8.0 Compiling the Code: In order to compile a custom workflow engine, the developer must have access to the SpectrumSCM server jar files. These jar files are all located in the following directory:

<SCM_INSTALL_DIR>/lib

The developer's CLASSPATH environment variable must be extended to include the **scmServer.jar** jar file. The extended CLASSPATH variable should look like the following when complete:

CLASSPATH="\$CLASSPATH:<SCM_INSTALL_DIR>/lib/scmServer.jar"

Note that path separators are platform dependent. This example will work on a Unix like platform. Once the CLASSPATH variable is set properly, compile the code with the normal java compiler arguments.

9.0 Installing the Code: The developer's compiled code must be included in the SpectrumSCM server's CLASSPATH. A directory named *custom_plugins* already exists in the directory structure and the developer's code can be placed in this directory.

<SCM_INSTALL_DIR>/SCM_VAR/custom_plugins

If the developers code needs to reside in a jar file, the jar can be placed in the SpectrumSCM server lib directory. The script that is used to start the

server must be modified to include this jar file. If the server is running on a Windows platform edit the file **startServer.bat**. If the server is running on a Unix or Mac platform, edit the file **startServer**.

10.0 Modify the Plugins XML file: In order to tell the SpectrumSCM server that a plugin has been added to the system, the plugins XML file must be modified. This file is located in the following directory:

<SCM_INSTALL_DIR>/SCM_VAR/etc/plugins.xml

The following is an example of a valid plugins.XML file:

```
<!-- This is an example of what the plugin file should look like -->
<!-- Note that the STATUS element can be set to either ENABLED OR DISABLED -->

<PLUGINS>
  <!-- Put your plugin declarations here -->
  <PLUGIN>
    <NAME VALUE="Coreys Plugin"/>
    <CLASS VALUE="com.scm.TestPlugin"/>
    <PROJECT VALUE="SCM"/>
    <STATUS VALUE="ENABLED"/>
  </PLUGIN>
</PLUGINS>
```

There are four (4) XML elements that must be in this file. Each element is described below in the following bullet list.

- **NAME:** This is simply the name of the plugin and is used initially as a key to the plugin itself.
- **CLASS:** This is the actual class name of the class that implements the listeners described above.
- **PROJECT:** This is the Project name that this particular plugin should be associated with. The same plugin can be associated with separate Projects as long as the developer is careful not to include Java class attributes with global class scope in the plugin.
- **STATUS:** This determines whether the plugin should be used or not. Set this to DISABLED if the plugin needs to be turned off.

11.0 Example plugins: The basic plugin skeleton is trivial to construct. Here is a working plugin that implements both interfaces but doesn't really do anything:

```
import java.io.*;

import scm.pub.interfaces.*;
import scm.pub.transport.*;
import scm.pub.exceptions.*;

public class TestPlugin implements SystemListener, ChangeRequestListener {

    public void startUp() {
        System.out.println("Startup called..");
    }

    public void shutDown() {
        System.out.println("ShutDown called..");
    }

    public void
    changeRequestCreated(String project, ChangeRequest_d cr_d) {
        System.out.println("changeRequestCreated..");
    }

    public void
    changeRequestTransition(String project, ChangeRequest_d cr_d) {
        System.out.println("ChangeRequestTransition called...");
    }
}
```

Note that the only thing this plugin does is report to standard output when the interface methods have fired. Compile and add this plugin to the system to see which user level actions cause these methods to execute. For instance, creating a new Change Request will cause the `changeRequestCreated()` method to execute. Progressing that CR into the TBA state will cause the `changeRequestTransition()` method to execute. And obviously the `start()` and `stop()` methods will execute when the server is started and stopped.

This next code snippet accesses all of the major first class objects and extracts some information out of a Change Request:

```
public void
changeRequestCreated(String project, ChangeRequest_d cr_d) {
    java.lang.System.out.println(cr_d.toString());

    try {
        ScmSystem    sys = ScmSystem.getInstance();
        Project      proj = new Project(project);
        ChangeRequest cr  = new ChangeRequest(proj, cr_d.getCRId());

        System.out.println(cr.getInfo().toString());

    } catch(Exception e) {
        System.err.println("Caught: " + e.getMessage());
    }
}
```

Note that this code is actually redundant. The `ChangeRequest_d` information that was extracted from the `cr.getInfo()` call was already handed to the enclosing method as an argument. The example is just to show how to access some of the more important objects. Also notice that the `System` object is accessed by simply calling the static method `getInstance()` on the `System` class. `Projects` and `ChangeRequests` can be constructed as often as necessary. All of these objects can be stored for later use once they have been constructed.

This next example is a more complete example of an automated workflow engine. In this code snippet, the ChangeRequest passed to the transition method is examined and automatically progressed into the next life-cycle phase:

```

public void changeRequestTransition(String project, ChangeRequest_d cr_d) {

    ScmSystem    sys = null;
    Project      proj = null;
    ChangeRequest cr  = null;
    Vector       phases = null;
    try {
        sys = ScmSystem.getInstance();
        proj = new Project(project);
        phases = proj.getLifeCyclePhases();
        cr = new ChangeRequest(proj, cr_d.getCRId());
    } catch(Exception e) {
        System.err.println("Caught: " + e.getMessage());
        return;
    }

    try {
        ChangeRequest_d      crObj = cr.getInfo();
        ChangeRequestHistory_d crh_d = null;

        Vector      history = crObj.getHistoryInfo();
        String      lastPhase = null;
        String      nextPhase = null;
        int         index = -1;

        for(int indx = history.size() - 1; indx >= 0; indx--) {
            crh_d = (ChangeRequestHistory_d)history.get(indx);
            if(crh_d.getPhase().endsWith("note")) {
                continue;
            } else {
                lastPhase = crh_d.getPhase();
                break;
            }
        }
        index = phases.indexOf(lastPhase);
        nextPhase = (String)phases.get(index+1);
        cr.assignToPhase(crh_d.getUser(), crObj.getCurrentGeneric(),
            nextPhase, "Here's some more work");
        sys.sendEMail("joe@x.com", "CR Status", "Assigned CR <" +
            crObj.getCRId() + "> to phase <" +
            nextPhase + ">");
    } catch(Exception e) {
        System.err.println("Caught: " + e.getMessage());
    }
}

