HP Service Manager 9.20 Integration Package for CCRM Solution 9.00

Installation and Administration Guide

Table of Contents

Overview

The HP Service Manager 9.20 Integration Package for Change Configuration and Release Management (CCRM) Solution 9.00 (referred to as "the CCRM integration package" throughout this document) is designed to be installed on top of HP Service Manager 9.20. This package provides enhancements to the out-of-box Release Control (RC) and Operations Orchestration (OO) integrations in Service Manager 9.20.

This document describes how System Administrators should install the CCRM integration package and set up their environments for using the enhanced RC and OO integrations.

For information about how end users can use the features of the CCRM integration package, see the *HP Service Manager 9.20 Integration Package for CCRM Solution 9.00 User Guide.* To download this user guide, visit the HP manual site at: http://h20230.www2.hp.com/selfsolve/manuals

RC integration enhancement

An enhancement to the out-of-box Release Control (RC) integration that is available in Service Manager 9.20. This enhancement provides support for multitenancy.

System Administrators can synchronize tenant information from Service Manager to RC so that they have matched tenants. When System Administrators enable the multitenancy mode in Service Manager, users can view only the change information of their own tenants in both Service Manager and RC.

OO integration enhancement

An enhancement to the out-of-box Operations Orchestration integration that is available in Service Manager 9.20. This enhancement enables Change Management users to implement automated deployment of requests for change (RFCs) through manually or automatically launching OO flows from RFCs.

For more information, see the *HP Service Manager 9.20 Integration Package for CCRM Solution 9.00 User Guide.*

Installation

This section describes how to install the CCRM integration package in your production system. The CCRM integration package provides two unloads:

Unload Name	Description
ccrm-core.unl	Contains new records, which will not modify any existing records in Service Manager.
ccrm-demo.unl	Contains records that will modify some existing records in Service Manager. For the existing records to be modified by the CCRM integration package, see Table 1 and Table 2.

You can select to automatically or manually install the CCRM integration package depending on whether and how your production system has been customized.

If you have made customizations to any records listed in tables 1 and 2, you may want to perform a manual installation so that your customizations will not be overwritten; if not, you can perform an automatic installation.

Table 1: Existing records to be modified by the RC integration enhancement

File	KeyValue
ScriptLibrary	RCUrl
ScriptLibrary	smis_RestClient
DisplayScreen	cm.open.display
SMISRegistry	SMtoRC
dbdict	company
datadict	company
formatctrl	company
format	company.g
process	save.company
process	delete.company

Table 2: Existing records to be modified by the OO integration enhancement

File	KeyValue
Process	cm.update
Process	cm.update.save
Process	cm.open
Process	cm.open.save
States	cm.open
States	cm.view
cm3rcatphase	Change Implementation
cm3rcatphase	Build and Test
datadict	cm3r
datadict	info
dbdict	cm3r
dbdict	info
Format	CM.change.approval
Format	CM.change.approvalprep
Format	CM.change.closure
Format	CM.change.implementation
Format	CM.change.logging
Format	CM.change.planning
Format	CM.change.review
Format	cm3r.release
Format	cm3r.release.close
link	cm3r
formatctrl	cm3r
Displayoption	db.view

Automatic installation

The following steps install both the OO and RC integration enhancements in your system.

1. Log on to Service Manager as a System Administrator.
2. Load the `ccrm-core.unl` file into the system.
3. Load the `ccrm-demo.unl` file into the system.

Manual installation

This procedure includes loading the CCRM core unload and manually implementing the CCRM demo unload for both the RC and OO integration enhancements.
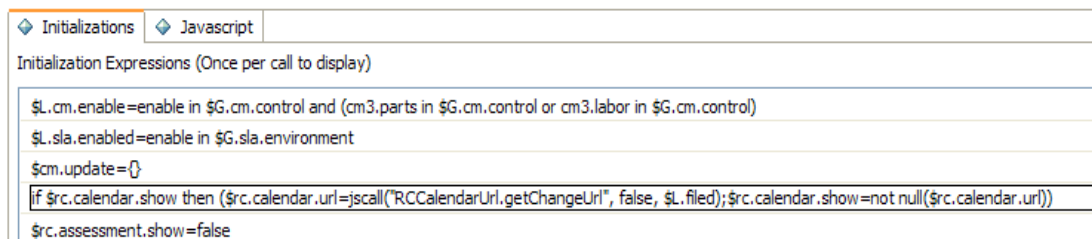
Task 1: Load the CCRM core unload

1. Log on to Service Manager as a System Administrator.
2. Load the ccrm-core.unl file into your system.

   All the new records from the CCRM integration package are now installed in your system.

