

SDK Developer Guide

Faster software development and integration

8Manage SDK Developer Guide

Revision 1.8

Wisage Technology

Content

1 Summary	1
2 Integration with Web Service Interface	1
3 SDK Customization Features	1
3.1 User Defined Role Privilege	2
3.1.1 Summary	2
3.1.2 Implementation	2
3.1.3 Implementation Examples	4
3.1.4 Special Notes	5
3.2 Request Callback Mechanism	5
3.2.1 Summary	5
3.2.2 Customize Callback Mechanism	5
3.2.3 Commonly Used Methods	8
3.2.3.1 Error Message	8
3.2.3.1.1 Detect Error Message in HTTP Request Context	8
3.2.3.1.2 Save Error Message to HTTP Request Context	8
3.2.3.2 Retrieve Current User	8
3.2.3.3 Retrieve Current Time	9
3.2.3.4 Retrieve Hibernate Session	9
3.2.3.5 Retrieve Page Privilege Information	9
3.2.4 Implementation Examples	9
3.2.4.1 Add Fields to List Page of Business Object	9
3.2.4.1.1 Adding "Approver" on Purchase Request List Page	9
3.2.4.1.2 Adding Other Fields on List Page of Campaign	11
3.2.4.1.3 Adding Other Fields on List Page of Opportunity/Strategy	12
3.2.4.1.4 Adding Other Fields on List Page of Contract	12
3.2.4.1.5 Adding Other Fields on List Page of Internal Human Resource	12
3.2.4.1.6 Adding Other Fields on List Page of Client	13
3.2.4.2 Add Fields to Basic Page of Business Object	13
3.2.4.2.1 Add Supplier Field to Basic Page of Client Contract	13
3.2.4.2.2 Add Other Fields to Basic Page of Campaign	34
3.2.4.2.3 Add Other Fields to Basic Page of Opportunity/Strategy	35
3.2.4.3 Synchronize Information with External Database	36
3.2.4.3.1 Synchronize User Information with External Database	36
3.2.5 Frequently Used 'actionPoint' in 8Manage	40
3.3 Customize Interface Using 8Manage UI Component	45
3.3.1 Summary	45
3.3.2 Implementation	45
3.4 Customize Menu	45
3.4.1 Summary	45
3.4.2 Implementation	45
3.5 Customized Field Name	48
3.5.1 Summary	48
3.5.2 Implementation	48
3.5.3 Implementation Examples	48
3.5.4 Special Notes	50
3.6 Customize Code Table	50
3.6.1 Summary	50
3.6.2 Implementation	50
3.7 Data Privilege Checking Mechanism	51
3.7.1 Summary	51
3.7.2 Call data right checking	51
3.7.3 Type List	52
3.8 Setup SDK Remote Debug Environment	53
3.8.1 Summary	53

3.8.2 Start 8Manage Service in Debug Mode	53
3.9 Implementing Refresh-less Multiple Files Upload and Document Linking	57
3.9.1 Summary	57
3.9.2 Implementation.....	57
3.10 Implementation of Audit Trail	61
3.10.1 Summary	61
3.10.2 Log's Generic Table Structure.....	61
3.10.3 Java Class and Hibernate Configuration Files.....	62
3.10.4 Write Log Table with SQL and Trigger, and Write Key-value in Resource File	62
3.10.5 Code for List Query and Table JSP	65
3.11 Send Customized Email.....	67
3.12 Customize Search Criteria.....	69
3.13 Customize Discussion Name.....	70
4 8Manage Directory Structure	71
4.1 8Manage Default Structure.....	71
4.2 Customization Directory Suggestion.....	75
5 How to Deploy Customized Codes	75
5.1 Deploy JSP Codes.....	75
5.2 Deploy JAVA Codes	75
5.3 Deploy Configuration Files.....	75

1 Summary

This document introduces 8Manage Software Development Kit (SDK) for developers. It covers 8Manage features that support customization by programming; the implementation methods used, and code examples.

In general, 8Manage supports 3 types of customization:

1. Point-and-Click Customization
2. Integration with Web Service Interface
3. Customization with SDK Programming API

This document provides instructions for customization with SDK programming API only. Please refer to the section, SDK Customization Features, for more detail.

2 Integration with Web Service Interface

8Manage supports importing and creating data via its Web Service Interface with external applications, such as creating project in 8Manage with imported .mpp file.

Please refer to a separate document, *8Manage Web Service Interface*, for more information on how to integrate 8Manage with external applications. You can locate it on the 8Manage release CD, if it is part of the release as stated in README.

3 SDK Customization Features

8Manage supports customization by API programming. This section introduces the many SDK features that can help customize system behavior, enhance usability, and special requirement on data modeling and exchange.

3.1 User Defined Role Privilege

3.1.1 Summary

8Manage manages user privilege by assigning each user one or more system “roles”. Each role is defined by enabling or disabling an appropriate set of information access rights (or “role privilege”) to 8Manage features. The available privilege options used in a role privilege definition may not be sufficient. 8Manage SDK allows the customization of privilege options in a role’s definition.

This section introduces how to implement User Defined Role Privilege customization in 8Manage.

3.1.2 Implementation

1) Adding Data into Table ‘func’

8Manage uses a table, **func**, for storing system role privilege information. The table structure of **func** is as follows:

Field Name	Field Type	Allow Null Value	Description
GroupDispOrder	int	No	Function module group display order
GroupName	nvarchar(256)	No	Function module group name. It should be the key value of multilingual information defined by the system. See the example below for the setting of multilingual information.
FuncId	int	No	Primary key, function module id. 8Manage reserves all value smaller than “10000” in FuncID. Please start numbering from 10000 for secondary development to avoid conflicts.
FuncType	varchar(11)	No	Function type. The value can be "VIEW" or "EDIT".
FuncDispOrder	int	No	Function module display order
ResourceName	nvarchar(256)	No	Function module name. It should be the key value of multilingual information defined by the system. See the example below for the setting of multilingual information.

If the secondary development module can be included in the module group of 8Manage, you can use the existing function module group of 8Manage. The display order of the existing function module group name:

GroupDispOrder	GroupName	Description
0	display.menu.search.global	Search
10	display.menu.TOPMENU	Top Menu
15	name.func.topmenu.hr	HCM
20	display.menu.ENTERPRISE	Enterprise
25	display.menu.MEMBERSHIP.MANAGEMENT	Membership
30	display.menu.version.productSystem	Product
40	display.menu.clientBusiness	Client Business
50	display.menu.supplierBusiness	Supplier Business
90	display.menu.PROJECT	Project
120	display.menu.DOCUMENT.LIBRARY	EDMS
160	display.menu.SYSTEMADMIN	Admin

2) Adding Role Privilege Data into Table 'func_url'

8Manage uses table **func_url** to store authorized Request URI in the system.

Table structure of **func_url**:

Field Name	Field Type	Allow Null Value	Description
FuncId	int	No	Function module id, associated with the foreign key of the Func table
ActionName	nvarchar(64)	No	Request URI of the function module. For example, in the Request URL http://cary:8080/8thManage/projectManagement/deal.do?method=listStrategy , the Request URI is /projectManagement/deal.do
MethodName	nvarchar(64)	No	The method name of the function module. As in the above Request URL, MethodName is listStrategy.

3.1.3 Implementation Examples

Goal: Add the access right My Purchase Request before New Purchase Request in the Supplier Business access right group.

Action	Status	Enabled
Demand	✓	<input checked="" type="checkbox"/>
Receipt	✓	<input checked="" type="checkbox"/>
Add Quota Arrangement		<input checked="" type="checkbox"/>
Quota Arrangement	✓	<input checked="" type="checkbox"/>
New Purchase Request		<input checked="" type="checkbox"/>
Delete Purchase Request		<input checked="" type="checkbox"/>
Purchase Request	✓	<input checked="" type="checkbox"/>
New Quick Quote		<input checked="" type="checkbox"/>

1. Adding module privilege data into the **func** table:

```
INSERT INTO func (GroupDispOrder, GroupName, FuncId, FuncType, FuncDispOrder,
ResourceName) VALUES (50, 'display.menu.supplierBusiness', 10000, 'VIEW', 19198,
'project.label.purchasePlan.newPurchasePlan.my');
```

```
INSERT INTO func (GroupDispOrder, GroupName, FuncId, FuncType, FuncDispOrder,
ResourceName) VALUES (50, 'display.menu.supplierBusiness', 10001, 'EDIT', 19199,
'project.label.purchasePlan.newPurchasePlan.my');
```

You can query the FuncDispOrder of “Supplier Business” and “New Purchase Request” in table **func** using the following SQL:

```
select * from func where funcId = 148
```

FuncDispOrder of the newly added “My Purchase Request” role privilege data must be between these two values shown in the SQL above.

Please remember to add a key-value entry as show below in all resource files under `$(siteHome)customize\resources`.

```
project.label.purchasePlan.newPurchasePlan.my=My Purchase Request (Please translate this value
for different languages)
```


2. Adding module privilege data into **func_url** table:

```
INSERT INTO func_url (FuncId, ActionName, MethodName) VALUES (10000,
'projectManagement/myPurchasePlan.do', 'doShowList');
```

The resulting customized role privilege page looks like below:

Action	Enabled	Checked
Demand	✓	☑
Receipt	✓	☑
Add Quota Arrangement		☑
Quota Arrangement	✓	✓
My Purchase Request	☐	☐
New Purchase Request		☑
Delete Purchase Request		☑
Purchase Request	✓	✓

3.1.4 Special Notes

8Manage reserves all value smaller than “10000” in FuncID. Please start the numbering from 10000 for secondary development to avoid conflicts.

3.2 Request Callback Mechanism

3.2.1 Summary

Every action (click) in 8Manage can be customized. For example, the customized fields added by client can be saved as part of the database transaction when the page is saved. Client can also modify the List page to display the customized fields.

This section introduces how to implement customized request callback in 8Manage.

3.2.2 Customize Callback Mechanism

Configuration file CustomizedCallback.xml is the key to implementing customized callback. You can locate this file under `/${siteHome}/customize/callback`, in which `/${siteHome}` represent the main directory of 8Manage installation. The example below shows the syntax and format of configuration file CustomizedCallback.xml.

```

<?xml version="1.0" encoding="UTF-8" ?>
<customized>
  <actions>
    <action actionPoint="actionMethod|methodName" location="pre|post"
      sameTransaction="true|false" useForward="true|false"
      actionClass="CustomizedActionClass" actionMethod="CustomizedMethod" />
    <action actionPoint="actionMethod|methodName" location="pre|post"
      sameTransaction="true|false" useForward="true|false"
      actionClass="CustomizedActionClass" actionMethod="CustomizedMethod" />
  </actions>
</customized>

```

- **customized**

“customized” is the root node of configuration file CustomizedCallback.xml and this node MUST be present.

- **actions**

“actions” node is the direct child of root note “customized”, representing the configuration of “action” for customized callback. This is also a required node.

- **action**

“action” node is child node of “actions”, representing a single customized callback configuration. The “action” node contains the following child nodes as its properties: actionPoint、location、sameTransaction、useForward、actionClass and actionMethod. Description of each of these properties are highlighted below. This is a required node.

- **actionPoint**

“actionPoint” property is the trigger of customized callback, representing a certain method used in 8Manage runtime environment. That is to say, the customized callback will be executed whenever the method specified in “actionPoint” is executed. This is a required node.

Please refer to Commonly Used Methods for more detail.

- **location**

“location” property represents entry point of customized callback. It has two values: “pre” or “post”.

“pre” -- means customized callback will be executed before the “actionPoint” is executed.

“post” -- means customized callback will be executed after the “actionPoint” is executed.

This is a required node.

- **sameTransaction**

“sameTransaction” property represents whether or not the customized callback belongs to the same database transaction as the “actionPoint”. It has 2 values: “true” or “false”.

1. “true” -- means customized callback belong to the same database transaction as the “actionPoint”. In this case, error occurs during the execution of either the customized callback or the “actionPoint” will cause the entire transaction to roll-back.
2. “false” -- means customized callback belong to a different database transaction as the “actionPoint”. In this case, any database transaction error occurs in either executing the customized callback or the “actionPoint” will not interfere with each other.

8Manage treats the following execution scenarios as errors:

1. Not yet captured exception thrown to the calling method;
2. Return “true” when “CommonUtil.hasActionErrors(request)” is called.

Note:

If “sameTransaction” returns “true”, meaning user wants entry point “actionPoint” and customized callback in the same database transaction, then customized callback must use the method below to operate database:

Use the Session object that “SessionUtil.currentSession()” returns to Hibernate to operate database;

Or, use JDBC connection returned from “SessionUtil.currentSession().connection()” to operate database. When doing so, please make sure this code is encapsulated by the transaction.

User defined JDBC connection used in customized callback will not be in the same database transaction as “actionPoint”. In this case, user needs to manage the database transaction during the callback.

This node is a required node.

- **useForward**

“useForward” property is used to determined whether the returned page should be re-assigned as the page specified in the customized callback. It has 2 values: “true” or “false”.

1. “true” -- Use the return page specified in the customized callback as the final return page.
2. “false” -- Retain the return page specified in the entry point “actionPoint” as the final return page.

This is a required node.

- **actionClass**

“actionClass” property is the specific class for customized callback. This class must contain implementation of “com.gearteks.test.customized.Callbackable” interface.

This is a required node.

- **actionMethod**

“actionMethod” property is the specific method for customized callback. Parameters of this method must be serialized as follows:

```
ActionMapping mapping
ActionForm form
HttpServletRequest request
HttpServletResponse response
```

This is a required node.

3.2.3 Commonly Used Methods

3.2.3.1 Error Message

3.2.3.1.1 Detect Error Message in HTTP Request Context

Please call the following method using “CommonUtil”:

```
public static boolean hasActionErrors(HttpServletRequest request)
```

3.2.3.1.2 Save Error Message to HTTP Request Context

Please call the following method using “CommonUtil”:

```
public static void saveErrors(HttpServletRequest request, ProcessErrors pes)
public static void saveErrors(HttpServletRequest request, ActionErrors ae)
public static void saveErrors(HttpServletRequest request, ActionMessages errors)
```

3.2.3.2 Retrieve Current User

Please call the following method using “SessionUtil”:

```
public static User getCurrentUser(HttpServletRequest request)
```

3.2.3.3 Retrieve Current Time

Please call the following method using “DateUtil”:

```
public static Date getDBCurentDate()
public static Timestamp getDBCurentTimestamp()
```

3.2.3.4 Retrieve Hibernate Session

Please call the following method using “SessionUtil”:

```
public static Session currentSession() throws HibernateException
```

3.2.3.5 Retrieve Page Privilege Information

You can retrieve page privilege information by the following way:

```
boolean readonly = StringUtil.toBoolean (request.getAttribute(HtmlTagHelper.USE_POLICY_KEY),
false);
```

If “readonly” is “true”, then current page privilege is read only, or it is editable.