```

Unfortunately, in order to get all of the code into this single example, some of the empty lines had to be removed from the text and the vast majority of the

error handling code has also been removed. The last few lines in the example are the most important. The method `assignToPhase()` called against the `ChangeRequest` actually assigns this particular CR to the next phase in the life-cycle and adds a small note. The next line uses the e-mail interface to send mail to an interested party.

12.0 Summary: The SpectrumSCM API allows a developer to easily create a fully automated workflow engines and external system interface points. Currently the API is limited to this type of functionality. The developers of the API chose to exclude a file level listener interface from the current API implementation. The existence of such a listener has limited use in a fully integrated tool like SpectrumSCM. One of the basic tenets of SpectrumSCM is that individual files are worked or changed as part of a larger issue or change request. In this scenario, the need to know when a single file has changed, or to act upon a single file change is unnecessary. In other systems that are *interfaced* instead of *integrated*, this type of functionality may be necessary as individual file changes are not already associated with a traceable statement of work.

Appendix – A: Java Docs

[Overview](#) [Package](#) **[Class](#)** [Tree](#) [Deprecated](#) [Index](#) [Help](#)

PREV CLASS [NEXT CLASS](#)

[FRAMES](#) [NO FRAMES](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

1.1 *scm.pub.interfaces* *Interface ChangeRequestListener*

public interface **ChangeRequestListener**

The ChangeRequest listener interface

Method Summary

void	changeRequestCreated (java.lang.String project, ChangeRequest d crd) This method will be called whenever a ChangeRequest is created.
void	changeRequestTransition (java.lang.String project, ChangeRequest d crd) This method will be called whenever a ChangeRequest transitions into the TBA state.

Method Detail

1.1.1 **changeRequestTransition**

```
public void changeRequestTransition(java.lang.String project,
                                     ChangeRequest d crd)
```

This method will be called whenever a ChangeRequest transitions into the TBA state.

Parameters:

`project` - The name of the project involved.
`crd` - The current ChangeRequest information.

1.1.2 **changeRequestCreated**

```
public void changeRequestCreated(java.lang.String project,
                                   ChangeRequest d crd)
```

This method will be called whenever a ChangeRequest is created.

Parameters:

`project` - The name of the project involved.
`crd` - The current ChangeRequest information.

[Overview](#) [Package](#) **[Class](#)** [Tree](#) [Deprecated](#) [Index](#) [Help](#)

PREV CLASS [NEXT CLASS](#)

[FRAMES](#) [NO FRAMES](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

[Overview](#) [Package](#) **[Class](#)** [Tree](#) [Deprecated](#) [Index](#) [Help](#)

PREV CLASS [NEXT CLASS](#)

[FRAMES](#) [NO FRAMES](#)
DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

1.2 *scm.pub.transport* Class *AttributeMap_d*

```
java.lang.Object
|
+--scm.pub.transport.AttributeMap_d
```

```
public final class AttributeMap_d
extends java.lang.Object
AttributeMap_d
```

Constructor Summary

AttributeMap_d ()	
	Create a new empty AttributeMap_d

Method Summary

void	addValueSet (java.lang.String key, java.util.Vector values) Add a set of values to a particular key.
java.util.HashSet	getValueSets (java.lang.String key) Get the value sets for the supplied key.

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

1.2.1 AttributeMap_d

```
public AttributeMap_d()
Create a new empty AttributeMap_d
```

Method Detail

1.2.2 addValueSet

```
public void addValueSet (java.lang.String key,
                        java.util.Vector values)
Add a set of values to a particular key.
Parameters:
key - The key value for the supplied set of values
values - A Vector of Strings as values for the supplied key
```

1.2.3 `getValueSets`

```
public java.util.HashSet getValueSets(java.lang.String key)
```

Get the value sets for the supplied key.

Parameters:

`key` - The key for the requested values

Returns:

The return value is a HashSet of Vectors. Each Vector contains a set of Strings, which represent a set of values for the supplied key.

[Overview](#) [Package](#) **[Class](#)** [Tree](#) [Deprecated](#) [Index](#) [Help](#)

PREV CLASS [NEXT CLASS](#)

[FRAMES](#) [NO FRAMES](#)

SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

[Overview](#) [Package](#) **[Class](#)** [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

[FRAMES](#) [NO FRAMES](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

1.3 *scm.pub.transport* Class *ChangeRequest_d*

```
java.lang.Object
|
+-scm.pub.transport.ChangeRequest_d
```

```
public final class ChangeRequest_d
extends java.lang.Object
ChangeRequest_d
```

Constructor Summary

[ChangeRequest_d](#) (scm.persistent.ChangeRequest_p cr_p)
Create a ChangeRequest_d object from a persisten ChangeRequest_p.