Task 2: Manually implement an RC Change Calendar hot fix

   NOTE: This is a mandatory task, and you must manually finish the hot fix implementation.

1. Log on to Service Manager as a System Administrator.
2. Update the **cm.open.display** display screen record.

   a. Go to **Tailoring** > **Tailoring Tools** > **Display Screens**, and open the **cm.open.display** screen record.

   b. On the **Initialization** tab, locate `if $rc.calendar.show then ($rc.calendar.url=jscall("RCCalendarUrl.getChangeUrl", ($L.tableAccess.update), $L.filed);$rc.calendar.show=not null($rc.calendar.url))` in Initialization Expressions.

   c. Replace `($L.tableAccess.update)` with `false`. See the screenshot as below:

```
◇ Initializations | ◇ Javascript
Initialization Expressions (Once per call to display)
$L.cm.enable=enable in $G.cm.control and (cm3.parts in $G.cm.control or cm3.labor in $G.cm.control)
$L.sla.enabled=enable in $G.sla.environment
$cm.update={}
if $rc.calendar.show then ($rc.calendar.url=jscall("RCCalendarUrl.getChangeUrl", false, $L.filed);$rc.calendar.show=not null($rc.calendar.url))
$rc.assessment.show=false
```

   d. Save the record.

Task 3: Manually implement the RC integration enhancement

1. Update the script **RCUrl** in Script Library.

   a. Add the following function to the end of the script:
```
function getParameterForTenant(){
var company = vars.$lo_operator.company;
if(company==null){
return "";
}else{
return "&tenant_field_val=" + company
}
```

   b. Update the function **getCommonParameters**. The newly added part is in bold:
```
function getCommonParameters(){
var para = "hideRCHeader=true";
```

```
            para += getParameterForLocale();
            para += getParameterForTimezone();
            if(vars.$G_multi){
            para += getParameterForTenant();
            }
            return para;
            }
```

2. Update the script **smis_RestClient** in Script Library.

   a. Add the following **doRESTPostRequest** and **doRESTDeleteRequest** functions to the script
      smis_RestClient:

```
/**
 * @function:
 *   Do POST Action using global method: doHTTPRequest.
 *   @param url:
 *      url; type:string
 * @param httpHeaders:
 *      http headers; type:array
 * @param postBody:
 *      post body; type:string
 * @param connTimeout, sendTimeout, recvTimeout.
 *      timeout parameters; type:number
 *   @param errorMsg:
 *      errorMsg; type:string
 * @return:
 *   response; type:string
 *
 */
function doRESTPostRequest(url, httpHeaders, postBody, connTimeout,
sendTimeout, recvTimeout)
{
  // constants
  const HTTP_ACTION = "POST";
  var rawResponse;
  if(connTimeout == null){
   connTimeout = 30;
  }
  if(sendTimeout == null){
   sendTimeout = 30;
  }
  if(recvTimeout == null){
   recvTimeout = 30;
  }
  try{
   rawResponse = doHTTPRequest(HTTP_ACTION, url, httpHeaders, postBody,
connTimeout, sendTimeout, recvTimeout);
   return rawResponse;
  } catch(ex) {
   throw new Error(ex);
  }
}
/**
 * @function:
 *   Do DELETE Action using global method: doHTTPRequest.
 *   @param url:
 *      url; type:string
```

```
    * @param httpHeaders:
    *     http headers; type:array
    * @param connTimeout, sendTimeout, recvTimeout.
    *     timeout parameters; type:number
    *   @param errorMsg:
    *     errorMsg; type:string
    * @return:
    *   response; type:string
    *
    */
  function doRESTDeleteRequest(url, httpHeaders, connTimeout, sendTimeout,
recvTimeout)
  {
    // constants
    const HTTP_ACTION = "DELETE";
    var rawResponse;
    if(connTimeout == null){
     connTimeout = 30;
    }
    if(sendTimeout == null){
     sendTimeout = 30;
    }
    if(recvTimeout == null){
     recvTimeout = 30;
    }
    try{
     rawResponse = doHTTPRequest(HTTP_ACTION, url, httpHeaders, null,
connTimeout, sendTimeout, recvTimeout);
     return rawResponse;
    } catch(ex) {
     throw new Error(ex);
    }
  }
```
    b. Save your changes.