3.2.4 Implementation Examples

3.2.4.1 Add Fields to List Page of Business Object

3.2.4.1.1 Adding “Approver” on Purchase Request List Page

Suppose we have a “List” page look like this:

Name	Requestor	Owners	Product	Purchase Amount	Age	
大沙发	1 thomaslo	1 thomaslo	2	(HKD) 200.00	14 Days	✕

1 purchase request found. [Display Column Settings](#)

激活 Windows
转到“设置”以激活 Windows。

We need to display “Approver” after the “Age” field:

Name	Requestor	Owners	Product	Purchase Amount	Age	Approvers
大沙发	1 thomaslo	1 thomaslo	2	(HKD) 200.00	14 Days	admin

1 purchase request found. [Display Column Settings](#)

激活 Windows
转到“设置”以激活 Windows。

Steps to achieve this customization:

1. Create class “CustomizedOrderAction”, with the “Callbackable” interface for this class:

```
package com.gearteks.test.customized;

import java.util.Collection;
import java.util.HashMap;
import java.util.Map;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;

import com.gearteks.framework.model.approval.Approval;
import com.gearteks.framework.process.assist.DisplayObjAssistProcess;
import com.gearteks.imatech.model.projmgmt.PurchasePlan;
import com.gearteks.imatech.model.quotation.Quotation;
import com.wisagetech.framework.callback.Callbackable;

public class CustomizedPurchasePlanAction implements Callbackable {
    /**
     *
     * @param mapping
     * @param form
     * @param request
     * @param response
     * @return
     * @throws Exception
     */
    public ActionForward doShowList(ActionMapping mapping, ActionForm form, HttpServletRequest
request,
    HttpServletResponse response) throws Exception {
        Collection<PurchasePlan> planLst = (Collection<PurchasePlan>)
request.getAttribute("planCol");
        if (planLst != null && !planLst.isEmpty()) {
```

```

Map approverMap = new HashMap();
for (PurchasePlan plan : planLst) {
    Approval appr = plan.getApproval();
    approverMap.put(plan.getPlanId(),
        DisplayObjAssistProcess.getDefaultToolTips(appr.getApprovers(), request));
}
request.setAttribute("approverMap", approverMap);
}
return null;
}
}

```

2. To display “Approver” in “purchasePlanListTable.jsp”, retrieve the “approverMap” set:

```
java.util.Map approverMap = (java.util.Map)request.getAttribute("approverMap");
```

- and add display approver after the “Responsible Person” column on List page:

```

<gearTable:column titleKey=" name.approval.approver.s" sortable="false" headerClass="sortable" >
<%= approverMap.get(((Quotation)quotation).getQuotationId()) != null ?
approverMap.get(((Quotation)quotation).getQuotationId()) :
HtmlTagHelper.EMPTY_DISPLAY_VALUE %>
</gearTable:column>

```

3. Finally, add the following configuration into `/${siteHome}/customize/callback/CustomizedCallback.xml`

```

<action actionPoint="quotation|doShowList" location="post"
    sameTransaction="true" useForward="false"
actionClass="com.gearteks.test.customized.CustomizedOrderAction" actionMethod="doShowList" />

```

4. Re-start 8Manage and check your customization.

3.2.4.1.2 Adding Other Fields on List Page of Campaign

Class creation is similar to Adding “Approver” on List Page. Use the method below to retrieve the Campaign list:

```
Collection campaignLst = (Collection)request.getAttribute(Constants.LIST);
```

Add configuration into `/${siteHome}/customize/callback/CustomizedCallback.xml` with code similar to this:

```
<action actionPoint="campaign|listStrategy" location="post"
        sameTransaction="true" useForward="false"
actionClass="com.gearteks.test.customized.CustomizedCampaignAction" actionMethod="listStrategy"
/>
```

3.2.4.1.3 Adding Other Fields on List Page of Opportunity/Strategy

Class creation is similar to Adding “Approver” on List Page. Use the method below to retrieve the Opportunity/Strategy list:

```
Collection strategyLst = (Collection)request.getAttribute(Constants.LIST);
```

Add configuration into `/${siteHome}/customize/callback/CustomizedCallback.xml` similar to this:

```
<action actionPoint="strategy|listStrategy" location="post"
        sameTransaction="true" useForward="false"
actionClass="com.gearteks.test.customized.CustomizedStrategyAction" actionMethod="listStrategy" />
```

3.2.4.1.4 Adding Other Fields on List Page of Contract

Class creation is similar to Adding “Approver” on List Page. Use the method below to retrieve the Contract list:

```
Collection contractLst = (Collection)request.getAttribute(Constants.LIST);
```

Add configuration into `/${siteHome}/customize/callback/CustomizedCallback.xml` similar to this:

```
<action actionPoint="deal|listStrategy" location="post"
        sameTransaction="true" useForward="false"
actionClass="com.gearteks.test.customized.CustomizedDealAction" actionMethod="listStrategy" />
```

3.2.4.1.5 Adding Other Fields on List Page of Internal Human Resource

Class creation is similar to Adding “Approver” on List Page. Use the method below to retrieve the

Internal Human Resource list:

```
Collection userLst = (Collection)request.getAttribute(Constants. KEY_USERLIST);
```

Add configuration to `/${siteHome}/customize/callback/CustomizedCallback.xml` with code similar to this:

```
<action actionPoint="user|doShowList" location="post"
        sameTransaction="true" useForward="false"
        actionClass="com.gearteks.test.customized.CustomizedUserAction" actionMethod="doShowList" />
```

3.2.4.1.6 Adding Other Fields on List Page of Client

Class creation is similar to Adding “Approver” on List Page. Use the method below to retrieve the Client list:

```
Collection clientLst = (Collection)request.getAttribute(Constants. KEY_COMPANYLIST);
```

Add configuration into `/${siteHome}/customize/callback/CustomizedCallback.xml` similar to this:

```
<action actionPoint="company|doShowList" location="post"
        sameTransaction="true" useForward="false"
        actionClass="com.gearteks.test.customized.CustomizedCompanyAction" actionMethod="doShowList"
/>
```

3.2.4.2 Add Fields to Basic Page of Business Object

3.2.4.2.1 Add Supplier Field to Basic Page of Client Contract

Example below shows how a new field “Supplier” is added to the Basic page of a client Contract.

Before adding the “Supplier” field, the Contract Basic page has a “Proposed Client Contract” looks like figure below:

After the field “Supplier” is added, the same page looks like this:

The screenshot shows a web form titled "Proposed Client Contract" with the following fields and values:

- Name: test contract
- ID: 1001
- Owners: admin
- Sales Reps: admin
- Approver List: User Defined
- Assisting Groups: (empty)
- Client: XXX客户
- Expected Start Date: 2024-10-07
- Estimated Signed Date: 2025-01-05
- Currency: HKD
- Status: Not Yet Submitted For Internal Approval
- Supplier: XXX supplier (highlighted with a red box)
- Owner Groups: Engineering
- Legal Entity: Corporate
- Approvers: admin
- Assisting Persons: (empty)
- Contacts: (empty)
- Expected Finish Date: 2025-01-05
- In The Funnel: 0% Confidence

At the bottom left, there is a "Submit Approval:" label followed by a "Submit" button. At the bottom right, there is a Windows watermark: "激活 Windows 转到“设置”以激活 Windows。"

The steps required to implement the customization shown in example above are as follows:

1. Create the table for storing supplier information on the database server (“carson”, in this example).

```
CREATE TABLE customized_members (  
    ObjId          id NOT NULL,  
    ObjType        type NOT NULL,  
    memberType     type NOT NULL,  
    memberObjId    id NOT NULL,  
    memberObjType  type NOT NULL  
);  
go  
  
ALTER TABLE customized_members ADD  
    CONSTRAINT PK_customized_members PRIMARY KEY CLUSTERED (ObjId, ObjType,  
memberType, memberObjId, memberObjType);  
go  
  
ALTER TABLE members ADD  
    CONSTRAINT CK1_customized_members  
    CHECK (ObjType in ('Deal'));  
go  
  
ALTER TABLE customized_members ADD  
    CONSTRAINT CK2_customized_members  
    CHECK (memberObjType in ('Company', 'Party', 'User'));  
go
```

2. Create class CustomizedMembers

```
/**  
 *  
 */  
package com.gearteks.test.customized.model;  
  
import java.io.Serializable;  
  
import org.apache.commons.lang.builder.EqualsBuilder;  
import org.apache.commons.lang.builder.HashCodeBuilder;
```

```
import com.gearteks.framework.common.Constants;
import com.gearteks.framework.common.exception.GTSystemException;
import com.gearteks.framework.model.assist.DisplayObjInterFace;
import com.gearteks.framework.object.BusinessObject;
import com.gearteks.framework.process.CompanyProcess;
import com.gearteks.framework.process.UserProcess;

/**
 * @author ericchchen
 *
 */
public class CustomizedMembers implements Comparable, Serializable, DisplayObjInterFace {
    private CustomizedMembersPK _comp_id;

    public static class CustomizedMembersPK implements Serializable {
        private Integer _objId;
        private String _objType;
        private String _memberType;
        private Integer _memberObjId;
        private String _memberObjType;

        public CustomizedMembersPK() {
            ;
        }

        public CustomizedMembersPK(Integer objId, String objType, String memberType, Integer
memberObjId, String memberObjType) {
            this._objId = objId;
            this._objType = objType;
            this._memberType = memberType;
            this._memberObjId = memberObjId;
            this._memberObjType = memberObjType;
        }

        /* equals/hashCode */
        public boolean equals(Object other) {
            if (this == other) {
                return true;
            }
        }
    }
}
```

```
    }
    if (other == null) {
        return false;
    }
    if (!(other instanceof CustomizedMembersPK)) {
        return false;
    }
    CustomizedMembersPK castOther = (CustomizedMembersPK) other;

    return new EqualsBuilder().append(this.getObjId(), castOther.getObjId())
        .append(this.getObjType(), castOther.getObjType())
        .append(this.getMemberType(), castOther.getMemberType())
        .append(this.getMemberObjId(), castOther.getMemberObjId())
        .append(this.getMemberObjType(), castOther.getMemberObjType())
        .isEquals();
}

public int hashCode() {
    return new HashCodeBuilder().append(this.getObjId())
        .append(this.getObjType())
        .append(this.getMemberType())
        .append(this.getMemberObjId())
        .append(this.getMemberObjType())
        .toHashCode();
}

/* Getter/Setter */
public Integer getMemberObjId() {
    return _memberObjId;
}

public void setMemberObjId(Integer memberObjId) {
    _memberObjId = memberObjId;
}

public String getMemberObjType() {
    return _memberObjType;
}

public void setMemberObjType(String memberObjType) {
    _memberObjType = memberObjType;
}
```

```
}
public String getMemberType() {
    return _memberType;
}
public void setMemberType(String memberType) {
    _memberType = memberType;
}
public Integer getObjId() {
    return _objId;
}
public void setObjId(Integer objId) {
    _objId = objId;
}
public String getObjType() {
    return _objType;
}
public void setObjType(String objType) {
    _objType = objType;
}
}

public CustomizedMembers() {
    ;
}
public CustomizedMembers(CustomizedMembersPK comp_id) {
    this._comp_id = comp_id;
}
public CustomizedMembers(Integer objId, String objType, String memberType, Integer
memberObjId, String memberObjType) {
    CustomizedMembersPK comp_id = new CustomizedMembersPK(objId, objType, memberType,
memberObjId, memberObjType);
    this.setComp_id(comp_id);
}
/* (non-Javadoc)
 * @see java.lang.Comparable#compareTo(java.lang.Object)
 */
public int compareTo(Object o) {
```

```

try {
    BusinessObject otherMem = ((CustomizedMembers)o).getDisplayObj();
    BusinessObject thisMem = this.getDisplayObj();
    if (thisMem instanceof Comparable && otherMem instanceof Comparable) {
        return ((Comparable)thisMem).compareTo(((Comparable)otherMem));
    }
    return thisMem.getName().compareTo(otherMem.getName());
} catch (Exception e) {
    return 0;
}
}

public BusinessObject getDisplayObj() throws GTSystemException {
    Integer thisId = this.getComp_id().getMemberObjId();
    String type = this.getComp_id().getMemberObjType();

    if (type.equals(Constants.OBJ_TYPE_COMPANY)) {
        return CompanyProcess.getInstance().getCompany(thisId);
    } else if (type.equals(Constants.OBJ_TYPE_USER)) {
        return UserProcess.getInstance().getUser(thisId);
    } else {
        return null;
    }
}

/* (non-Javadoc)
 *
 * @see
com.gearteks.framework.model.assist.DisplayObjInterFace#setDisplayType(java.lang.String)
 */
public void setDisplayType(String displayType) {
    ;
}

/**
 * @return the comp_id
 */
public CustomizedMembersPK getComp_id() {
    return _comp_id;
}

/**
 * @param comp_id the comp_id to set

```

```

    */
    public void setComp_id(CustomizedMembersPK comp_id) {
        _comp_id = comp_id;
    }
    /* (non-Javadoc)
    * @see java.lang.Object#hashCode()
    */
    @Override
    public int hashCode() {
        return new HashCodeBuilder().append(this.getComp_id()).toHashCode();
    }
    /* (non-Javadoc)
    * @see java.lang.Object#equals(java.lang.Object)
    */
    @Override
    public boolean equals(Object obj) {
        if (this == obj) {
            return true;
        }
        if (obj == null) {
            return false;
        }
        if (!(obj instanceof CustomizedMembers)) {
            return false;
        }
        CustomizedMembers castOther = (CustomizedMembers) obj;

        return new EqualsBuilder().append(this.getComp_id(), castOther.getComp_id()).isEquals();
    }
}