Method Summary

java.util.Vector	getAttachments () Get the list of attachments associated with this CR.
java.util.HashMap	getAttribute () Get a HashMap of ChangeRequest attributes and their values.
java.lang.String	getCreatedState () Get the Creation State of this ChangeRequest
java.lang.String	getCreator () Get the User id of the person who created this ChangeRequest
java.lang.String	getCRId () Get the CRid of this ChangeRequest
int	getCRNum () Get just the CR Number of this ChangeRequest
java.lang.String	getCurrentGeneric () Get the Current Generic that this ChangeRequest is assigned to.
java.lang.String	getCurrentState () Get the Current State this ChangeRequest
java.lang.String	getCurrentUser () Get the user that this CR is assigned to.
java.lang.String	getDateCreated () Get the Creation Date of this ChangeRequest
java.lang.String	getDateLastUsed () Get the last used Date of this ChangeRequest
java.lang.String	getDescription ()

	Get the CR description of this ChangeRequest
java.lang.String	getHeader () Get the CR header of this ChangeRequest
java.util.Vector	getHistoryInfo () Get a list of ChangeRequestHistory_d objects for this ChangeRequest
java.lang.String	toString () Dump the contents of this change request into a String

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Constructor Detail

1.3.1 ChangeRequest_d

```
public ChangeRequest_d(scm.persistent.ChangeRequest_p cr_p)
    throws java.lang.Exception
    Create a ChangeRequest_d object from a persistent ChangeRequest_p.
Throws:
    java.lang.Exception - If the persistent information cannot be accessed.
```

Method Detail

1.3.2 getCRNum

```
public int getCRNum()
    Get just the CR Number of this ChangeRequest
Returns:
    The CRid as a String.
```

1.3.3 getCRId

```
public java.lang.String getCRId()
    Get the CRid of this ChangeRequest
Returns:
    The CRid as a String.
```

1.3.4 getHeader

```
public java.lang.String getHeader()
    Get the CR header of this ChangeRequest
Returns:
    The CR header as a String.
```

1.3.5 getDescription

```
public java.lang.String getDescription()
    Get the CR description of this ChangeRequest
Returns:
    The CR description as a String.
```

1.3.6 **getCreator**

```
public java.lang.String getCreator()
```

Get the User id of the person who created this ChangeRequest

Returns:
The user id as a String.

1.3.7 **getCreatedState**

```
public java.lang.String getCreatedState()
```

Get the Creation State of this ChangeRequest

Returns:
The Creation State as a String.

1.3.8 **getDateCreated**

```
public java.lang.String getDateCreated()
```

Get the Creation Date of this ChangeRequest

Returns:
The Creation Date as a String.

1.3.9 **getDateLastUsed**

```
public java.lang.String getDateLastUsed()
```

Get the last used Date of this ChangeRequest

Returns:
The last used Date as a String.

1.3.10 **getCurrentState**

```
public java.lang.String getCurrentState()
```

Get the Current State this ChangeRequest

Returns:
The Current State as a String.

1.3.11 **getCurrentGeneric**

```
public java.lang.String getCurrentGeneric()
```

Get the Current Generic that this ChangeRequest is assigned to.

Returns:
The Current Generic as a String.

1.3.12 **getCurrentUser**

```
public java.lang.String getCurrentUser()
```

Get the user that this CR is assigned to.

Returns:
The Current User as a String.

1.3.13 **getAttachments**

```
public java.util.Vector getAttachments()
```

Get the list of attachments associated with this CR.

Returns:
A list of Strings of attachment names.

1.3.14 `getHistoryInfo`

```
public java.util.Vector getHistoryInfo()
```

Get a list of `ChangeRequestHistory_d` objects for this `ChangeRequest`

Returns:
A list of `ChangeRequestHistory_d` objects.

1.3.15 `getAttribute`

```
public java.util.HashMap getAttribute()
```

Get a `HashMap` of `ChangeRequest` attributes and their values.

Returns:
A map of key/value pairs of the CR attributes and their values.

1.3.16 `toString`

```
public java.lang.String toString()
```

Dump the contents of this change request into a `String`

Overrides:
`toString` in class `java.lang.Object`

Returns:
A `String` containing the complete contents of this `ChangeRequest`

[Overview](#) [Package](#) **[Class](#)** [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

[FRAMES](#) [NO FRAMES](#)

SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

[Overview](#) [Package](#) **[Class](#)** [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

[FRAMES](#) [NO FRAMES](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

1.4 scm.pub.transport Class ChangeRequestCreator_d

```
java.lang.Object
|
+--scm.pub.transport.ChangeRequestCreator_d
```

public final class **ChangeRequestCreator_d**
extends java.lang.Object

ChangeRequestCreator_d. Use an instance of this class and the Project gateway to create a new ChangeRequest in the system.

Constructor Summary

[ChangeRequestCreator_d](#)(java.lang.String creator, java.lang.String assignee, java.lang.String header, java.lang.String description, java.lang.String creationState, java.lang.String assignedState, java.lang.String assignedGeneric, java.util.HashMap attributes)
Create a ChangeRequest_d object from a persisten ChangeRequest_p.

Method Summary

java.lang.String	getAssignedGeneric () Get the assigned generic information for this CR.
java.lang.String	getAssignedState () Get the assigned state information for this CR.
java.lang.String	getAssignee () Get the assignee information for this CR.
java.util.HashMap	getAttributes () Get the attribute/value pairs for this CR.
java.lang.String	getCreationState () Get the creation state information for this CR.
java.lang.String	getCreator () Get the creator information for this CR.
java.lang.String	getDescription () Get the description information for this CR.
java.lang.String	getHeader () Get the header information for this CR.
void	setAssignedState (java.lang.String state) Set the assigned state for this CR.
void	setAttributeAndValue (java.lang.String attribute,

	java.lang.String value) Add a CR attribute/value pair to this CR.
void	setCreatedState (java.lang.String state) Set the Creation state for this CR
void	setDescription (java.lang.String desc) Set the description for this CR
void	setGeneric (java.lang.String generic) Set the assigned generic for this CR
void	setHeader (java.lang.String header) Set the head for this CR

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

1.4.1 ChangeRequestCreator_d

```
public ChangeRequestCreator_d(java.lang.String creator,
                               java.lang.String assignee,
                               java.lang.String header,
                               java.lang.String description,
                               java.lang.String createState,
                               java.lang.String assignedState,
                               java.lang.String assignedGeneric,
                               java.util.HashMap attributes)
    throws InvalidUser
```

Create a ChangeRequest_d object from a persistent ChangeRequest_p.