3. Add two new parameters to the SMtoRC integration template:

    a. Open the **SMISRegistry.g** form in Database Manager, and go to the **Parameters** tab.

    b. Delete **Global** in Parameter Category Definition, and click **Search**.

    c. Select **SMtoRC** and add the following parameters to the record:

Name	Value	Category	Description
rc.username	admin	General	RC username used for MT
rc.password	admin	General	RC password used for MT

    d. Save the record.

4. Update the **company** table:

    a. Go to **System Definition** > **Tables** > **company**.

    b. Add a new field **rc.synced** to the table, and select **Logical** as its Data type.

    c. Save your changes.

5. Update the **company** format control.

    a. Go to **Tailoring** > **Format Control**, and open the **company** record.

    b. In the **JavaScript** tab, add a new variable:

`vars.$G_RC_integration = vars.$G_system_info.SMtoRC`. See the result as below:

Add	Upd	Del	Dis	Initial
				true

```
1vars.$G_RC_integration = vars.$G_system_info.SMtoRC
```

    c.   Save the record.

6. Modify the **company.g** form:

    a.   Go to **Tailoring** > **Forms Designer**.

    b.   Open the **temp.rc.multitenant** form, click **Design**, and copy the widgets in this form into the **company.g** form, and adjust the widgets properly. When this is done, the **company.g** form looks like this:

    c.   Select the widget of **Company Code** field, and add **[rc.synced]=true** as the Read-Only condition in the Properties section.

    d.   Save the **company.g** form.

7. Update the **save.company** process.

    a.   Go to **Tailoring** > **Document Engine** > **Processes**.

    b.   Search for the **save.company** process.

    c.   On the **RAD** tab, replace the expressions as follows:

Expression	Replacement
`$L.msg=scmsg(3, "ucmdb")`	`if (nullsub(SMtoRC in $G.system.info, false)=true) then ($L.msg=scmsg("ucmdbAndRC_inactivate", "multi-tenant")) else ($L.msg=scmsg(3, "ucmdb"))`
`$L.msg=scmsg(7, "ucmdb")`	`if (nullsub(SMtoRC in $G.system.info, false)=true) then ($L.msg=scmsg("ucmdbAndRC_reactivate", "multi-tenant")) else ($L.msg=scmsg(7, "ucmdb"))`

    d.   Save your changes.

8. Update the **delete.company** process.

    a.   Search for the **delete.company** process.

    b.   Click **RAD**.

    c.   Replace `$L.msg=scmsg(8, "ucmdb")` with `if (nullsub(SMtoRC in $G.system.info, false)=true) then ($L.msg=scmsg("ucmdbAndRC_delete", "multi-tenant")) else ($L.msg=scmsg(8, "ucmdb"))`

    d.   Save your changes.

9. Update the **company** Data Policy:

    a.   Enter **datadict** in the command line.

b. Search for **company**.

c. On the **General** tab, locate **rc.synced** under Field Name, and select **System** as its Usage Type.

d. Save your changes.

Task 4: Manually implement the OO integration enhancement

1. Log on to Service Manager as a System Administrator.

2. Update the cm.update.save process.

    a. Go to **Tailoring** > **Document Engine** > **Processes**.

    b. Open the cm.update.save process.

    c. On the **Initial Expressions** tab, add the following lines:

```
$L.calcfieldvalue=jscall("SMOOAlert.getCalcFieldValue")

if (calc.interval in $L.file=NULL) then ($L.interval=0) else
($L.interval=calc.interval in $L.file)

if (execute.mode in $L.file="byfield" and
$L.calcfieldvalue~=NULL) then (calc.time in
$L.file=$L.calcfieldvalue+$L.interval)

if (current.phase in $L.file~=current.phase in $L.file.save)
then ($oo.changed=true) else ($oo.changed=false)
```

    d. On the **Final Expressions** tab, add the following line:

```
cleanup($oo.changed)
```

    e. Click **Save**.