```

3. The Hibernate mapping file for creating class CustomizedMembers

```

<?xml version="1.0"?>
<!DOCTYPE hibernate-mapping PUBLIC "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
"http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
<!-- Generated 2007-9-12 9:48:14 by Hibernate Tools 3.2.0.b9 -->

```



```

<hibernate-mapping>
  <class name="com.gearteks.test.customized.model.CustomizedMembers"
table="customized_members">
    <composite-id name="comp_id"
class="com.gearteks.test.customized.model.CustomizedMembers$CustomizedMembersPK">
        <key-property name="objId" type="integer">
            <column name="objId" />
        </key-property>
        <key-property name="objType" type="string">
            <column name="ObjType" length="11" />
        </key-property>
        <key-property name="memberType" type="string">
            <column name="memberType" length="11" />
        </key-property>
        <key-property name="memberObjId" type="integer">
            <column name="memberObjId" />

```

4. Add configuration to `$(8thManageHome)/WEB-INF/classes/hibernate/hibernate.cfg.xml` similar to the example below:

```

</key-property>
<key-property name="memberObjType" type="string">
    <column name="memberObjType" length="11" />

```

5. Create class `CustomizedContractAction`, with the following `Callbackable` interface:

```

    </key-property>
  </composite-id>
</class>
</hibernate-mapping>

<mapping resource="hibernate/CustomizedMembers.hbm.xml" />

/**
 *
 */
package com.gearteks.test.customized;

```

```
import java.util.ArrayList;
import java.util.Collection;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;

import com.gearteks.framework.common.Constants;
import com.gearteks.framework.common.exception.GTSystemException;
import com.gearteks.framework.common.util.CommonUtil;
import com.gearteks.framework.common.util.SessionUtil;
import com.gearteks.framework.model.Company;
import com.gearteks.framework.model.Party;
import com.gearteks.framework.model.User;
import com.gearteks.framework.object.BusinessObject;
import com.gearteks.framework.process.ProcessErrors;
import com.gearteks.framework.process.assist.MatchObjectAssist;
import com.gearteks.imatech.controller.projmgmt.StrategyForm;
import com.gearteks.imatech.model.projmgmt.Strategy;
import com.gearteks.test.customized.model.CustomizedMembers;
import com.gearteks.test.customized.util.DbUtil;
import com.wisagetech.framework.callback.Callbackable;

/**
 * @author ericchenn
 *
 */
public class CustomizedContractAction implements Callbackable {
    /**
     *
     * @param mapping
     * @param form
     * @param request
     * @param response
     */
}
```

```

* @return
* @throws Exception
*/
public ActionForward viewStrategy(ActionMapping mapping, ActionForm form, HttpServletRequest
request, HttpServletResponse response)
-----
throws Exception {
    StrategyForm strategyForm = (StrategyForm)form;

    CustomizedMembers oldMember = this.getMember(strategyForm.getStrategyId(),
strategyForm.getObjType());
    if (oldMember != null) {
        BusinessObject bo = oldMember.getDisplayObj();
        if (bo != null) {
            request.setAttribute("suppliers", bo.getName());
        }
    }

    return null;
}
/**
*
* @param mapping
* @param form
* @param request
* @param response
* @return
* @throws Exception
*/
public ActionForward saveStrategy(ActionMapping mapping, ActionForm form, HttpServletRequest
request, HttpServletResponse response)
throws Exception {
    String suppliers = request.getParameter("suppliers");
    request.setAttribute("suppliers", suppliers);

    if (CommonUtil.hasActionErrors(request)) {
        return null;
    }
}

```

```

Strategy strategy = this.getStrategy(form, request);
if (strategy == null) {
    return null;
}

ProcessErrors errors = new ProcessErrors();
if (suppliers != null && !suppliers.trim().equals("")) {
    BusinessObject supplier =
(BusinessObject)MatchObjectAssist.matchCompanyByType(suppliers,
Company.COMPANY_TYPE_SUPPLIER, errors, null,
SessionUtil.getMessage("quotation.label.supplier"),
SessionUtil.getMessage("quotation.label.supplier"));
    if (supplier != null) {
        CustomizedMembers oldMember = this.getMember(strategy.getId(),
strategy.getObjType());
        CustomizedMembers newMember = this.toMember("Supplier", supplier, strategy);
        this.saveCustomizedMembers(oldMember, newMember);
    }
}

if (errors.isHaveErrors()) {
    CommonUtil.saveErrors(request, errors);
}

return null;
}
/**
 *
 * @param form
 * @param request
 * @return
 * @throws GTSYSTEMException
 */
private Strategy getStrategy(ActionForm form, HttpServletRequest request) throws
GTSYSTEMException {
    StrategyForm strategyForm = (StrategyForm)form;
    Strategy strategy = (Strategy)request.getAttribute("strategy");
    if (strategy == null) {

```

```

        strategy = (Strategy)DbUtil.find(Strategy.class, strategyForm.getStrategyId(), true);
    }
    return strategy;
}
/**
 *
 * @param memberType
 * @param bo
 * @return
 */
private CustomizedMembers toMember(String memberType, BusinessObject bo, Strategy strategy)
{
    String memberObjType = null;
    if (bo instanceof User) {
        memberObjType = Constants.OBJ_TYPE_USER;
    } else if (bo instanceof Party) {
        memberObjType = Constants.OBJ_TYPE_PARTY;
    } else if (bo instanceof Company) {
        memberObjType = Constants.OBJ_TYPE_COMPANY;
    }
    if (memberObjType != null) {
        return new CustomizedMembers(strategy.getId(), strategy.getObjType(), memberType,
bo.getId(), memberObjType);
    } else {
        return null;
    }
}
/**
 *
 * @param objId
 * @param objType
 * @return
 * @throws GTSYSTEMException
 */
private CustomizedMembers getMember(Integer objId, String objType) throws GTSYSTEMException
{
    Collection<CustomizedMembers> col = this.getMembers(objId, objType);
    if (col != null && !col.isEmpty()) {

```

```

        return col.iterator().next();
    }
    return null;
}
/**
 *
 * @param objId
 * @param objType
 * @return
 * @throws GTSystemException
 */
private Collection<CustomizedMembers> getMembers(Integer objId, String objType) throws
GTSystemException {
    return this.getMembers(objId, objType, null, null, null);
}
/**
 *
 * @param objId
 * @param objType
 * @param memberType
 * @param memberObjId
 * @param memberObjType
 * @return
 * @throws GTSystemException
 */
private Collection<CustomizedMembers> getMembers(Integer objId, String objType, String
memberType, Integer memberObjId, String memberObjType) throws GTSystemException {
    StringBuffer sb = new StringBuffer();
    sb.append("select m from CustomizedMembers m where 2 > 1 ");

    ArrayList<Object> paramList = new ArrayList<Object>();
    if (objId != null) {
        sb.append(" and m.comp_id.objId = ? ");
        paramList.add(objId);
    }
    if (objType != null && !"".equals(objType.trim())) {
        sb.append(" and m.comp_id.objType = ? ");
        paramList.add(objType);
    }
}

```

```

    }
    if (memberType != null && !"".equals(memberType.trim())) {
        sb.append(" and m.comp_id.memberType = ? ");
        paramList.add(memberType);
    }
    if (memberObjId != null) {
        sb.append(" and m.comp_id.memberObjId = ? ");
        paramList.add(memberObjId);
    }
    if (memberObjType != null && !"".equals(memberObjType.trim())) {
        sb.append(" and m.comp_id.memberObjType = ? ");
        paramList.add(memberObjType);
    }

    return DbUtil.getQueryData(sb.toString(), paramList);
}
/**
 *
 * @param oldMember
 * @param newMember
 * @throws GTSYSTEMException
 */
private void saveCustomizedMembers(CustomizedMembers oldMember, CustomizedMembers
newMember) throws GTSYSTEMException {
    ArrayList<CustomizedMembers> oldCol = new ArrayList<CustomizedMembers>();
    if (oldMember != null) {
        oldCol.add(oldMember);
    }
    ArrayList<CustomizedMembers> newCol = new ArrayList<CustomizedMembers>();
    if (newMember != null) {
        newCol.add(newMember);
    }
    this.saveCustomizedMembers(oldCol, newCol);
}
/**
 *
 * @param oldCol
 * @param newCol

```

```

* @return
* @throws GTSystemException
*/
private boolean saveCustomizedMembers(Collection<CustomizedMembers> oldCol,
Collection<CustomizedMembers> newCol) throws GTSystemException {
    if (newCol == null) {
        return false;
    }
    boolean changed = false;
    if (oldCol == null || oldCol.isEmpty()) {
        for (CustomizedMembers newMember : newCol) {
            DbUtil.add(newMember);
            changed = true;
        }
    } else {
        for (CustomizedMembers newMember : newCol) {
            if (!oldCol.contains(newMember)) {
                DbUtil.add(newMember);
                changed = true;
            }
        }
        for (CustomizedMembers oldMember : oldCol) {
            if (!newCol.contains(oldMember)) {
                DbUtil.delete(oldMember);
                changed = true;
            }
        }
    }
    return changed;
}
}

```

6. Show "Supplier" field in strategy.jsp:

```

<td width="130" class="fieldName">
    <%=SessionUtil.getMessage(request, "quotation.label.supplier")%>:
</td>
<td width="3">&nbsp;&nbsp;&nbsp;</td>

```



```

<td>
<% if (!readonly) { %>
<input type="text" name="suppliers" id="suppliers" class="text"
value="<%=StringUtil.isEmpty(request.getAttribute("suppliers")) ? "" :
request.getAttribute("suppliers")%>" />
',true,'true', null, null,
'companyType=Supplier&amp;hideCompanyType=true');return false;"
title="<%=SessionUtil.getMessage(request, "prompt.title.companySelect.company")%>" />
<% } else { %>
<%=StringUtil.isEmpty(request.getAttribute("suppliers")) ? "" : request.getAttribute("suppliers")%>
<input type="hidden" name="suppliers" id="suppliers"
value="<%=StringUtil.isEmpty(request.getAttribute("suppliers")) ? "" :
request.getAttribute("suppliers")%>" />
<% } %>
</td>

```

Note: The root of a web application (as in `request.getContextPath()`) is appended to custom menu/field URL automatically by the system, and thus no need to append it manually.

7. Add the following JavaScript method into `strategyJavaScript.jsp`

```

function setSupplier(objId, objName, otherParam, type, contact) {
    if (objId == "" || objName == "") {
        return;
    }
    var oSupplier = document.getElementById("suppliers");
    oSupplier.value = objName;
    return false;
}

```

8. Create a utility class for database operation:

```

/**
 *
 */
package com.gearteks.test.customized.util;

```

```
import java.io.Serializable;
import java.util.Collection;
import java.util.Iterator;

import org.hibernate.HibernateException;
import org.hibernate.Query;
import org.hibernate.Session;

import com.gearteks.framework.common.exception.GTObjectNotFoundException;
import com.gearteks.framework.common.exception.GTSystemException;
import com.gearteks.framework.common.util.SessionUtil;

/**
 * @author ericchenn
 *
 */
public abstract class DbUtil {
    /**
     *
     * @param currClass
     * @param pk
     * @param throwException
     * @return
     * @throws GTSystemException
     */
    public static Object find(Class currClass, Serializable pk, boolean throwException)
        throws GTSystemException {
        try {
            Session session = SessionUtil.currentSession();
            Object ug = session.get(currClass, pk);

            if (ug == null) {
                if (throwException) {
                    throw new GTObjectNotFoundException(currClass, pk);
                }
            }
        }
    }
}
```

```
        return ug;
    } catch (HibernateException e) {
        throw new GTSystemException(e);
    }
}
/**
 *
 * @param obj
 * @return
 * @throws GTSystemException
 */
public static boolean add(Object obj) throws GTSystemException {
    try {
        Session session = SessionUtil.currentSession();
        session.save(obj);
        return true;
    } catch (HibernateException e) {
        throw new GTSystemException(e);
    }
}
/**
 *
 * @return
 */
public static boolean update(Object obj) throws GTSystemException {
    try {
        Session session = SessionUtil.currentSession();
        session.update(obj);
        return true;
    } catch (HibernateException e) {
        throw new GTSystemException(e);
    }
}
/**
 *
 * @param obj
 * @return
 * @throws GTSystemException
```

```
*/
public static boolean delete(Object obj) throws GTSYSTEMException {
    try {
        Session session = SessionUtil.currentSession();
        session.delete(obj);
        return true;
    } catch (HibernateException e) {
        throw new GTSYSTEMException(e);
    }
}
/**
 *
 * @param obj
 * @return
 * @throws GTSYSTEMException
 */
public static boolean saveOrUpdate(Object obj) throws GTSYSTEMException {
    try {
        Session session = SessionUtil.currentSession();
        session.saveOrUpdate(obj);
        return true;
    } catch (HibernateException e) {
        throw new GTSYSTEMException(e);
    }
}
/**
 *
 * @param sql
 * @param paramLst
 * @return
 * @throws GTSYSTEMException
 */
public static Collection getQueryData(String hql, Collection paramLst) throws GTSYSTEMException {
    return getQueryData(hql, paramLst, false);
}
/**
 *
 * @param sql
```

```

* @param list
* @param useQueryCache
* @return
* @throws GTSystemException
*/
public static Collection getQueryData(String hql, Collection paramLst, boolean useQueryCache)
throws GTSystemException {
    if (hql == null || hql.trim().equals("") || paramLst == null) {
        throw new IllegalArgumentException("Null name or empty name is found, cannot perform " +
"getNamedQueryData(String s) function.");
    }

    try {
        Session session = SessionUtil.currentSession();
        Query query = session.createQuery(hql);
        if (useQueryCache) {
            query.setCacheable(true);
        }

        int i = 0;
        for (Iterator it = paramLst.iterator(); it.hasNext();) {
            query.setParameter(i++, it.next());
        }
        return query.list();
    } catch (HibernateException e) {
        throw new GTSystemException(e);
    }
}
}

```

9. Add the following configuration to `/${siteHome}/customize/callback/CustomizedCallback.xml`

```

<action actionPoint="deal|viewStrategy" location="post"
    sameTransaction="true" useForward="false"
    actionClass="com.gearteks.test.customized.CustomizedContractAction"
actionMethod="viewStrategy" />

<action actionPoint="deal|saveStrategy" location="post"