Parameters:

creator - The user that is creating this CR.
 assignee - The user that this CR is to be assigned to.
 header - The header information for this CR. This is usually limited to a single line of text.
 description - The multi-line description for this CR.
 createState - The current project life-cycle state. For instance, if the current project is in the "Testing" phase, then this attribute would be set to "Testing".
 assignedState - The phase in the development life-cycle that this CR should be set to.
 assignedGeneric - The generic/branch that this CR should be assigned to.
 attribute - A HashMap of attribute/value pairs for this CR.

Throws:

[InvalidUser](#) - If the user information is incorrect.

Method Detail

1.4.2 getAssignedGeneric

```
public java.lang.String getAssignedGeneric()
```

Get the assigned generic information for this CR.

Returns:
 The assigned generic information as a String.

1.4.3 `getAttributes`

```
public java.util.HashMap getAttributes()
```

Get the attribute/value pairs for this CR.

Returns:
The attribute/value pairs as a HashMap.

1.4.4 `getAssignedState`

```
public java.lang.String getAssignedState()
```

Get the assigned state information for this CR.

Returns:
The assigned state information as a String.

1.4.5 `getCreationState`

```
public java.lang.String getCreationState()
```

Get the creation state information for this CR.

Returns:
The creation state information as a String.

1.4.6 `getAssignee`

```
public java.lang.String getAssignee()
```

Get the assignee information for this CR.

Returns:
The assignee information as a String.

1.4.7 `getCreator`

```
public java.lang.String getCreator()
```

Get the creator information for this CR.

Returns:
The creator information as a String.

1.4.8 `getDescription`

```
public java.lang.String getDescription()
```

Get the description information for this CR.

Returns:
The description information as a String.

1.4.9 `getHeader`

```
public java.lang.String getHeader()
```

Get the header information for this CR.

Returns:
The header information as a String.

1.4.10 `setAttributeAndValue`

```
public void setAttributeAndValue(java.lang.String attribute,  
                                 java.lang.String value)
```

Add a CR attribute/value pair to this CR. CR attributes and values are well defined for a particular project. Use the Project object to get the applicable values for the parameters to this call.

Parameters:

`attribute` - An attribute for this CR.
`value` - The value of the supplied attribute.

1.4.11 `setCreatedState`

```
public void setCreatedState(java.lang.String state)
    Set the Creation state for this CR
```

Parameters:

`state` - The creation state for a CR does not have to match the assigned state for the CR. The creation state is an indication of the current state of development that this CR was created in.

1.4.12 `setAssignedState`

```
public void setAssignedState(java.lang.String state)
    Set the assigned state for this CR. The assigned state can be any within the projects associated life-cycle. Use the Project object to get a complete list of possible states.
```

Parameters:

`state` - The state to assign this CR to.

1.4.13 `setHeader`

```
public void setHeader(java.lang.String header)
    Set the head for this CR
```

Parameters:

`header` - The header information for this CR. Normally the header is no more than one (1) line long.

1.4.14 `setDescription`

```
public void setDescription(java.lang.String desc)
    Set the description for this CR
```

Parameters:

`desc` - The Description for a CR is any amount of text and can include embedded new lines.

1.4.15 `setGeneric`

```
public void setGeneric(java.lang.String generic)
    Set the assigned generic for this CR
```

Parameters:

`generic` - The generic that this CR should be assigned to.

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

[FRAMES](#) [NO FRAMES](#)

SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

[Overview](#) [Package](#) **[Class](#)** [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

[FRAMES](#) [NO FRAMES](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

1.5 scm.pub.transport Class ChangeRequestHistory_d

```
java.lang.Object
|
+-scm.pub.transport.ChangeRequestHistory_d
```

```
public final class ChangeRequestHistory_d
extends java.lang.Object
ChangeRequestHistory_d
```

Constructor Summary

[ChangeRequestHistory_d](#) (scm.transport.CRHistory_d crh)
Construct a new ChangeRequestHistory_d object with the supplied CRHistory_p object

Method Summary

java.lang.String	getAuxInfo () Get any auxillary information associated with this history element.
java.lang.String	getBeginDate () Get the date that this CR transitioned into this state.
java.lang.String	getEndDate () Get the date that this CR transitioned out of this state.
java.lang.String	getPhase () Get the LifeCycle Phase associated with this history element.
java.lang.String	getUser () Get the User associated with this transition
java.lang.String	toString () Convert this object into a String

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Constructor Detail

1.5.1 ChangeRequestHistory_d

```
public ChangeRequestHistory_d (scm.transport.CRHistory_d crh)
    throws java.lang.Exception
```


Construct a new ChangeRequestHistory_d object with the supplied CRHistory_p object

Throws:

java.lang.Exception - If the attributes of the persistent object cannot be accessed.

Method Detail

1.5.2 getUser

```
public java.lang.String getUser()
```

Get the User associated with this transition

Returns:
The User Name as a String.

1.5.3 getBeginDate

```
public java.lang.String getBeginDate()
```

Get the date that this CR transitioned into this state.

Returns:
The Date as a String.

1.5.4 getEndDate

```
public java.lang.String getEndDate()
```

Get the date that this CR transitioned out of this state.

Returns:
The Date as a String.

1.5.5 getPhase

```
public java.lang.String getPhase()
```

Get the LifeCycle Phase associated with this history element.

Returns:
The Date as a String.

1.5.6 getAuxInfo

```
public java.lang.String getAuxInfo()
```

Get any auxillary information associated with this history element.