3. Update the cm.open.save process.

    a. Go to **Tailoring** > **Document Engine** > **Processes**.

    b. Open the cm.open.save process.

    c. On the **Initial Expressions** tab, add the following lines.

```
$L.calcfieldvalue=jscall("SMOOAlert.getCalcFieldValue")

if (calc.interval in $L.file=NULL) then ($L.interval=0) else
($L.interval=calc.interval in $L.file)

if (execute.mode in $L.file="byfield" and
$L.calcfieldvalue~=NULL) then (calc.time in
$L.file=$L.calcfieldvalue+calc.interval in $L.file)
```

    d. Save the record.

4. Update the cm.update and cm.open processes.

    a. Go to **Tailoring** > **Document Engine** > **Processes**.

    b. Open the cm.update and cm.open processes.

    c. On the **Initial Expressions** tab of each process, add the following lines.

```
if (filename($L.file)="cm3r") then
($SMOO=sysinfo.get("environment")#"scguiwweb" and SMOO in
$G.system.info and OOCM in $G.system.info=true)

if (null(execute.mode in $L.file) and filename($L.file)="cm3r")
then (execute.mode in $L.file="noauto")
```

    d. Save the records.

5. Update the cm.open state record.

a. Go to **Tailoring** > **Document Engine** > **States**.

b. Open the cm.open state record.

c. Add the following row to the Non-base methods table.

Display Action	Process Name	Condition
add link	cm.add.link	true

d. Save the record.

6. Update the cm.view state record.

a. Go to **Tailoring** > **Document Engine** > **States**.

b. Open the cm.view state record.

c. Add the following rows to the Non-base methods table.

Display Action	Process Name	Condition
add link	cm.add.link	true
exeooflow	cm.exeooflow	true

d. Save the record.

7. Update the Change Implementation, and Build and Test change phases.

a. Go to **Change Management** > **Changes** > **Change Phases**.

b. Open the **Change Implementation**, and **Build and Test** change phases.

c. On the **Alerts/Open & Close Behavior** tab of each change phase, add the SMOO Auto-Approval, SMOO Auto-Field, and SMOO Auto-Time alerts.

Alerts

SMOO Auto-Approval

SMOO Auto-Field

SMOO Auto-Time

d. On the **Alerts/Open & Close Behavior** tab of each change phase, go to the Alert Controls section, and change **Recalc.** to false.

8. Update the cm3r table.

a. Go to **System Definition** > **Tables** > **cm3r**.

b. Add a new field named **smoo**, and set its data type to **Structure**.

c. From the Fields list, select **smoo**, and click **New field** repeatedly to add the following new fields to the smoo structure.

Name	Data type	Caption
calc.field	Character	Calc Field
calc.interval	Date/Time	Calc Interval
calc.operator	Character	Calc Operator
calc.time	Date/Time	Calc Time
execute.mode	Character	Execute Mode
oo.alert	Logical	OO Alert
scheduled.time	Date/Time	Scheduled Time

The smoo structure should look like the following:
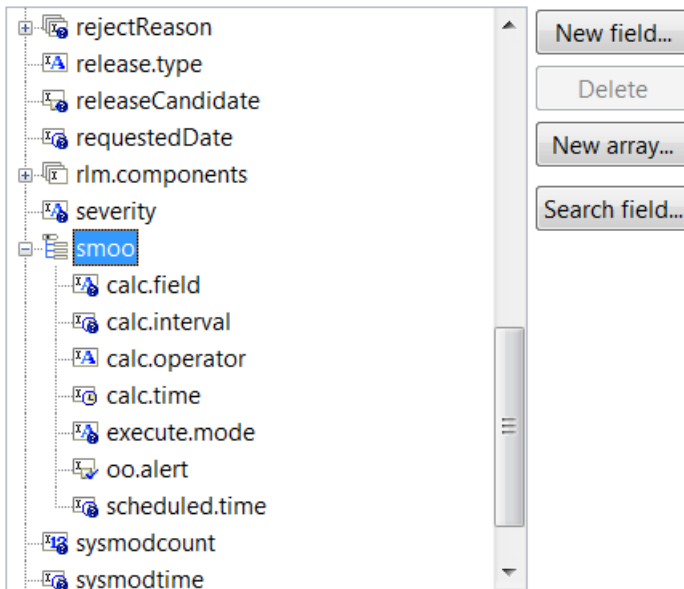
d. Locate the structure named **header**, and click the **number** field under it.

e. In the Aliases section, click **New…** to add a new alias number.vj.ooflow as shown in the following.

General properties

Edit the general properties of the field. Note: only scalar fields can change type.