```

```

sameTransaction="true" useForward="false"
actionClass="com.gearteks.test.customized.CustomizedContractAction"
actionMethod="saveStrategy" />

```

10. Restart 8Manage service

3.2.4.2.2 Add Other Fields to Basic Page of Campaign

This is a customization similar to “Add Supplier Field to Basic Page of Client Contract”, only implementation special to this operation is listed below:

Use the method below to retrieve Campaign object:

```

/**
 *
 * @param form
 * @param request
 * @return
 * @throws GTSYSTEMException
 */
private Strategy getStrategy(ActionForm form, HttpServletRequest request) throws
GTSYSTEMException {
    StrategyForm strategyForm = (StrategyForm)form;
    Strategy strategy = (Strategy)request.getAttribute("strategy");
    if (strategy == null) {
        strategy = (Strategy)DbUtil.find(Strategy.class, strategyForm.getStrategyId(), true);
    }
    return strategy;
}

```

Then, add the following configuration to `$(siteHome)/customize/callback/CustomizedCallback.xml`

```

<action actionPoint="campaign|viewStrategy" location="post"
    sameTransaction="true" useForward="false"
    actionClass="com.gearteks.test.customized.CustomizedCampaignAction"
    actionMethod="viewStrategy" />

<action actionPoint="campaign|saveStrategy" location="post"

```

```

sameTransaction="true" useForward="false"
actionClass="com.gearteks.test.customized.CustomizedCampaignAction"
actionMethod="saveStrategy" />

```

3.2.4.2.3 Add Other Fields to Basic Page of Opportunity/Strategy

This is a customization similar to “Add Supplier Field to Basic Page of Client Contract”, only implementation special to this operation is listed below:

Use the method below to retrieve Opportunity/Strategy object:

```

/**
 *
 * @param form
 * @param request
 * @return
 * @throws GTSYSTEMException
 */
private Strategy getStrategy(ActionForm form, HttpServletRequest request) throws
GTSYSTEMException {
    StrategyForm strategyForm = (StrategyForm)form;
    Strategy strategy = (Strategy)request.getAttribute("strategy");
    if (strategy == null) {
        strategy = (Strategy)DbUtil.find(Strategy.class, strategyForm.getStrategyId(), true);
    }
    return strategy;
}

```

Then add the following configuration to `/${siteHome}/customize/callback/CustomizedCallback.xml`

```

<action actionPoint="strategy|viewStrategy" location="post"
    sameTransaction="true" useForward="false"
    actionClass="com.gearteks.test.customized.CustomizedStrategyAction"
    actionMethod="viewStrategy" />

<action actionPoint="strategy|saveStrategy" location="post"
    sameTransaction="true" useForward="false"
    actionClass="com.gearteks.test.customized.CustomizedStrategyAction"

```

```
actionMethod="saveStrategy" />
```

3.2.4.3 Synchronize Information with External Database

3.2.4.3.1 Synchronize User Information with External Database

When creating user with 8Manage, we need to synchronize user information with external database. For example, to the “user” table of database “carson”. This customized callback and entry point “actionPoint” do not need to be in the same database transaction, so that error occurs during synchronization will not affect the use of 8Manage. Please follow steps below:

1. Create new user table in database server, “carson”, for example.

```
CREATE TABLE users (
  Address1          nvarchar(256) NULL,
  CreatorId        int NOT NULL,
  CreatorTS        datetime CONSTRAINT [DF1_users] DEFAULT (getdate()) NOT NULL,
  Description       nvarchar(2000) NULL,
  Email            nvarchar(64) NULL,
  Fax              nvarchar(64) NULL,
  FirstName        nvarchar(64) NOT NULL,
  LastName         nvarchar(64) NOT NULL,
  MiddleName       nvarchar(64) NULL,
  Password         nvarchar(64) NULL,
  Phone            nvarchar(64) NULL,
  Status           varchar(11) NOT NULL,
  TS               datetime CONSTRAINT [DF2_users] DEFAULT (getdate()) NOT NULL,
  TSUserId         int NOT NULL,
  UserId           int NOT NULL,
  UserName         nvarchar(64) NOT NULL,
  Zip              nvarchar(64) NULL,
  Sex              varchar(11) NULL,
  CONSTRAINT [CK1_users] CHECK ([Status] = 'Suspend' or ([Status] = 'Deleted' or [Status] =
'Active'))
);
go

ALTER TABLE users ADD
  CONSTRAINT PK_users PRIMARY KEY CLUSTERED (UserId);
go
```



```
CREATE NONCLUSTERED INDEX AK1_users ON users (UserName);  
go
```

2. Then create class “CustomizedUserAction”, with the following “Callbackable” interface for implementing this class:

```
package com.gearteks.test.customized;  
  
import java.sql.Connection;  
import java.sql.DriverManager;  
import java.sql.SQLException;  
import java.sql.Statement;  
  
import javax.servlet.http.HttpServletRequest;  
import javax.servlet.http.HttpServletResponse;  
  
import org.apache.struts.action.ActionForm;  
import org.apache.struts.action.ActionForward;  
import org.apache.struts.action.ActionMapping;  
  
import com.gearteks.framework.common.util.CommonUtil;  
import com.gearteks.framework.controller.UserForm;  
import com.wisagetech.framework.callback.Callbackable;  
  
/**  
 * @author ericchen  
 *  
 */  
public class CustomizedUserAction implements Callbackable {  
    /**  
     *  
     * @param mapping  
     * @param form  
     * @param request  
     * @param response  
     * @return  
     * @throws Exception  
     */  
}
```

```
public ActionForward doSync(ActionMapping mapping, ActionForm form, HttpServletRequest
request, HttpServletResponse response)
throws Exception {
    UserForm uf = (UserForm)form;

    if (CommonUtil.hasActionErrors(request)) {
        return null;
    }

    Connection conn = null;
    Statement stmt = null;
    try {
        StringBuffer sb = new StringBuffer();
        sb.append(" insert into users (Address1, CreatorId, CreatorTS, Description, Email, Fax,
FirstName, LastName, MiddleName, Password, Phone, Status, TS, TSUserId, UserId, UserName, Zip,
Sex) ")

        .append(" values (")
        .append(" ").append(uf.getAddress1()).append(", ")
        .append(" ").append(uf.getCreator().getUserId()).append(", ")
        .append(" ").append(uf.getCreateTime()).append(", ")
        .append(" ").append(uf.getDescription()).append(", ")
        .append(" ").append(uf.getEmail()).append(", ")
        .append(" ").append(uf.getFax()).append(", ")
        .append(" ").append(uf.getFirstName()).append(", ")
        .append(" ").append(uf.getLastName()).append(", ")
        .append(" ").append(uf.getMiddleName()).append(", ")
        .append(" ").append(uf.getPassword()).append(", ")
        .append(" ").append(uf.getPhone()).append(", ")
        .append(" ").append(uf.getStatus()).append(", ")
        .append(" ").append(uf.getTs()).append(", ")
        .append(" ").append(uf.getTsUser().getUserId()).append(", ")
        .append(" ").append(uf.getUserId()).append(", ")
        .append(" ").append(uf.getUserName()).append(", ")
        .append(" ").append(uf.getPostCode()).append(", ")
        .append(" ").append(uf.getSex()).append(" ")
        .append(") ");

        conn = this.getConnection();
        stmt = conn.createStatement();
```

```

        stmt.execute(sb.toString());
    } finally {
        this.closeQuietly(stmt);
        this.closeQuietly(conn);
    }
    return null;
}
/**
 *
 * @return
 */
private Connection getConnection() {
    String driverClass = "net.sourceforge.jtds.jdbc.Driver";
    String connUrl = "jdbc:jtds:sqlserver://carson:1433/ericchen141;lastUpdateCount=true;instance=";
    String dbUser = "dbuser";
    String dbPwd = "db1234";

    try {
        Class.forName(driverClass);
        return DriverManager.getConnection(connUrl, dbUser, dbPwd);
    } catch (Exception ex) {
        ;
    }
    return null;
}
/**
 *
 * @param stmt
 */
private void closeQuietly(Statement stmt) {
    if (stmt != null) {
        try {
            stmt.close();
        } catch (SQLException e) {
            ;
        }
    }
}

```

```

}
/**
 *
 * @param conn
 */
private void closeQuietly(Connection conn) {
    if (conn != null) {
        try {
            conn.close();
        } catch (SQLException e) {
            ;
        }
    }
}
}
}
}
}

```

3. Add the following configuration into `/${siteHome}/customize/callback/CustomizedCallback.xml`:

```

<action actionPoint="user|doAdd" location="post"
        sameTransaction="false" useForward="false"
        actionClass="com.gearteks.test.customized.CustomizedUserAction"
        actionMethod="doSync" />

```

4. Restart 8Manage service.

3.2.5 Frequently Used 'actionPoint' in 8Manage

ActionName	MethodName	Interceptor Point Description	Additional Parameters?	Sample Code
campaign	listStrategy	Open List page of Campaign	N.	Retrieve Campaign List information in customized callback method: Collection campaignLst = (Collection)request.getAttribute(Constants.LIST);
campaign	viewStrategy	Open a Campaign	N.	Retrieve Campaign object in customized callback method:

ActionName	MethodName	Interceptor Point Description	Additional Parameters?	Sample Code
				Strategy campaign = (Strategy)request.getAttribute("strategy");
campaign	saveStrategy	Save a Campaign	N. "isUpdatedSpecial" will save partial content; "isAddJobStrategy" will add records associated with "JobStrategy"; "msgld" will add records associated with "StrategyObjMember".	Retrieve Campaign in customized callback method: private Strategy getStrategy(ActionForm form, HttpServletRequest request) throws GTSYSTEMException { StrategyForm strategyForm = (StrategyForm)form; Strategy strategy = (Strategy)request.getAttribute("strategy"); if (strategy == null) { strategy = (Strategy)DbUtil.find(Strategy.class, strategyForm.getStrategyId(), true); } return strategy; }
strategy	listStrategy	Open List page of Opportunity/Strategy	N.	Retrieve Opportunity/Strategy List information in customized callback method: Collection strategyLst = (Collection)request.getAttribute(Constants.LIST);
strategy	viewStrategy	Open a Opportunity/Strategy	N.	Retrieve Opportunity/Strategy object in customized callback method: Strategy strategy = (Strategy)request.getAttribute("strategy");
strategy	saveStrategy	Save Opportunity/	N. "isUpdatedSpe	Retrieve Opportunity/Strategy object in customized callback method:

ActionName	MethodName	Interceptor Point Description	Additional Parameters?	Sample Code
		Strategy	<p>cial” will save partial content;</p> <p>“isAddJobStrategy” will add records associated with “JobStrategy”;</p> <p>“msgld” will add records associated with “StrategyObjMember”.</p>	<pre>private Strategy getStrategy(ActionForm form, HttpServletRequest request) throws GTSystemException { StrategyForm strategyForm = (StrategyForm)form; Strategy strategy = (Strategy)request.getAttribute("strate gy"); if (strategy == null) { strategy = (Strategy)DbUtil.find(Strategy.class, strategyForm.getStrategyId(), true); } return strategy; }</pre>
deal	listStrategy	Open Contract List page	N.	Retrieve Contract List information in customized callback method: Collection contractLst = (Collection)request.getAttribute(Constants.LIST);
deal	viewStrategy	Open a Contract	N.	Retrieve Contract object in customized callback method: Strategy contract = (Strategy)request.getAttribute("strate gy");
deal	saveStrategy	Save a Contract	<p>N.</p> <p>“isUpdatedSpecial” will save partial content;</p> <p>“isAddJobStrategy” will add records associated</p>	Retrieve Contract object in customized callback method: private Strategy getStrategy(ActionForm form, HttpServletRequest request) throws GTSystemException { StrategyForm strategyForm = (StrategyForm)form; Strategy strategy =

ActionName	MethodName	Interceptor Point Description	Additional Parameters?	Sample Code
			with "JobStrategy"; "msgld" will add records associated with "StrategyObjMember".	<pre>(Strategy)request.getAttribute("strategy"); if (strategy == null) { strategy = (Strategy)DbUtil.find(Strategy.class, strategyForm.getStrategyId(), true); } return strategy; }</pre>
quotation	doShowList	Open Order List page	N.	Retrieve Order List information in customized callback method: <pre>Collection<Quotation> quotationLst = (Collection<Quotation>)request.getAttribute(Constants.LIST);</pre>
quotation	doShowInfo	Open an Order	N.	Retrieve Order object in customized callback method: <pre>Quotation quotation = (Quotation)request.getAttribute("quotation");</pre>
quotation	doSave	Save an Order	N.	Retrieve Order object in customized callback method: <pre>private Quotation getQuotation(ActionForm form, HttpServletRequest request) throws GTSYSTEMException { QuotationForm quotationForm = (QuotationForm)form; Quotation quotation = (Quotation)request.getAttribute("quotation"); if (quotation == null) { quotation = (Quotation)DbUtil.find(Quotation .class,</pre>

ActionName	MethodName	Interceptor Point Description	Additional Parameters?	Sample Code
				<pre>quotationForm.getQuotationId(), true); } return quotation; }</pre>

Information on other “actionPoint” are also available upon request to 8Manage Support (support@wisagetech.com).

Note:

All Custom Actions take effect when the same actionPoint has multiple Custom Actions.

When an actionPoint is performing multiple operations as required by business logic, the actionPoint's name takes the form of “action|method” to distinguish the different operations.

3.3 Customize Interface Using 8Manage UI Component

3.3.1 Summary

8Manage allows client to extend 8Manage functionality by developing their own page or modifying existing pages. 8Manage provides the source code of all user interfaces (JSP) allowing user to customize any page easily. User can add new fields in an existing page or even adding a total new page. 8Manage also provides UI components (e.g. calendar, user selection box and auto match, etc.) allowing the customized UI looks just like an out-of-the-box feature.

When developing user interface for a customized feature, you can always re-use existing 8Manage UI components to minimize your custom development efforts.

3.3.2 Implementation

Please refer to the examples in the section 3.2.4.2.

3.4 Customize Menu

3.4.1 Summary

Menu items are entry points to major modules/features.