Returns:
The info as a String.

1.5.7 toString

```
public java.lang.String toString()
```

Convert this object into a String

Overrides:
toString in class java.lang.Object

Returns:
The string representation of this object

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

[FRAMES](#) [NO FRAMES](#)

SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

[Overview](#) [Package](#) **[Class](#)** [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

[FRAMES](#) [NO FRAMES](#)
DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

1.6 scm.pub.transport Class User_d

```
java.lang.Object
|
+--scm.pub.transport.User_d
```

```
public final class User_d
extends java.lang.Object
```

The User_d class contains all the relevant information about a particular user.

Constructor Summary

[User_d](#)([User_d](#) user_d, scm.persistent.PUser_p puser_p)
Create a User_d instance given the supplied User_p and PUser_p persistent objects.

[User_d](#)(scm.persistent.User_p user_p)
Create a User_d instance given the supplied User_p persistent object.

Method Summary

java.lang.String	getAuxInfo () Get any auxiliary information associated with this user.
java.lang.String	getEmailAddress () Get the users EMail address
java.lang.String	getLocation () Get the users location
java.lang.String	getPhone () Get the users phone number
java.util.Vector	getProjects () Get the list of projects that this user is assigned to.
java.lang.String	getUserId () Get the userid
java.lang.String	getUserName () Get the user names
java.util.Vector	getUserRoles () Get the list of user roles that this user is assigned to on this project.
boolean	isAdministrator () Determines whether this user is an SpectrumSCM System Administrator.
boolean	isProjectEngineer () Determines whether this user is an SpectrumSCM Project Engineer.
java.lang.String	toString () Convert this object into a String.

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Constructor Detail**1.6.1 User_d**

```
public User_d(scm.persistent.User_p user_p)
    throws java.lang.Exception
```

Create a User_d instance given the supplied User_p persistent object.

Throws:

java.lang.Exception - If there's a failure accessing any of the persistent object information

1.6.2 User_d

```
public User_d(User\_d user_d,
             scm.persistent.PUser_p puser_p)
    throws java.lang.Exception
```

Create a User_d instance given the supplied User_p and PUser_p persistent objects.

Throws:

java.lang.Exception - If there's a failure accessing any of the persistent object information

Method Detail**1.6.3 getUserId**

```
public java.lang.String getUserId()
    Get the userid
```

Returns:

The userId as a String.

1.6.4 getUserName

```
public java.lang.String getUserName()
    Get the user names
```

Returns:

The user name as a String.

1.6.5 getAuxInfo

```
public java.lang.String getAuxInfo()
    Get any auxillary information associated with this user.
```

Returns:

User auxillary information.

1.6.6 getLocation

```
public java.lang.String getLocation()
    Get the users location
```

Returns:

The users location.

1.6.7 getEmailAddress

```
public java.lang.String getEmailAddress()
```

Get the users EMail address

Returns:
The users EMail address.

1.6.8 getPhone

```
public java.lang.String getPhone()
```

Get the users phone number

Returns:
The users phone number

1.6.9 isAdministrator

```
public boolean isAdministrator()
```

Determines whether this user is an SpectrumSCM System Administrator.

Returns:
A boolean signalling whether this user is an admin user or not.

1.6.10 isProjectEngineer

```
public boolean isProjectEngineer()
```

Determines whether this user is an SpectrumSCM Project Engineer.

Returns:
A boolean signalling whether this user is an project engineer or not.

1.6.11 getProjects

```
public java.util.Vector getProjects()
```

Get the list of projects that this user is assigned to.

Returns:
A Vector of project names as Strings.

1.6.12 getUserRoles

```
public java.util.Vector getUserRoles()
```

Get the list of user roles that this user is assigned to on this project.

Returns:
A Vector of use role names as Strings.

1.6.13 toString

```
public java.lang.String toString()
```

Convert this object into a String.

Overrides:
`toString` in class `java.lang.Object`

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

[Overview](#) [Package](#) **[Class](#)** [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

[FRAMES](#) [NO FRAMES](#)

SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

1.7 scm.pub.exceptions Class SystemException

```

java.lang.Object
|
+-java.lang.Throwable
    |
    +-java.lang.Exception
        |
        +-scm.pub.exceptions.SystemException
  
```

All Implemented Interfaces:

java.io.Serializable

public class **SystemException**

extends java.lang.Exception

See Also:

[Serialized Form](#)

Constructor Summary

[SystemException](#) ()

[SystemException](#) (java.lang.String errorText)

Methods inherited from class java.lang.Throwable

fillInStackTrace, getLocalizedMessage, getMessage, printStackTrace, printStackTrace, printStackTrace, toString

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Constructor Detail

1.7.1 SystemException

public **SystemException** ()

1.7.2 SystemException

public **SystemException** (java.lang.String errorText)

[Overview](#) [Package](#) **[Class](#)** [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

[FRAMES](#) [NO FRAMES](#)

SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

[Overview](#) [Package](#) **[Class](#)** [Tree](#) [Deprecated](#) [Index](#) [Help](#)

PREV CLASS [NEXT CLASS](#)

[FRAMES](#) [NO FRAMES](#)

SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

1.8 *scm.pub.exceptions* Class *InvalidChangeRequest*

```

java.lang.Object
|
+-java.lang.Throwable
   |
   +-java.lang.Exception
      |
      +-scm.pub.exceptions.InvalidChangeRequest
  
```

All Implemented Interfaces:

[java.io.Serializable](#)

public class **InvalidChangeRequest**

extends [java.lang.Exception](#)

See Also:

[Serialized Form](#)

Constructor Summary

[InvalidChangeRequest](#) ()

[InvalidChangeRequest](#) (java.lang.String errorText)

Methods inherited from class [java.lang.Throwable](#)