Name :   number

* Data type: Character
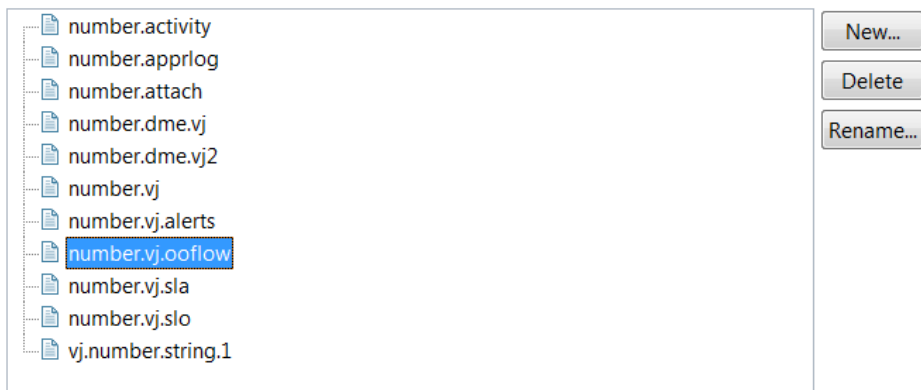
Index: 1

Caption: Change ID

Rename the field

Edit default help on this field

Search for specific help on this field in forms

▾ **Aliases**

Add or delete aliases.

- number.activity
- number.apprlog
- number.attach
- number.dme.vj
- number.dme.vj2
- number.vj
- number.vj.alerts
- number.vj.ooflow
- number.vj.sla
- number.vj.slo
- vj.number.string.1

New…

Delete

Rename…

   f. Save the cm3r table.

9. Update the info table.

   a. Go to **System Definition** > **Tables** > **info**.

   b. Under the SMIS structure field, add a new field **OOCM** with a data type of **Logical**. The info table structure now looks like the following.
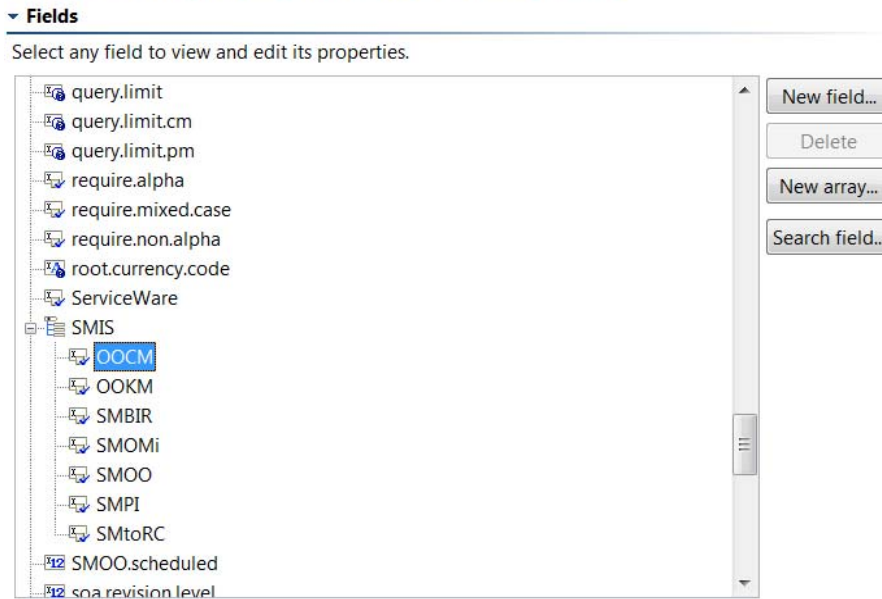
10. Add a display option named db.view_do nothing_openreport.

   a. Go to **Tailoring** > **Tailoring Tools** > **Display Options**.

   b. Add a new display option record as described in the following table.