8Manage provides interface allowing user to add links to application/module in the menu. User can also add their application link or 8Manage page shortcut by modifying the sub-menu.

This section introduces how to add customized menu items to the menu of 8Manage.

3.4.2 Implementation

- 1) **All customized menu items are configured in `customizedMenus-8thManage.xml`.**

You can locate this configuration file under `$(siteHome)/customize/menu`, in which `$(siteHome)` is the 8Manage main directory.

Configuration example in `customizedMenus-8thManage.xml`:

```
<?xml version="1.0" encoding="UTF-8" ?>
```

```
<application id="8thManage" >
  <menu-group id="myAttentionCustomizedMenu" >
    <menu-item id="menu1" key="WisageTech" location="http://www.8manage.com" />
    <menu-item id="menu2" key="Google" location="http://www.google.com" />
  </menu-group>
</application>
```

- **application**

“application” node is the root node of customizedMenus-8thManage.xml. This is a required node.

- **menu-group**

“menu-group” is a child node of “application”, representing a group of user defined menu item configuration. “menu-group” contains one property node, “id”, used as the unique identifier for a group of user defined menu items and it also determines exactly where they are placed on the left menu navigation. This is a required node.

- **menu-item**

“menu-item” is a child node of “menu-group”, representing the configuration of a single menu item. “menu-item” contains 3 properties/child nodes, id, key, and location. They are introduced in detail below.

This is a required node.

- **id**

“id” is the unique identifier of a menu item. It has to be unique among all 8Manage menu IDs. Please make sure this id is named in a way that will not be duplicated.

This is a required node.

- **key**

“key” represents a menu item name. This name will appear on the 8Manage left menu. This name can also be the key value used in resource files. If the key value in resource files is used, then 8Manage will use the corresponding value of the this key as left menu item name.

This is a required node.

- **location**

“location” property represents the menu item URL. When clicked, the inner frame of 8Manage user interface will be directed to this URL, and display the interface of this customized menu item.

2) User-defined menu file “menu-group” ID for UI location

The table below shows the location of customized “menu-group” and current 8Manage menu group.

ID	Module	Location of Appearance
myAttentionCustomizedMenu	My Attention	To the end of “My Attention”
enterpriseAdminCustomizedMenu	Enterprise	To the end of “Import and Replace”
enterpriseInfoCustomizedMenu	Enterprise	To the end of “Information” menu group
enterpriseTrackingCustomizedMenu	Enterprise	To the end of “Tracking” menu group
clientBusinessCustomizedMenu	Work	To the end of “Client Business” menu group
supplierBusinessCustomizedMenu	Work	To the end of “Supplier Related” menu group
internalBusinessCustomizedMenu	Work	To the end of “Internal Program” menu group
myProjectsCustomizedMenu	Work	To the end of “Project & Support” menu group
internalOperationCustomizedMenu	Admin	To the end of “Internal Operation” menu group
projectOpportunityCustomizedMenu	Admin	To then end of “Work” menu group
clientAdminCustomizedMenu	Admin	To the end of “Client & Supplier” menu group
changeOrderAdminCustomizedMenu	Admin	To the end of “Change Request” menu group
productSystemCustomizedMenu	Admin	To the end of “Product & Service” menu group

ID	Module	Location of Appearance
purgeSettingsCustomizedMenu	Admin	To the end of “Purged Data” menu group
systemAdminCustomizedMenu	Admin	To the end of “System” menu group

3.5 Customized Field Name

3.5.1 Summary

In 8Manage, every field name is designed with deliberate efforts. However, user organization may have its own convention or other special requirement in these field names and prefers to use names other than the standard ones. 8Manage supports this type of customization.

3.5.2 Implementation

- 1) Identify the .JSP file containing the field name to be modified/customized.
- 2) Find the key string representing the field name. This key string is also used in the resource language resource file.
- 3) Identify this key string in all resource files under `$(siteHome)\customize\resources`, and replace the standard field name with your desirable one in all these resource files.
- 4) For non-English language string, you need to use the “native2ascii” command comes with JDK to convert your string’s encoding to ASCII in the resource files.
- 5) Re-start 8Manage.

3.5.3 Implementation Examples

Goal: replace the field name “ID Number” used in Contract with the new name “ID” in English version (as shown below)

Proposed Client Contract

* Name:

ID:

* Owners:

* Sales Reps:

* Approver List:

Assisting Groups:

Status: [Not Yet Submitted For Internal Approval](#)

* Owner Groups:

* Legal Entity:

* Approvers:

Assisting Persons:

- 1) Find the page in 8Manage where the field name is located, i.e. "ID Number" in our example, as shown in figure above.
- 2) Open this page in source code view using your web browser and locate a string looks like the one below near the top of the source code page:

```
<base href="http://localhost:8080/8thManage/projectManagement/strategy.jsp">
```

 In this example, "strategy.jsp" is the .JSP file containing the field name to be modified.
- 3) Locate the key string for "ID Number" in "strategy.jsp", i.e. "strategy.label.strategyNo"

```
<table border="0" cellspacing="0" cellpadding="0"
<%=setting != null && setting.isOnlyShowInShowAll() ? "id=\"dynic\" name=\"dynic\" alt=\"strategyNo\"
style=\"display:none\" : \"\"%>>
  <tr><td height="3" colspan="8"></td></tr>
  <tr>
    <td width="<%=String.valueOf(HtmlTagHelper.getWidth(request, width_en, width_zh))%>"
class="fieldName">
<% if (setting != null && !setting.isAutoGen() && !setting.isOptional()) { %><span
class="star">*</span><% } %>
    <%=SessionUtil.getMessage(request, "strategy.label.strategyNo")%></td>
    <td width="3">&nbsp;</td>
    <td>
      <% if (setting != null && !setting.isAutoGen()) { %>
        <gearTag:text property="strategyNo" size="90" maxlength="64"
styleClass="text" alt="<%=setting != null && setting.isOptional() ? "" : "nonnull"%>"
title="<%=SessionUtil.getMessage(request, "strategy.label.strategyNo")%>"
gearUsePolicy="<%=!isIDCanBeUpdate %>" gearHaveHidden="true"/>
        <% } else { %>
          <html:hidden property="strategyNo"/>
          <%=strategyForm.getStrategyNo() != null ? strategyForm.getStrategyNo() :
HtmlTagHelper.EMPTY_DISPLAY_VALUE %>
          <% } %>
        </td>
      <td colspan="5">&nbsp;</td>
    </tr>
  </table>
```

- 4) Add the following key-value to

```

${siteHome}\customize\resources\CustomizedResources_en.properties
strategy.label.strategyNo=ID

```

```
strategy.label.strategyNo=ID
```

- 5) Restart Tomcat

3.5.4 Special Notes

Not all fields can be easily modified. Please take special care for the following field renaming scenarios:

- 1) The most frequently used field names in the system, such as “Responsible Person”, “Description”, “Status”. Their corresponding Key are, *disdisplay.fieldName.responsible*, *display.fieldName.description*, and *display.fieldName.status*. Do not change them unless you are sure that the entire system requires a different name.
- 2) Multiple modules may share one Key. As example above shows, “ID Number” is a key shared among Campaign/Opportunity/Contract. It is all right if you rename “ID Number” to “ID”, but not appropriate if you rename “ID Number” to “Contract Number” because the name “Contract Number” will also appear in Campaign and Opportunity and this seems odd.

3.6 Customize Code Table

3.6.1 Summary

Users may want to customize the code table to stored different kinds of foundation data in 8Manage. This chapter describes how to achieve this goal.

3.6.2 Implementation

- 1) To customize the code table, users first must define a customized owner type string. For example, ‘CustomCodeType’ is chosen as the owner type string.
- 2) Put three customized properties into the customizedResources_XXX.properties file. For example:

```
admin.label.customized.CustomProductType=Customized product type  
admin.label.customized.CustomProductType.l=customized product type  
admin.label.customized.CustomProductType.l.s= Customized product types
```

The property key should follow this rule: Starts with “admin.label.customized.”, then followed with the customized owner type. And then, for the second and the third properties, ends with the suffix.

- 3) In order to add function right checking support. Users should insert 6 records into the func_url_param table, for example:

```
insert into func_url_param(funcId, actionName, methodName, params, dependFuncId)
  values (-1, 'adminType', 'doAdd',      'ownerType=CustomProductType', 9);
insert into func_url_param(funcId, actionName, methodName, params, dependFuncId)
  values (-1, 'adminType', 'doDelete',   'ownerType=CustomProductType', 9);
insert into func_url_param(funcId, actionName, methodName, params, dependFuncId)
  values (-1, 'adminType', 'doShowAdd',  'ownerType=CustomProductType', 9);
insert into func_url_param(funcId, actionName, methodName, params, dependFuncId)
  values (-1, 'adminType', 'doShowInfo', 'ownerType=CustomProductType', 8);
insert into func_url_param(funcId, actionName, methodName, params, dependFuncId)
  values (-1, 'adminType', 'doShowList', 'ownerType=CustomProductType', 8);
insert into func_url_param(funcId, actionName, methodName, params, dependFuncId)
  values (-1, 'adminType', 'doUpdate',   'ownerType=CustomProductType', 9);
```

Please be noted that the “params” field value should be in format “ownerType=YOUR_OWNER_TYPE”. Also the “dependFuncId” field value should be the function Id that your customized code table function belongs to. Users can see the function Id in the “func” table in the database, or can contact wisagetech support service for help.

- 4) Then create a link, for example
 “/common/adminType.do?method=doShowList&ownerType=**CustomProductType**”. Do remember to replace the ownerType to your defined owner type string. Then everything is done.

3.7 Data Privilege Checking Mechanism

3.7.1 Summary

When user needs to read / edit individual record, system may need to perform data right checking. In 8Manage, developers may invoke the data right checking and check whether current user is allowed to read / edit current data object.

3.7.2 Call data right checking

- 1) To check whether current user has right to read the record.
com.gearteks.imatech.common.DataRightManager.getInstance().isReadAllowed(id,

type, context);

Here is the description of the parameters.

- I. id, the data object id.
- II. type, the data object type. See 4.10.3 for detail.
- III. context, user can get the current context through **SessionUtil.getCurrentAclContext(HttpServletRequest request)**

2) To check whether current user has right to edit the record.

com.gearteks.imanage.common.DataRightManager.getInstance().isEditAllowed(id, type, context);

Here is the description of the parameters.

- I. id, the data object id.
- II. type, the data object type. See 4.10.3 for detail.
- III. context, user can get the current context through **SessionUtil.getCurrentAclContext(HttpServletRequest request)**

3) To get the filter criteria for current user for certain type of data object.

com.gearteks.imanage.common.DataRightManager.getInstance().getFilterCriteria(type, context, alias, isHql);

Here is the description of the parameters.

- I. type, the data object type. See 4.10.3 for detail.
- II. context, user can get the current context through **SessionUtil.getCurrentAclContext(HttpServletRequest request)**
- III. alias, the alias in the SQL that represent the data object table.
- IV. isHql, means whether returns Hql format criteria, isHql equals to false returns SQL format criteria.

Please be noted that if no criteria constraint, it will return empty string.

3.7.3 Type List

Here we list some frequently used types.

Category	Type name
Client/Agent/Supplier/Partner/Competitor/Other organizations	Company
Group	Group

Category	Type name
Internal User	InternalUser
External User	ExternalUser
Project	Project
Activity	Activity
Contract	Deal
Opportunity	Strategy
Campaign	Campaign

3.8 Setup SDK Remote Debug Environment

3.8.1 Summary

This section introduces how to setup remote debug environment for 8Manage SDK.

3.8.2 Start 8Manage Service in Debug Mode

A. If your 8Manage is installed under C:\WisageTech, then copy startup-remoteDebug.bat to



startup-remoteDebug.bat

directory C:\WisageTech_installer\8thManage\tomcat\bin.

B. If port 8000 is already in use on the server in which 8Manage is about to run, please open startup-remoteDebug.bat, find a line as below, and rename the port address from 8000 to an available port.

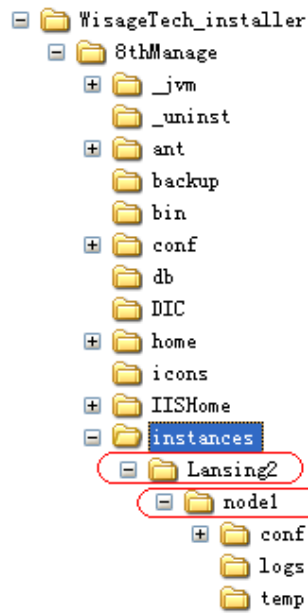
```
set          JAVA_OPTS=%JAVA_OPTS%          -Xdebug
-Xrunjdp:transport=dt_socket,address=8000,server=y,suspend=y
```

C. Open a command line console by typing “cmd” in “Start - Run”, change directory to C:\WisageTech_installer\8thManage\tomcat\bin, and enter the following command:

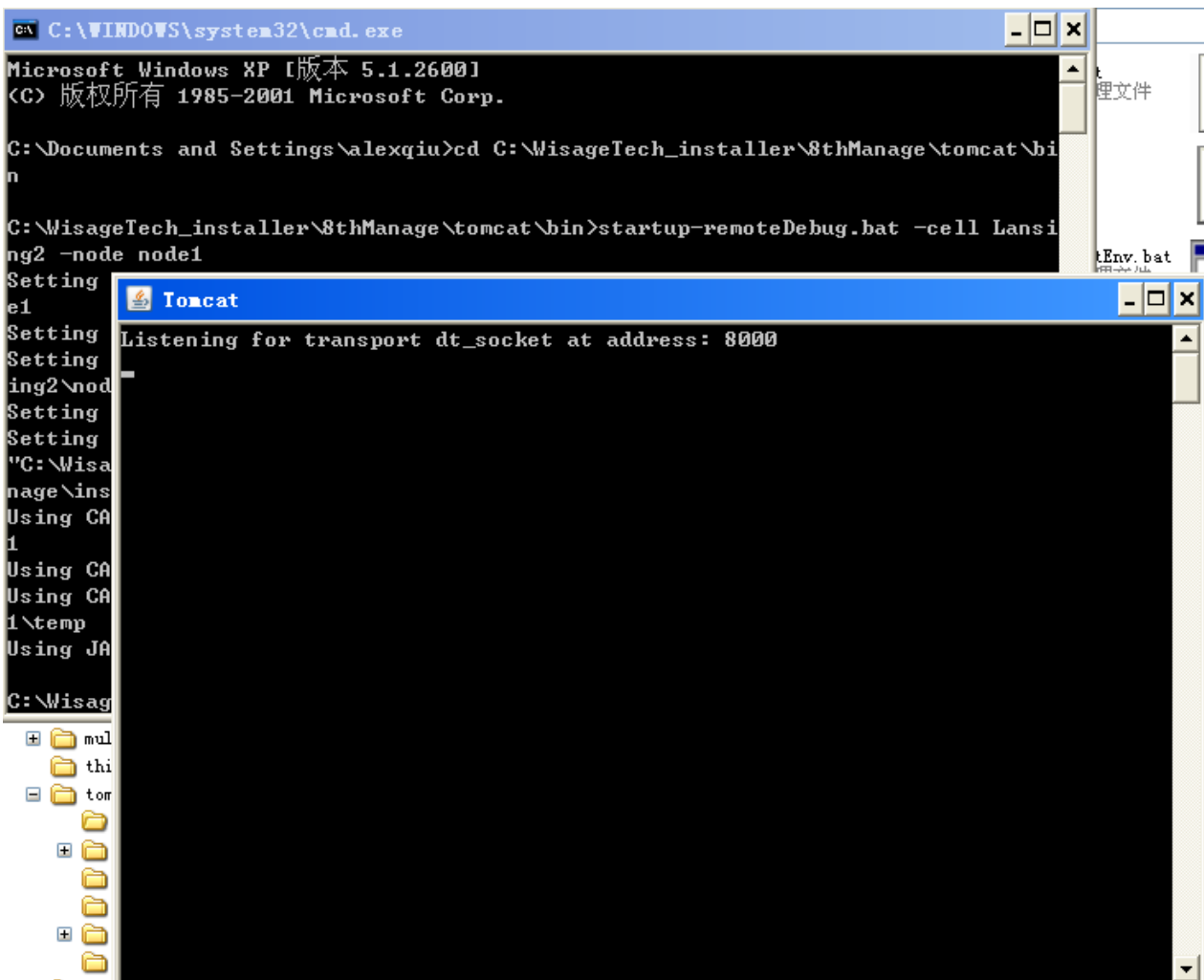
```
startup-remoteDebug.bat -cell $CELLNAME$ -node $NODENAME$
```

Using the example in figure below, the \$CELLNAME\$ is “Lansing2”, \$NODENAME\$ is “node1”, then the command we type is:

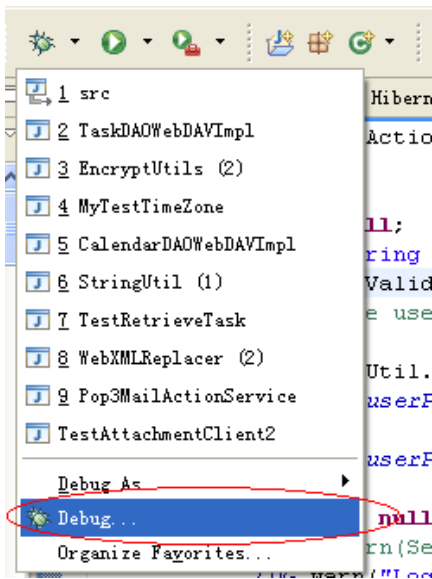
```
startup-remoteDebug.bat -cell Lansing2 -node node1
```



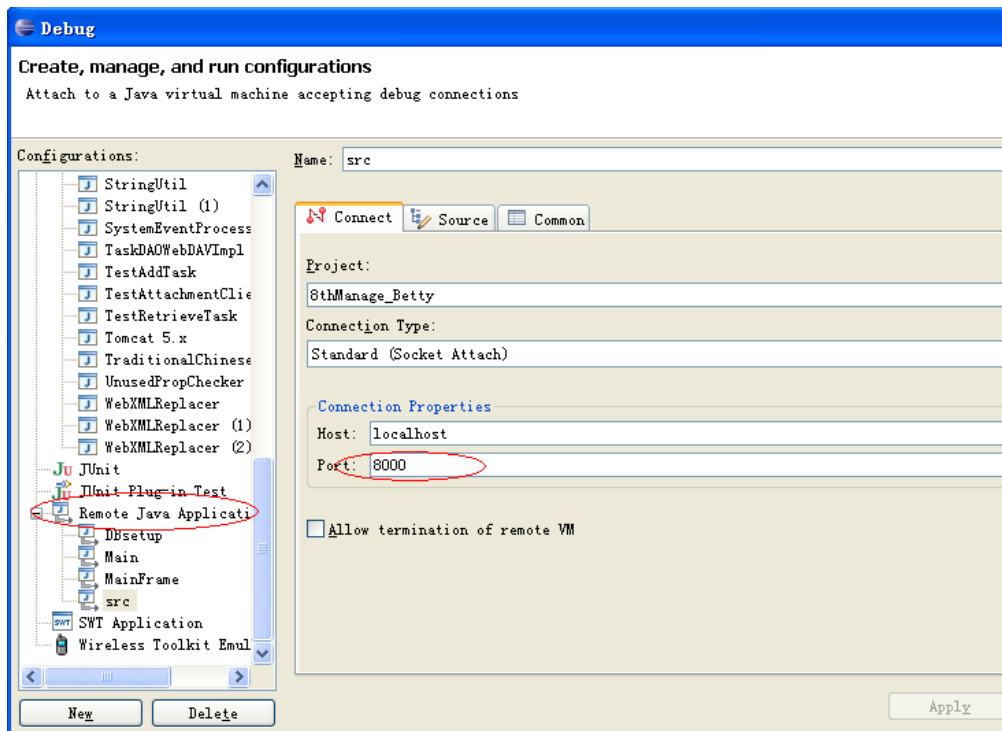
After 8Manage service is started this way, you will see command line console like figure below:



D. New a project in Eclipse. The type of project created does not matter.

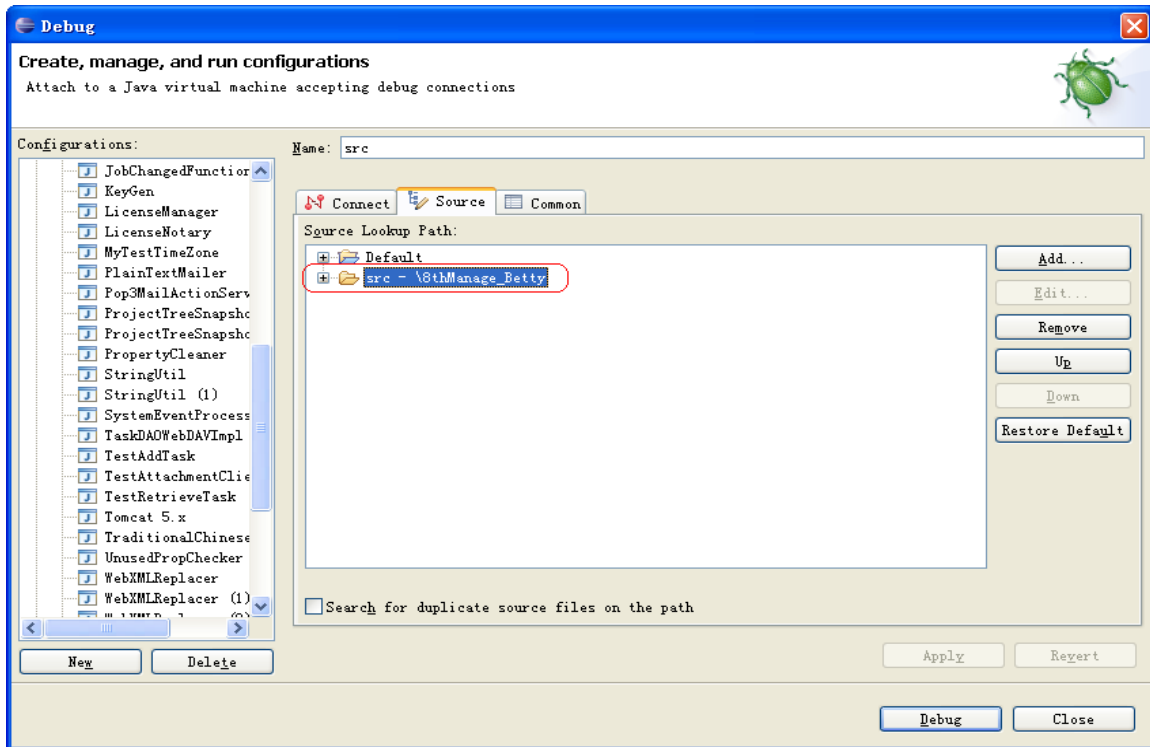


In “Remote Java Application”, create a **Debug** configuration, and configure Host and Port, in which Port is the remote debug port specified in the startup-remoteDebug.bat file.

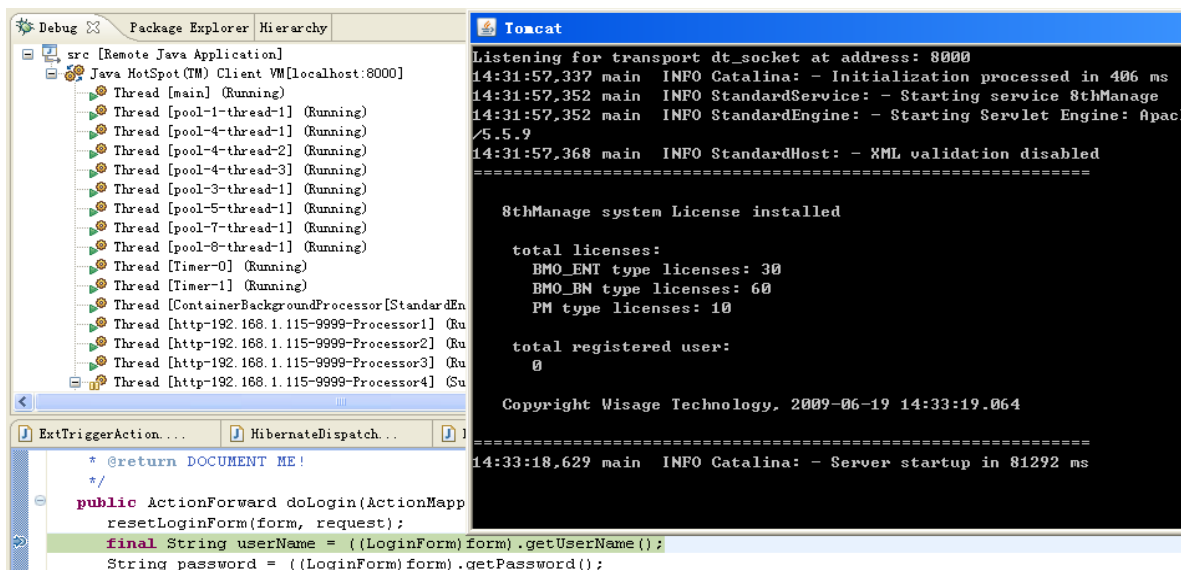
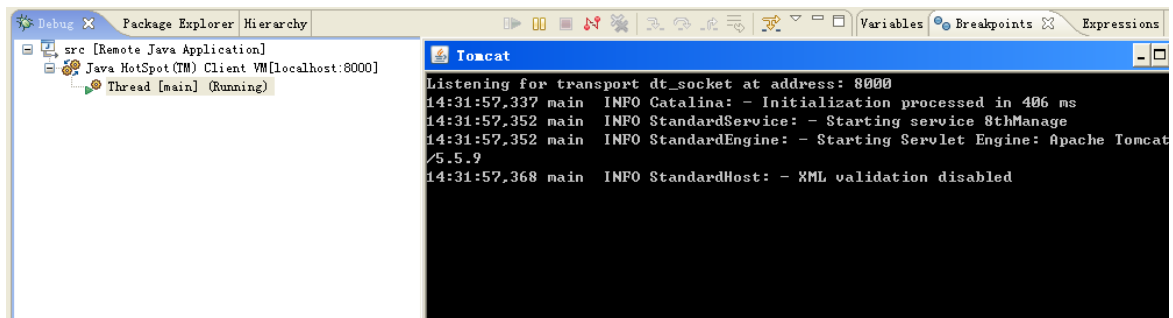


Include the customized source code based on 8Manage. Of course, you need to compile your custom source code classes before you publish them to:

C:\WisageTech_installer\8thManage\webApps\8thManage\WEB-INF\classes



Select “Debug” button and you will see messages in command line consoles, as shown in figures below:



After JSP were compiled into Java, you can find them in the “org” folder under
C:\WisageTech_installer\8thManage\instances\Lansing2\node1\conf\8thManage\localhost

3.9 Implementing Refresh-less Multiple Files Upload and Document Linking

3.9.1 Summary

There are many places in 8Manage requires multiple files uploading. In order to avoid page refreshing when uploading files and to retrieve form information after large file upload fails, this module uses iframe to implement pseudo-Ajax file upload.

3.9.2 Implementation

A. Reference related classes and JavaScript:

```
<%@page import=" com.gearteks.framework.controller. AttachmentHelper "%>
<script type='text/javascript' src='<%=request.getContextPath()%>/js/GTAjaxTable.js'></script>
<script type="text/javascript" src="<%=request.getContextPath()%>/js/gtQuery.js"></script>
```

B. Create a button for browsing local files or creating links to files in Document Library:

File upload button:

```
<%=AttachmentHelper.genAjaxUploadFileButton(String ownerType, Integer ownerId , String title)%>
```

Button for linking to files in Document Library:

```
<%=AttachmentHelper.genAjaxAddDocButton(String ownerType, Integer ownerId, String title)%>
```

Button for moving attachment files to Document Library:

```
<%=AttachmentHelper.genAjaxMoveToLibButton(String ownerType, Integer ownerId) %>
```

ownerType: The belonging object type of the attachment. To avoid duplicated naming, please prefix it with company short name, such as *WisageTech_xxx*.

ownerId: The attachment’s belonging object ID. Use *null* for newly created page.

title: Button name, “New Attachment” by default if no input or input is *null*.