[fillInStackTrace](#), [getLocalizedMessage](#), [getMessage](#), [printStackTrace](#), [printStackTrace](#), [printStackTrace](#), [toString](#)

Methods inherited from class [java.lang.Object](#)

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [wait](#), [wait](#), [wait](#)

Constructor Detail

1.8.1 **InvalidChangeRequest**

public **InvalidChangeRequest** ()

1.8.2 **InvalidChangeRequest**

public **InvalidChangeRequest** (java.lang.String errorText)

[Overview](#) [Package](#) **[Class](#)** [Tree](#) [Deprecated](#) [Index](#) [Help](#)

PREV CLASS [NEXT CLASS](#)

[FRAMES](#) [NO FRAMES](#)

SUMMARY: INNER | FIELD | [CONSTR](#) | [METHOD](#)

DETAIL: FIELD | [CONSTR](#) | METHOD

[Overview](#) [Package](#) **[Class](#)** [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

[FRAMES](#) [NO FRAMES](#)

SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

1.9 *scm.pub.exceptions* **Class InvalidPhase**

```

java.lang.Object
|
+-java.lang.Throwable
   |
   +-java.lang.Exception
      |
      +-scm.pub.exceptions.InvalidPhase

```

All Implemented Interfaces:

[java.io.Serializable](#)

```

public class InvalidPhase
extends java.lang.Exception

```

See Also:

[Serialized Form](#)

Constructor Summary

[InvalidPhase](#) ()

[InvalidPhase](#) (java.lang.String errorText)

Methods inherited from class java.lang.Throwable

[fillInStackTrace](#), [getLocalizedMessage](#), [getMessage](#), [printStackTrace](#), [printStackTrace](#), [printStackTrace](#), [toString](#)

Methods inherited from class java.lang.Object

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [wait](#), [wait](#), [wait](#)

Constructor Detail

1.9.1 InvalidPhase

```

public InvalidPhase()

```

1.9.2 InvalidPhase

```

public InvalidPhase(java.lang.String errorText)

```

[Overview](#) [Package](#) **Class** [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

[FRAMES](#) [NO FRAMES](#)

SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

[Overview](#) [Package](#) **[Class](#)** [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

[FRAMES](#) [NO FRAMES](#)

SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

1.10 *scm.pub.exceptions* **Class InvalidProject**

```

java.lang.Object
|
+-java.lang.Throwable
   |
   +-java.lang.Exception
      |
      +-scm.pub.exceptions.InvalidProject

```

All Implemented Interfaces:

[java.io.Serializable](#)

public class **InvalidProject**
extends [java.lang.Exception](#)

See Also:

[Serialized Form](#)

Constructor Summary

[InvalidProject](#) ()

[InvalidProject](#) (java.lang.String errorText)

Methods inherited from class [java.lang.Throwable](#)

[fillInStackTrace](#), [getLocalizedMessage](#), [getMessage](#), [printStackTrace](#),
[printStackTrace](#), [printStackTrace](#), [toString](#)

Methods inherited from class [java.lang.Object](#)

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [wait](#),
[wait](#), [wait](#)

Constructor Detail

1.10.1 **InvalidProject**

```
public InvalidProject()
```

1.10.2 **InvalidProject**

```
public InvalidProject(java.lang.String errorText)
```

[Overview](#) [Package](#) **Class** [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

[FRAMES](#) [NO FRAMES](#)

SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

[Overview](#) [Package](#) **[Class](#)** [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

[FRAMES](#) [NO FRAMES](#)

SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

1.11 scm.pub.exceptions Class InvalidUser

```

java.lang.Object
|
+--java.lang.Throwable
    |
    +--java.lang.Exception
        |
        +--scm.pub.exceptions.InvalidUser
  
```

All Implemented Interfaces:

java.io.Serializable

```

public class InvalidUser
extends java.lang.Exception
  
```

See Also:

[Serialized Form](#)

Constructor Summary

[InvalidUser](#) ()

[InvalidUser](#) (java.lang.String errorText)

Methods inherited from class java.lang.Throwable

fillInStackTrace, getLocalizedMessage, getMessage, printStackTrace, printStackTrace, printStackTrace, toString

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Constructor Detail

1.11.1 InvalidUser

```

public InvalidUser ()
  
```

1.11.2 InvalidUser

```

public InvalidUser (java.lang.String errorText)
  
```

[Overview](#) [Package](#) **[Class](#)** [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

[FRAMES](#) [NO FRAMES](#)

SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

[Overview](#) [Package](#) **[Class](#)** [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

[FRAMES](#) [NO FRAMES](#)
 DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

1.12 *scm.pub.gateway.changerequest* Class *ChangeRequest*

```
java.lang.Object
|
+--scm.pub.gateway.changerequest.ChangeRequest
```

public final class **ChangeRequest**
 extends java.lang.Object

A ChangeRequest object is essentially a proxy object for a live persistent ChangeRequest object. Asking this object for the ChangeRequest information **getInfo** will result in the latest ChangeRequest information being pulled from the active project database.

Constructor Summary

[ChangeRequest](#)([Project](#) proj, java.lang.String CRid)
 Create a gateway to a particular ChangeRequest in this project.

Method Summary

void	addNote (java.lang.String user, java.lang.String note) Add a note to this ChangeRequest.
void	assignToPhase (java.lang.String user, java.lang.String generic, java.lang.String phase, java.lang.String note) Assign this ChangeRequest to a particular user, on a particular Generic and to a particular phase.
ChangeRequest d	getInfo () Get all of the latest information for this ChangeRequest.
void	progress (java.lang.String user, java.lang.String note) Progress this ChangeRequest into the TBA state.