Field	Value
Screen ID	db.view
Action	do nothing
Unique ID	db.view_do nothing_openreport
GUI option	6
Text Option	6
Bank	1
Balloon Help (If Option < 200)	Open OO Flow Result Report
Default Label	Open Report
Text Alternative	Open Report
Condition	$L.format#"activity" and type in $L.file="OO Response" and SMOO in $G.system.info=true and index("Report Link: http", str(description in $L.filed))>0
RAD <Pre RAD Expressions>	$haslink=false $url=""

| RAD <Post RAD Expressions> | ```
var i=0;
var size =
vars.$L_file.description.getSize();
while( i < size && !vars.$haslink ){
 var des = vars.$L_file.description[i];
 if( des.indexOf("Report Link") != -1 ){
 var i = des.indexOf("http");
 if( i != -1){
   url = des.substr(i);
   vars.$url=url;
   vars.$haslink=true;
 }
 }
 i++;
}
``` |
| RAD<Rad> | ```
RAD Application: us.launch.external
Separate Thread?: $haslink
Names: name
Values: $url
``` |

11. Update the cm.view.display_find display option.
    a. Go to **Tailoring** > **Tailoring Tools** > **Display Options**.
    b. In the **Unique ID** field, enter **cm.view.display_find**, and click **Search**.
    c. Add the following field value.