C. Create area for displaying uploaded files:

```
<%=AttachmentHelper.genAjaxUploadFileContent(HttpServletRequest request, String ownerType, Integer ownerId) %>
```

ownerType: The belonging object type of the attachment. To avoid duplicated naming, please prefix it with company short name, such as *WisageTech_xxx*.

ownerId: The attachment's belonging object ID. Use *null* for newly created page.

- D. For newly created page, please add code to the background action for saving the uploaded files, or otherwise there is no need to call it:

```
AttachmentHelper.updateMulAttachments(HttpServletRequest request, boolean hasErrors, String ownerType, Integer ownerId)
```

hasErrors: Checking if there is error message in the background.

ownerType: The attachment's belonging object. To avoid duplicated naming, please prefix it with company short name, such as *WisageTech_xxx*.

ownerId: The attachment's belonging object ID.

- E. Display file list on page:

```
<%=AttachmentHelper.genAjaxAttachmentListTable(HttpServletRequest request, String ownerType, Integer ownerId, boolean editAllowed, Integer pageSize, boolean isShowCheckBox)%>
```

ownerType: The attachment's belonging object. To avoid duplicated naming, please prefix it with company short name, such as *WisageTech_xxx*.

ownerId: The attachment's belonging object's ID.

editAllowed: Having privilege to add or delete attachment?

pageSize: Number of files to display per page.

isShowCheckBox: Displaying check box for moving files to Document Library or not.

Input true if required, or false if not required.

Example code, which can be located in the 8Manage environment and executable when copied to a JSP file.

```
<%@ page contentType="text/html; charset=UTF-8"%>
<%@ include file="/GTcommonJspHeader2.jsp"%>
<%@page import=" com.gearteks.framework.controller.AttachmentHelper "%>
<html>
<head>
<%@ include file="/css.inc" %>
<script type='text/javascript' src='<%=request.getContextPath()%>/js/GTAjaxTable.js'></script>
```

```

<script type="text/javascript" src="<%=request.getContextPath()%>/js/gtQuery.js"></script>
<script>
</script>
</head>
<body scroll="no">
<table width="100%" height="100%" border="0" cellpadding="0" cellspacing="0" class="titleTop">
<tr><td>
  <table width="100%" height="100%" border="0" cellpadding="0" cellspacing="0" >
    <tr><td valign="top">
      <table width="100%" border="0" cellpadding="0" cellspacing="0" id="tblHead">
        <tr>
          <td width="10"></td>
          <td>&nbsp;</td>
          <td></td>
          <td align="right">
            <table border="0" cellpadding="0" cellspacing="0" >
              <tr>
                <td><%=AttachmentHelper.genAjaxUploadFileButton("WisageTech_User",
1)%></td>
                <td width="3" ></td>
                <td width="10" align="center" ></td>
                <td width="3" ></td>
                <td><%=AttachmentHelper.genAjaxAddDocButton("WisageTech_User",
1)%></td>
                <td width="3" ></td>
                <td width="10" align="center" ></td>
                <td width="3" ></td>
                <td><%=AttachmentHelper.genAjaxMoveToLibButton("WisageTech_User",
1)%></td>
              </tr>
            </table>
          </td>
          <td width="10"></td>
        </tr>
      </table>
    </td>
  </table>
</tr>
</table>

```

```
<tr><td class="Linetop"></td></tr><tr><td height="10"></td></tr>
</table>
<%@ include file="/errorTable2.inc"%>
<table width="100%" border="0" cellpadding="0" cellspacing="0">
<tr>
<td><%=AttachmentHelper.genAjaxUploadFileContent(request, " WisageTech_User",
1) %></td>
</tr>
<tr>
<td><%=AttachmentHelper.genAjaxAttachmentListTable(request, " WisageTech_User",
1, true, 20)%></td>
</tr>
</table>
</td></tr>
</table>
</td></tr>
</table>
</body>
</html>
```


3.10 Implementation of Audit Trail

3.10.1 Summary

Audit Trail is used for keeping logs on database fields that are modified by users. Audit Trail records the following information, who performed the operation/change, time, event type, type of change, values before and after the change, etc. These logging information are recorded into log tables either via trigger or programmatically.

3.10.2 Log's Generic Table Structure

```
CREATE TABLE strategy_log(
  LogId          int          IDENTITY(1,1) NOT NULL, //Self-increment ID
  Event          type3       NOT NULL, //Event type
  StrategyId    id          NOT NULL, //Main table id where the change took place
  FieldType     type3       NULL, //Field type being changed
  FieldName     string2     NULL, //Name of customized field
  FieldId       id          NULL, //ID of customized field
  ExtendId      id          NULL, //ID of customized field type

  NewValueStr   string3     NULL, //New string value
  NewValueNum   float       NULL, //New number value
  NewValueDate  datetime    NULL, //New date value
  NewValueId    id          NULL, //New integer value
  NewObjType    string1     NULL, //New object type value
  NewObjId      id          NULL, //New object ID value

  OldValueStr   string3     NULL, //Old string value
  OldValueNum   float       NULL, //Old number value
  OldValueDate  datetime    NULL, //Old date value
  OldValueId    id          NULL, //Old integer value
  OldObjType    string1     NULL, //Old object type value
  OldObjId      id          NULL, //Old object ID value

  TS ts        CONSTRAINT [DF1_strategy_log] DEFAULT (getdate()) NOT NULL, //Time of user
change
  TSUserId      id          NOT NULL //User who performed the change
)
```

```

go
ALTER TABLE strategy_log ADD
    CONSTRAINT PK_strategy_log PRIMARY KEY CLUSTERED (LogId)
go
ALTER TABLE strategy_log ADD CONSTRAINT IF1_strategy_log
    FOREIGN KEY (StrategyId)
    REFERENCES strategy(StrategyId) ON DELETE CASCADE
go
CREATE INDEX IE1_strategy_log ON strategy_log (StrategyId, TS, Event)
go

```

Note: In table “company_log” and “job_log”, “NewValueStr” corresponds to “NewValueString” and “OldValueStr” corresponds to “OldValueString”.

3.10.3 Java Class and Hibernate Configuration Files

All Java classes inherit from super class `com.gearteks.framework.object.AbstractObjectLog`. Please refer to the Hibernate configuration files for `StrategyLog`, as well as the `StrategyLog.hbm.xml`.

3.10.4 Write Log Table with SQL and Trigger, and Write Key-value in Resource File

Code example: SQL insert code

```

INSERT INTO strategy_log (Event, FieldType, StrategyId, NewValueStr, OldValueStr, TS, TSUserId)
values('Mod_StrategyName', 'Str', 1, 'NewName', 'OldName', getdate(), 1);

INSERT INTO strategy_log (Event, FieldType, StrategyId, NewValueNum, OldValueNum, TS,
TSUserId)
values('Mod_StrategyProgress', 'Num', 1, 100, 80, getdate(), 1);

INSERT INTO strategy _log (Event, FieldType, StrategyId, NewValueDate, OldValueDate, TS,
TSUserId)
values('Mod_StrategyDealDate', 'Date', 1, '2009-10-29', '2009-10-21', getdate(), 1);

INSERT INTO strategy_log (Event, FieldType, StrategyId, NewObjType, NewObjId, NewValueStr,
OldObjType, OldObjId, OldValueStr, TS, TSUserId)
values('Mod_StrategyCurrency', 'Select', 1, 'Currency', 3, 'CNY', 'Currency', 1, 'USD', getdate(), 1);

```

```
INSERT INTO strategy_log (Event, FieldType, StrategyId, NewValueNum, OldValueNum, TS,
TSUserId)
```

```
values('Mod_StrategyIsOutSourcing', 'Yesno', 1, 1, 0, getdate(), 1);
```

```
INSERT INTO strategy_log (Event, FieldType, StrategyId, NewObjType, NewObjId, TS, TSUserId)
```

```
values('Add_StrategyResp', 'SystemObj', 1, 'User', 17, getdate(), 1);
```

```
INSERT INTO strategy_log (Event, FieldType, StrategyId, OldObjType, OldObjId, TS, TSUserId)
```

```
values('Del_StrategyResp', 'SystemObj', 1, 'User', 5, getdate(), 1);
```

For trigger, use Strategy_u.trg

```
INSERT INTO strategy_log (Event, FieldType, StrategyId, NewValueStr, OldValueStr, TS, TSUserId)
  SELECT 'Mod_StrategyName', 'Str', i.StrategyId, i.StrategyName, d.StrategyName, getdate(),
i.TSUserId
```

```
  FROM inserted i, deleted d
```

```
  WHERE i.StrategyId = d.StrategyId AND
```

```
        i.StrategyName != d.StrategyName;
```

```
INSERT INTO strategy_log (Event, FieldType, StrategyId, NewValueNum, OldValueNum, TS,
TSUserId)
```

```
  SELECT 'Mod_StrategyProgress', 'Num', i.StrategyId, i.Progress, d.Progress, getdate(), i.TSUserId
```

```
  FROM inserted i, deleted d
```

```
  WHERE i.StrategyId = d.StrategyId AND
```

```
        i.Progress != d.Progress;
```

```
INSERT INTO strategy_log (Event, FieldType, StrategyId, NewValueDate, OldValueDate, TS,
TSUserId)
```

```
  SELECT 'Mod_StrategyDealDate', 'Date', i.StrategyId, i.DealDate, d.DealDate, getdate(), i.TSUserId
```

```
  FROM inserted i, deleted d
```

```
  WHERE i.StrategyId = d.StrategyId AND
```

```
        i.DealDate != d.DealDate;
```

```
INSERT INTO strategy_log (Event, FieldType, StrategyId, NewObjType, NewObjId, NewValueStr,
OldObjType, OldObjId, OldValueStr, TS, TSUserId)
```

```
  SELECT 'Mod_StrategyCurrency', 'Select', i.StrategyId, 'Currency', i.CurrencyId,
newCurrency.CurrencyName, 'Currency', d.CurrencyId, oldCurrency.CurrencyName, getdate(),
i.TSUserId
```

```

FROM inserted i, deleted d, currency newCurrency, currency oldCurrency
WHERE i.StrategyId = d.StrategyId AND
      i.CurrencyId != d.CurrencyId AND
      i.CurrencyId = newCurrency.CurrencyId AND
      d.CurrencyId = oldCurrency.CurrencyId;

INSERT INTO strategy_log (Event, FieldType, StrategyId, NewValueNum, OldValueNum, TS,
TSUserId)
  SELECT 'Mod_StrategyIsOutSourcing', 'Yesno', i.StrategyId, i.IsOutSourcing, d.IsOutSourcing,
getdate(), i.TSUserId
  FROM inserted i, deleted d
  WHERE i.StrategyId = d.StrategyId AND
        i.IsOutSourcing != d.IsOutSourcing;

```

strategy_member_i.trg

```

INSERT INTO strategy_log (Event, FieldType, StrategyId, NewObjType, NewObjId, TS, TSUserId)
  SELECT 'Add_StrategyResp', 'SystemObj', i.StrategyId, i.ObjType, i.ObjId, getdate(), s.TSUserId
  FROM inserted i, strategy s
  WHERE i.StrategyId = s.StrategyId
        i.Involvement = 'Resp';

```

strategy_member_d.trg

```

INSERT INTO strategy_log (Event, FieldType, StrategyId, OldObjType, OldObjId, TS, TSUserId)
  SELECT 'Del_StrategyResp', 'SystemObj', d.StrategyId, d.ObjType, d.ObjId, getdate(), s.TSUserId
  FROM deleted d, strategy s
  WHERE d.StrategyId = s.StrategyId
        d.Involvement = 'Resp';

```

All the Event above shall be written/updated corresponding key-value in resource files, as example below shows:

```
name.objectLog.Mod_StrategyProgress=Progress
```

Note:

For object (to be referenced) that may be deleted:

When writing value into NewObjType, NewObjId, OldObjType, and OldObjId, please don't forget to

also write the object name to NewValueStr and OldValueStr. In the Currency code example above, NewObjType and NewObjId are used when the currency being referenced is still in use whereas the NewValueStr previously saved is used as currency name when the currency was deleted.

For object (to be referenced) that will not be deleted:

No need to write to NewValueStr and OldValueStr, for example, User, Party, Company. Also in example of strategy_member_i.trg and strategy_member_d.trg above, no object name is written to NewValueStr and OldValueStr.