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

1.12.1 ChangeRequest

```
public ChangeRequest(Project proj,
                     java.lang.String CRid)
    throws InvalidProject,
```

[InvalidChangeRequest](#),
[SystemException](#)

Create a gateway to a particular ChangeRequest in this project.

Parameters:

proj - The project where the ChangeRequest resides.

CRid - The name of the ChangeRequest to retrieve.

Throws:

[InvalidProject](#) - If the Project name is invalid.

[InvalidChangeRequest](#) - If the change request does not exist.

[SystemException](#) - If a system level exception is caught.

Method Detail

1.12.2 assignToPhase

```
public void assignToPhase(java.lang.String user,
                          java.lang.String generic,
                          java.lang.String phase,
                          java.lang.String note)
    throws InvalidGeneric,
          InvalidUser,
          InvalidChangeRequest,
          InvalidPhase,
          SystemException
```

Assign this ChangeRequest to a particular user, on a particular Generic and to a particular phase.

Parameters:

user - The user this ChangeRequest should be assigned to.

generic - The generic this ChangeRequest should be assigned to.

phase - The phase this ChangeRequest should be assigned to.

note - A note that can be tacked onto the transition.

Throws:

[InvalidUser](#) - If the supplied user id is invalid.

[InvalidGeneric](#) - If the supplied generic is invalid.

[InvalidChangeRequest](#) - If the CR is invalid for whatever reason.

[InvalidPhase](#) - If the supplied phase is invalid.

[SystemException](#) - If an internal exception is caught.

1.12.3 addNote

```
public void addNote(java.lang.String user,
                    java.lang.String note)
    throws InvalidUser,
          SystemException
```

Add a note to this ChangeRequest.

Parameters:

user - The user this note should be associated with.

note - The contents of the note.

Throws:

[InvalidUser](#) - If the supplied user id is invalid.

[SystemException](#) - If an internal exception is caught.

1.12.4 progress

```
public void progress(java.lang.String user,
                    java.lang.String note)
    throws InvalidUser,
```


[InvalidChangeRequest](#),
[SystemException](#)

Progress this ChangeRequest into the TBA state.

Parameters:

user - The user this progression should be associated with.

note - An optional note.

Throws:

[InvalidUser](#) - If the supplied user id is invalid.

[InvalidChangeRequest](#) - If the CR is invalid for whatever reason.

[SystemException](#) - If an internal exception is caught.

1.12.5 getInfo

```
public ChangeRequest\_d getInfo()  
    throws InvalidChangeRequest,  
           SystemException
```

Get all of the latest information for this ChangeRequest. Calling this method results in a pull from the Project database so that up to date information is always returned.

Returns:

A ChangeRequest_d value object containing all of the data contained in this ChangeRequest

Throws:

[InvalidChangeRequest](#) - If the change request does not exist.

[SystemException](#) - If a system level exception is caught.

[Overview](#) [Package](#) **[Class](#)** [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

[FRAMES](#) [NO FRAMES](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

[Overview](#) [Package](#) **[Class](#)** [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

[FRAMES](#) [NO FRAMES](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

1.13 *scm.pub.gateway.project* Class *Project*

```
java.lang.Object
|
+--scm.pub.gateway.project.Project
```

public final class **Project**
extends java.lang.Object

The Project class is a proxy class for a live project defined in a SpectrumSCM federated system. Projects in SpectrumSCM define their own databases and are transactionally independent from the rest of the database federation. Each Project contains a user defined set of lifecycle phases, a set of ChangeRequest attributes with values and a set of users, defined to operate as certain roles within the project.

Constructor Summary

[Project](#)(java.lang.String projectName)
Create a Project associated with the given project name.

Method Summary

java.lang.String	createChangeRequest (ChangeRequestCreator d crc) Create a new ChangeRequest.
java.util.Vector	getActiveChangeRequests () Get all ChangeRequests that are currently active in this Project.
ChangeRequest d	getCRInfo (java.lang.String CRid) Get the information contained in this ChangeRequest.
java.util.Vector	getGenericNames () Get the names of all of the Generics associated with this Project
java.util.Vector	getInactiveChangeRequests () Get all ChangeRequests that are currently inactive in this Project.
java.util.Vector	getLifecyclePhases () Get the life cycle phases defined for this project.
java.lang.String	getName () Get the name of this Project
AttributeMap d	getProjectChangeRequestAttributes () Get the ChangeRequest attributes defined for this project.
User d	getUserInfo (java.lang.String user) Get information about a specific user including their assigned roles on this project.
java.util.Vector	getUsers ()

	Get the users assigned to this project.
boolean	isAttributeEditable (java.lang.String attrib) Determine whether the given project attribute name is user editable or not

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

1.13.1 Project

```
public Project(java.lang.String projectName)
    throws InvalidProject,
           SystemException
```

Create a Project associated with the given project name.

Parameters:

projectName - The name of the Project to connect to on the server.

Throws:

[InvalidProject](#) - If the Project name is invalid.

[SystemException](#) - If an internal exception is caught.

Method Detail

1.13.2 createChangeRequest

```
public java.lang.String createChangeRequest(ChangeRequestCreator d crc)
    throws SystemException,
           InvalidUser,
           InvalidGeneric,
           InvalidPhase
```

Create a new ChangeRequest.

Parameters:

ChangeRequestCreator_d - A ChangeRequestCreator_d object, which describes all of the necessary information for ChangeRequest creation.

Returns:

The name of the newly created CR.

Throws:

[SystemException](#) - If an internal exception is caught.

[InvalidUser](#) - If the assigned user or creation user is invalid.

[InvalidPhase](#) - If the creation or assigned states are invalid.

[InvalidGeneric](#) - If the assignment generic is invalid.

1.13.3 isAttributeEditable

```
public boolean isAttributeEditable(java.lang.String attrib)
    throws SystemException
```

Determine whether the given project attribute name is user editable or not

Parameters:

attribute - The attribute to check.

Returns:

Boolean whether the supplied attribute is user editable or not.

1.13.4 getName

```
public java.lang.String getName ()
```

Get the name of this Project

Returns:

The name of the Project

1.13.5 getGenericNames

```
public java.util.Vector getGenericNames ()
                               throws SystemException
```

Get the names of all of the Generics associated with this Project

Returns:

A Vector<String> of Generic names.

Throws:

[SystemException](#) - If an internal exception is caught.

1.13.6 getUsers

```
public java.util.Vector getUsers ()
                               throws SystemException
```

Get the users assigned to this project.

Returns:

A Vector of User names as Strings.

Throws:

[SystemException](#) - If an internal exception is caught.

1.13.7 getUserInfo

```
public User\_d getUserInfo (java.lang.String user)
                               throws InvalidUser,
                                     SystemException
```

Get information about a specific user including their assigned roles on this project.

Parameters:

User - The names of a particular user.

Returns:

A User_d object containing all of a users information for this project.

Throws:

[InvalidUser](#) - If the supplied user id does not exist.

[SystemException](#) - If an internal exception is caught.

1.13.8 getProjectChangeRequestAttributes

```
public AttributeMap\_d getProjectChangeRequestAttributes ()
                               throws SystemException
```

Get the ChangeRequest attributes defined for this project.

Returns:

The AttributeMap_d that gets returned contains the key and value sets defined for the ChangeRequest attributes in this project.

Throws:

[SystemException](#) - If an internal exception is caught.

1.13.9 getLifeCyclePhases

```
public java.util.Vector getLifeCyclePhases ()
                               throws SystemException
```

Get the life cycle phases defined for this project.

Returns:

A Vector containing String values representing the phases in the lifecycle defined for this project.

Throws:

[SystemException](#) - If an internal exception is caught.

1.13.10 `getInactiveChangeRequests`

```
public java.util.Vector getInactiveChangeRequests ()
                                throws SystemException
```

Get all ChangeRequests that are currently inactive in this Project.

Returns:

A Vector of Strings contains the IDs of all the ChangeRequests in this project that are either **Completed** or **Killed**

Throws:

[SystemException](#) - If an internal exception is caught.

1.13.11 `getActiveChangeRequests`

```
public java.util.Vector getActiveChangeRequests ()
                                throws SystemException
```

Get all ChangeRequests that are currently active in this Project.

Returns:

A Vector of Strings contains the IDs of all the ChangeRequests in this project that are in phases other than **Completed** or **Killed**

Throws:

[SystemException](#) - If an internal exception is caught.

1.13.12 `getCRInfo`

```
public ChangeRequest\_d getCRInfo(java.lang.String CRid)
                                throws InvalidChangeRequest,
                                       SystemException
```

Get the information contained in this ChangeRequest.

Parameters:

CRid - The name of the ChangeRequest.

Returns:

A `ChangeRequest_d` transport object, which contains all of the change request's current information.

Throws:

[InvalidChangeRequest](#) - If the supplied CRid is invalid.

[SystemException](#) - If an internal exception is caught.

[Overview](#) [Package](#) **[Class](#)** [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

[FRAMES](#) [NO FRAMES](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

[Overview](#) [Package](#) **[Class](#)** [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

[FRAMES](#) [NO FRAMES](#)

[DETAIL: FIELD](#) | [CONSTR](#) | [METHOD](#)

1.14 scm.pub.gateway.scmsystem Class ScmSystem

```
java.lang.Object
|
+--scm.pub.gateway.scmsystem.ScmSystem
```

public class ScmSystem
extends java.lang.Object

System

Method Summary

java.lang.String	getDbPath (java.lang.String db) Get the absolute path to a project database
static System	getInstance () Get the System object
java.util.Vector	getProjectNames () Get the names of the Projects defined in this system.
User d	getUserInfo (java.lang.String user) Get information about a specific user on this project.
java.util.Vector	getUsers () Get a listing of all of the Users defined in the System.
void	sendEMail (java.lang.String to, java.lang.String subject, java.lang.String text) Send an E-mail message to the supplied to address.

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Method Detail

1.14.1 getInstance

```
public static ScmSystem getInstance()
    throws SystemException
```

Get the ScmSystem object

Returns:

The ScmSystem object.

1.14.2 getDbPath

```
public java.lang.String getDbPath(java.lang.String db)
```

Get the absolute path to a project database

Parameters:
db - The name of the database to get the path for.

Returns:
The absolute path to the database file. Can return null if the database is unknown.

1.14.3 **getProjectNames**

```
public java.util.Vector getProjectNames()
```

throws [SystemException](#)

Get the names of the Projects defined in this system.

Returns:
A Vector of project names as Strings.

1.14.4 **getUsers**

```
public java.util.Vector getUsers()
```

throws [SystemException](#)

Get a listing of all of the Users defined in the System.

Returns:
A Vector of user names as Strings.

1.14.5 **getUserInfo**

```
public User\_d getUserInfo(java.lang.String user)
```

throws [InvalidUser](#),
[SystemException](#)

Get information about a specific user on this project.

Parameters:
User - The names of a particular user.

Returns:
A [User_d](#) object containing all of a users information except for information that is project specific, which includes the user roles on particular projects.

Throws:
[InvalidUser](#) - If the supplied user id does not exist.
[SystemException](#) - If an internal exception is caught.

1.14.6 **sendEMail**

```
public void sendEMail(java.lang.String to,  
                      java.lang.String subject,  
                      java.lang.String text)
```

throws [SystemException](#)

Send an E-mail message to the supplied **to** address.

Parameters:
from - The address this mail is coming from.
to - The address to send this mail to.
text - The text of the message to send.

Throws:
[SystemException](#) - If an internal exception is caught.

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