3.10.5 Code for List Query and Table JSP

To search the Audit Trail of certain object is actually to search its corresponding log table. For example, query for searching Audit Trail of Strategy is as followed:

SQL:

```
select * from strategy_log where StrategyId = ? order by TS desc, LogId desc
```

HQL

```
select sl from StrategyLog sl where sl.strategyId = ? order by sl.ts desc, sl.logId desc
```

Code example for list display table:

```
<gearTable:table name="requestScope.queryResult" id="item"
requestURI="<%=request.getContextPath() +
"/projectManagement/strategy.do?method=doShowLogList"%>" styleClass="displayTag"
pagesize="<%=pageSize%>" defaultsort="2" sort="list" border="0" width="100%" cellpadding="1"
cellpadding="0" needPagination="true">
    <gearTable:column align="left"
title="<%=SessionUtil.getMessage(request,"display.fieldName.date")%>" sortable="true"
headerClass="sortable" width="128" colExtInfo="ts">
        <%=DateUtil.tsFormat(((AbstractObjectLog)item).getTs(), request)%>
    </gearTable:column>
    <gearTable:column align="left"
title="<%=SessionUtil.getMessage(request,"name.coLog.changedBy")%>" sortable="true"
headerClass="sortable" colExtInfo="tsUser">
        <%=HtmlUtil.genLink(((AbstractObjectLog)item).getTsUser())%>
    </gearTable:column>
```

```

        <gearTable:column align="left"
title="<%=SessionUtil.getMessage(request,"name.job.Event")%>" sortable="true"
headerClass="sortable" colExtInfo="event">
            <%=((JobLog)item).getEventDisplay()%>
        </gearTable:column>
        <gearTable:column align="left"
title="<%=SessionUtil.getMessage(request,"name.coLog.newValue")%>" sortable="false"
headerClass="sortable">
            <%=StringUtil.nvl(((AbstractObjectLog)item).getNewValue(), "")%>
        </gearTable:column>
        <gearTable:column align="left"
title="<%=SessionUtil.getMessage(request,"name.coLog.oldValue")%>" sortable="false"
headerClass="sortable" >
            <%=StringUtil.nvl(((AbstractObjectLog)item).getOldValue(), "")%>
        </gearTable:column>
        <gearTable:setProperty name="paging.banner.item_name"
value="<%=TagUtils.getInstance().message(pageContext,null,null,"paging.banner.item_name",null) %
>" />
        <gearTable:setProperty name="paging.banner.items_name"
value="<%=TagUtils.getInstance().message(pageContext,null,null,"paging.banner.items_name",null)
%>" />
        <gearTable:setProperty name="paging.banner.all_items_found"
value="<%=TagUtils.getInstance().message(pageContext,null,null,"paging.banner.all_items_found",nul
l) %>" />
        <gearTable:setProperty name="paging.banner.some_items_found"
value="<%=TagUtils.getInstance().message(pageContext,null,null,"paging.banner.some_items_found"
,null) %>" />
        <gearTable:setProperty name="paging.banner.no_items_found"
value="<%=TagUtils.getInstance().message(pageContext,null,null,"paging.banner.no_items_found",nul
l) %>" />
        <gearTable:setProperty name="paging.banner.one_item_found"
value="<%=TagUtils.getInstance().message(pageContext,null,null,"paging.banner.one_item_found",nu
ll) %>" />
        <gearTable:setProperty name="basic.msg.empty_list"
value="<%=TagUtils.getInstance().message(pageContext,null,null,"basic.msg.empty_list",null) %>" />
        <gearTable:setProperty name="paging.banner.onepage"
value="<%=TagUtils.getInstance().message(pageContext,null,null,"paging.banner.onepage",null) %>"
/>

```

```

        <gearTable:setProperty name="paging.banner.page.selected"
value="<%=TagUtils.getInstance().message(pageContext,null,null,"paging.banner.page.selected",null)
%>" />
        <gearTable:setProperty name="paging.banner.page.link"
value="<%=TagUtils.getInstance().message(pageContext,null,null,"paging.banner.page.link",null) %>"
/>
        <gearTable:setProperty name="paging.banner.page.separator"
value="<%=TagUtils.getInstance().message(pageContext,null,null,"paging.banner.page.separator",null
) %>" />
        <gearTable:setProperty name="paging.banner.last"
value="<%=TagUtils.getInstance().message(pageContext,null,null,"paging.banner.last",null) %>" />
        <gearTable:setProperty name="paging.banner.first"
value="<%=TagUtils.getInstance().message(pageContext,null,null,"paging.banner.first",null) %>" />
        <gearTable:setProperty name="paging.banner.full"
value="<%=TagUtils.getInstance().message(pageContext,null,null,"paging.banner.full",null) %>" />
        <gearTable:setProperty name="paging.banner.placement"
value="<%=TagUtils.getInstance().message(pageContext,null,null,"paging.banner.placement",null) %>
" />
        <gearTable:setProperty name="basic.msg.empty_list_row"
value="<%=TagUtils.getInstance().message(pageContext,null,null,"basic.msg.empty_list_row",null) %>
" />
        <gearTable:setProperty name="paging.banner.group_size"
value="<%=TagUtils.getInstance().message(pageContext,null,null,"paging.banner.group_size",null) %
>" />
    </gearTable:table>

```

3.11 Send Customized Email

You can customize the email sent from 8Manage to users by using the class `com.gearteks.imanage.common.trigger.NotificationHandler`. The following is an example that will send customized email to responsible persons when user clicks “**Become Official**” button.

- Configuration

```

<action actionPoint="strategy|saveStrategy" location="post"
    sameTransaction="true" useForward="false"
    actionClass="com.gearteks.test.customized.CustomizedStrategyAction"
    actionMethod="sentEmailWhenOfficial" />

```

- Sample code

```
public class CustomizedStrategyAction implements Callbackable {
    public ActionForward sentEmailWhenOfficial(ActionMapping mapping,
    ActionForm form,
    HttpServletRequest request,
    HttpServletResponse response)
        throws GTEException {
        StrategyForm strategyForm = (StrategyForm)form;
        Integer strategyId = strategyForm.getStrategyId();
        Strategy strategy = null;
        if (strategyId != null) {
            strategy = (Strategy)DBObject.find(Strategy.class, strategyId, true);
        }

        Strategy srcStrategy = (Strategy)request.getAttribute(StrategyAlertFunction.OLD_STRATEGY);
        String oldStatus = srcStrategy.getStatus();

        if (strategy != null) {
            boolean becomeOfficial = Strategy.STATUS_APPROVE.equals(strategy.getStatus()) &&
                Strategy.adjustAllowed(oldStatus) &&
                strategy.isNoApproval(null, null, null, null, null, null);
            if (becomeOfficial) {
                NotificationHandler nh = new NotificationHandler();
                Message msg = new Message();

                Collection resps = strategy.getResponsibleMembers();
                Collection clients = StrategyLogic.getInstance().getStrategyClients(strategyId);
                String clientName = CollectionUtil.getNames(clients);

                msg.addReceivers(resps);
                msg.setSubject(strategy.getName());
                msg.setContent(strategy.getName() + " - " + clientName);

                nh.sendMessage(msg, null, false, true, false);
            }
        }
        return null;
    }
}
```



```
}

```

Here is some explanation about the arguments of `NotificationHandler.sendMessage` (`Message msg`, `Template template`, `boolean filterByUserSetting`, `boolean withSuspended`, `boolean filterContact`) API.

- `msg` - the argument that contains the email subject, content, to, cc, bcc, attachments.
- `template` - the argument that represents the email template, if it is null, means no email template is used.
- `filterByUserSetting` - If it is false, means that this email does not need to check user notification setting. (Always sent)
- `withSuspended` - If it is true, the email will be sent to the users that even suspended.
- `filterContact` - If it is false, means that the email does not need to filter contact users.

3.12 Customize Search Criteria

User can customize the search criteria in the search page. The following example shows that how to add customized search criteria in Opportunity/Strategy List page.

- Configuration

```
<action actionPoint="strategy|listStrategy" location="pre"
  sameTransaction="true" useForward="false"
  actionClass="com.gearteks.test.customized.CustomizedStrategyAction"
  actionMethod="setCustomizedCriteria" />

```

- Sample code

```
public ActionForward setCustomizedCriteria(ActionMapping mapping, ActionForm form,
HttpServletRequest request, HttpServletResponse response)
  throws GTEException {
  PageInfo pi = SessionUtil.getPageInfo(request);
  if (Strategy.STRATEGY_TYPE.equals(request.getParameter("objType")) &&
  DbFieldValue.STRATEGY_BUSINESSTYPE_SELLING.
  equals(request.getParameter("businessType"))) {
    pi.setHqlConjunction(" exists(select s1.startDate from Strategy s1 where s1.startDate >
'2010-01-01 00:00:00' and s1.strategyId = s.strategyId) ");
    pi.setSqlConjunction(" exists(select s1.startDate from Strategy s1 where s1.startDate >
'2010-01-01 00:00:00' and s1.strategyId = s.strategyId) ");
  }
  return null;
}

```

Here are the notes for this feature.

- Current version doesn't support parameters in the criteria.
- 8Manage uses both SQL & HQL for search, so the customized search must set both SQL & HQL by using `setSqlConjunction` & `setHqlConjunction`.
- The SQL & HQL alias in opportunity list page is "s", but this may be different in different List pages. You may want to contact 8Manage support for this if you want to customize search criteria in other list pages.

3.13 Customize Discussion Name

User can customize a discussion (forum) name with one's own logic. The following example shows how to customize name for the discussion (forum) in Opportunity/Strategy approval.

- Configuration

```
<action actionPoint="strategy|saveStrategy" location="post"
  sameTransaction="true" useForward="false"
  actionClass="com.gearteks.test.customized.CustomizedStrategyAction"
  actionMethod="resetDiscussionName" />
```

- Sample code

```
public ActionForward resetDiscussionName(ActionMapping mapping, ActionForm form,
HttpServletRequest request, HttpServletResponse response)
  throws GTEException {
  if (Strategy.STRATEGY_TYPE.equals(request.getParameter("objType")) &&
  DbFieldValue.STRATEGY_BUSINESSTYPE_SELLING.
  equals(request.getParameter("businessType"))) {
    StrategyForm strategyForm = (StrategyForm)form;
    Integer strategyId = strategyForm.getStrategyId();
    Strategy strategy = null;
    if (strategyId != null) {
      strategy = (Strategy)DbObject.find(Strategy.class, strategyId, true);
    }

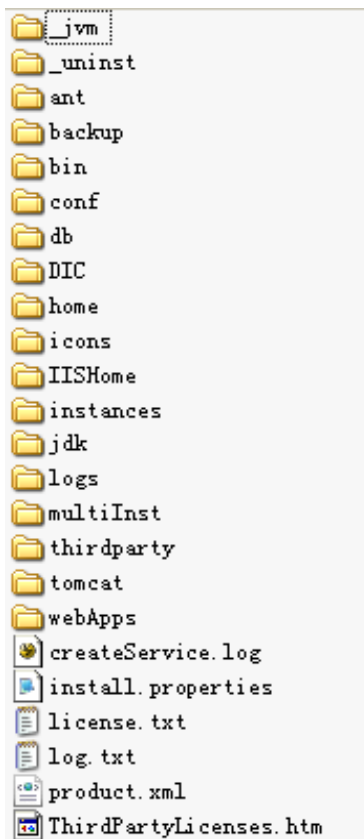
    if (strategy != null) {
      Approval appr = strategy.getCurrentApproval();
      if (appr != null) {
        Discussion discussion = ApprovalHelper.getFirstDiscussion(appr);
```

```
        discussion.setName("Here is a test...");
    }
}
return null;
}
```

4 8Manage Directory Structure

4.1 8Manage Default Structure

Main folders under the directory `%WisageTech_HOME%\8thManage` are listed in the screenshot below:



In 8Manage, the following directories are related to secondary development:

- Home\localhost\customize\callback: configuration files for request callback mechanism
- Home\localhost\customize\customize: configuration files for customizing system properties

- Home\localhost\customize\menu: configuration files for customized menu
- Home\localhost\customize\resource: configuration files for customized resources
- instances\%CELL_NAME%\%NODE_NAME%\conf\8thManage\localhost: used for storing system configuration files, including column definitions of importing/exporting excel files, log4j settings and system properties settings.
- Logs: used for storing 8Manage's system log files
- instances\%CELL_NAME%\%NODE_NAME%\logs: used for storing tomcat's log files

WebApps Directory Introduction



8Manage's subdirectories are listed below:

Approval: pages related to system's approval feature

Attachment: pages related to system's attachments

ChangeOrder: pages related to system's change request feature

Chat: pages related to system's instant chatting feature

Comment: pages related to system's comment feature

Common: pages related to public feature like system prompt.

Css: used for storing css files

Customization: customized pages

customizeReport: customized report pages

discussion: pages related to system's discussion feature

download: templates used in the system

ext: pages related to integration of third-party systems

help: help pages (user manual) integrated in the system

home: features related to system's "My Workbench" module. The directory tree under this directory includes Home Dashboard\Timesheet\My Report\Expense.

Images: for storing images

Integration: pages required by integration

Js: used for storing JavaScript files

Library: related to system's library feature

M: pages related to mobile access

Maintain: main objects in the system, including: user, group, product, resource, timesheet, system settings, approval, etc.

Menu: related to system's menus

Oais: pages of OA related and HR related features

Popup: system's popup related pages

proejctManagment: pages related to project management feature

psa: pages related to order, invoice, quotation, revenue, etc.

strategy: email campaign and cost related reports

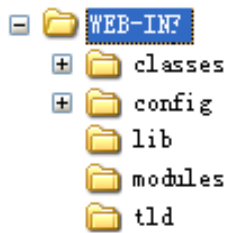
support: related to system's "Support" page

trackingmatrix: related to project's "Tracking" page

vm: templates for dashboard pages

WEB-INF: Compiled JAVA codes

WEB-INF Directory Introduction



Classes: compiled JAVA's class files

Config: configuration files for struts, axis2 and menus

Lib: 8Manage's jar files

Modules: modules used for axis2

Tld: tag library descriptor files, including struts tags and custom tags.

Com Directory Introduction



This directory is used for storing 8Manage's source codes, part of which will be obfuscated when they are released.

Gearteks: for storing system architecture source codes and various business modules' source codes

Wisagetech: for storing secondary development framework's source codes and customization source codes provided by WisageTech.

XXX: for storing customization source codes provided by 8Manage's users.

4.2 Customization Directory Suggestion

To avoid the situation where customized codes and system default codes are mixed up or overwritten by each other, it is highly suggested that customized codes be stored under a separate directory different from 8Manage's default directories. For example, you can create a folder whose directory looks like `com\xxx\xxx\report`. What's more, it is also suggested that JSP files be kept under another separate directory like `webApps\8thManage\xxx`.

5 How to Deploy Customized Codes

5.1 Deploy JSP Codes

- JSP codes can be copied and directly pasted to `%WisageTech_HOME%\8thManage\webApps\8thManage\xxx`. The "xxx" directory here refers to the folder where you store your customized JSP codes.
- Delete the existing files under the directory `%WisageTech_HOME%\8thManage\instances\%CELL_NAME%\%NODE_NAME%\conf\8thManage\localhost\org\apache\jsp\xxx`.
- Restart 8Manage service.

5.2 Deploy JAVA Codes

- Copy the bytecodes generated after compiling JAVA codes, and then them to the directory `%WisageTech_HOME%\8thManage\webApps\WEB-INF\classes\com\xxx`. The "xxx" directory here refers to the folder where you store your customized JAVA codes.
- Restart 8Manage service.

5.3 Deploy Configuration Files

- Directly modify 8Manage's existing configuration files.
- Add new configuration files.
- Restart 8Manage service.

8 can provide best combination of standard products & redevelopment services for enterprise management and over 500 corporations in Asia are using our following modules on-premises or SaaS:

8 CRM : Corporate Client CRM and Consumer CRM

8 Service : Service Management

8 SRM : Supplier Management, e-Procurement and e-Tender

8 PPM : Project and Portfolio Management

8 New Way : Visual Agile and Lean

8 Timesheet : Resource Time and Cost Management

8 EDMS : Electronic Document Management System

8 OA : Office Automation

8 HCM : Human Capital Management

8 All-in-one : Enterprise Full Automation