Field	Value
| RAD <Pre RAD Expressions> | ```
$oo.status=($L.mode="update" or $L.mode="add"
or $L.editable.flag) and (current.phase in
$L.file) isin {"Change Logging", "Change
Review", "Change Assessment & Planning",
"Prepare for Change Approval", "Assess",
"Plan and Design"}
``` |

    d. Save the display option record.
12. Update related forms.
    a. Go to **Tailoring** > **Forms Designer**.
    b. Search for the **temp.edit.ooflow** form, and click **Design**.
    c. Copy all the widgets in this form into somewhere between the Attachments and History groups of each of the following forms:
       - CM.change.logging
       - CM.change.review
       - CM.change.planning
       - CM.change.approvalprep
       - cm3r.release

d. Adjust the widgets in each of the above forms properly. When this is done, the forms look like the following:

e. Search for the temp.readonly.ooflow form, and click **Design**.

f. Copy all the widgets in this form into somewhere between the Attachments and History groups of each of the following forms:

- CM.change.approval

- CM.change.implementation

- CM.change.closure

- cm3r.release.close

g. Adjust the widgets in each of the above forms properly. When this is done, the forms look like the following:

Attachments

OO Flow Links

Select Path | Add Link

Links

Automation Type
- No Automation
- When approved
- Use a fixed date
- Use field in record + interval

Scheduled Time

Change Field

Interval   (+/- hhh hh:mm:ss)

History

13. Update the cm3r format control record.

    a. Go to **Tailoring** > **Format Control**.

    b. Search for and open the **cm3r** record.

    c. Click **Calculations**, and add a new row as follows.

add	update	delete	display	initial	calculation
true	true				$smoov=SMOO in $G.system.info and OOCM in $G.system.info and ((current.phase in $file) isin {"Change Implementation", "Build and Test"} and ($oo.changed=NULL or $oo.changed=true))

    d. Click **Validations**, and add the following validations.

15

Validation	not null(calc.field in $file)
Message	Please provide a Change Field.
Add	execute.mode in $file="byfield" and $smoov
Update	execute.mode in $file="byfield" and $smoov
Set Focus to	calc.field
Message ID	smoo_13

Validation	not null(scheduled.time in $file)
Message	Please provide a Scheduled Time.
Add	execute.mode in $file="byfixedtime" and $smoov
Update	execute.mode in $file="byfixedtime" and $smoov
Set Focus to	scheduled.time
Message ID	smoo_14

Validation	calc.time in $file>tod()
Message	The SMOO Automation Start time must be AFTER the current time. Re-enter the date.
Add	execute.mode in $file="byfield" and $smoov
Update	execute.mode in $file="byfield" and $smoov
Set Focus to	calc.field
Message ID	smoo_15

Validation	calc.time in $file>=planned.start in $file
Message	The SMOO Automation Start time must be AFTER the Planned Start. Re-enter one of the dates.
Add	execute.mode in $file="byfield" and $smoov
Update	execute.mode in $file="byfield" and $smoov
Set Focus to	calc.field
Message ID	smoo_16

Validation	calc.time in $file<=planned.end in $file
Message	The SMOO Automation Start time must be BEFORE the Planned End. Re-enter one of the dates.

Add	execute.mode in $file="byfield" and $smoov
Update	execute.mode in $file="byfield" and $smoov
Set Focus to	calc.field
Message ID	smoo_17

Validation	scheduled.time in $file>tod()
Message	The SMOO Automation Start time must be AFTER the current time. Re-enter the date.
Add	execute.mode in $file="byfixedtime" and $smoov
Update	execute.mode in $file="byfixedtime" and $smoov
Set Focus to	scheduled.time
Message ID	smoo_15

Validation	scheduled.time in $file>=planned.start in $file
Message	The SMOO Automation Start time must be AFTER the Planned Start. Re-enter one of the dates.
Add	execute.mode in $file="byfixedtime" and $smoov
Update	execute.mode in $file="byfixedtime" and $smoov
Set Focus to	scheduled.time
Message ID	smoo_16

Validation	scheduled.time in $file<=planned.end in $file
Message	The SMOO Automation Start time must be BEFORE the Planned End. Re-enter one of the dates.
Add	execute.mode in $file="byfixedtime" and $smoov
Update	execute.mode in $file="byfixedtime" and $smoov
Set Focus to	scheduled.time
Message ID	smoo_17

    e. Save the cm3r format control record.

14. Update the cm3r link.

    a. Go to **Tailoring** > **Tailoring Tools** > **Links**.

    b. Search for and open the cm3r link record.

    c. Add the following rows.

Source Field Name	Target File Name	Target Format Name	Target Field Name	Add Query
$ooflow	OOFlow		path	$query
OOFlow.uuid	OOFlowLinks	OOFlowLinks.cm.detail	OOFlow.uuid	$query
number.vj.ooflow	OOFlowLinks		number	$query
seq.no	OOFlowLinks	OOFlowLinks.cm.detail	seq.no	$query
OOFlow.path	OOFlowLinks	OOFlowLinks.cm.detail	OOFlow.path	$query

d. Click to highlight each of the above new rows, then right-click each row and select **Select Line**. Complete the detail information as listed in the following table.

Field(From/Source)	File(To/Target)	Format(To/Target)	Field(To/Target)	Query	QBE Format	Source Field(Fill To/Post From)	Target Field(Fill from/Post to)
$ooflow	OOFlow		path	$query	$query="path#\""+$ooflow+"\" or uuid#\""+$ooflow+"\""	$ooflow	path
OOFlow.uuid	OOFlowLinks	OOFlowLinks.cm.detail	OOFlow.uuid		$query="OOFlow.uuid = \""+nullsub(cursor.field.contents(),"XXX")+"\" and number = \""+number in $L.file+"\" and type = \"cm3r\""	OOFlow.uuid	OOFlow.uuid
					$action="update"	number	number
					$assets=assets in $L.file		

number.vj.ooflow	OOFlowLinks		number	$query	$query="type = \"cm3r\""	number.vj.ooflow	number
seq.no	OOFlowLinks	OOFlowLinks.cm.detail	seq.no	$query	$query="seq.no = \""+nullsub(cursor.field.contents(), "XXX")+"\" and number = \""+number in $L.file+"\" and type = \"cm3r\"" $action="update" $L.phase=current.phase in $L.file $assets=assets in $L.file	seq.no	seq.no
OOFlow.path	OOFlowLinks	OOFlowLinks.cm.detail	OOFlow.path	$query	$query="OOFlow.path = \""+nullsub(cursor.field.contents(), "XXX")+"\" and number = \""+number in $L.file+"\" and type = \"cm3r\""	OOFlow.path OOFlow.uuid	OOFlow.path OOFlow.uuid

    e.   Save the cm3r link record.

Configuration

After installing the CCRM integration package, System Administrators need to set up their production systems for end users to use the features.

Setting up the RC integration

The CCRM integration requires bidirectional data exchange between Service Manager and RC.

To correctly set up your production system for the CCRM solution to work, you are recommended to complete the following tasks:

- Task 1: Configure Service Manager to use SSL-based Trusted Sign-On and LW-SSO
- Task 2: Configure LW-SSO in RC
- Task 3: Configure the RC adapter
- Task 4: Add and enable an RC integration
- Task 5: Test the environment setup

Task 1: Configure Service Manager to use SSL-based Trusted Sign-On and LW-SSO

You are recommended to enable LW-SSO for the RC integration so that you can update or approve change requests from RC. To enable LW-SSO for the RC integration, complete the following procedures on the Service Manager side:

- Configuring Service Manager to Use SSL-based Trusted Sign-On
- Installing and Configuring the Symphony Adapter
- Configuring the Service Manager Web Client for LW-SSO Support

These procedures are documented in the *Configuring HP Service Manager to Use the SSL-based Trusted Sign-On and LW-SSO* white paper (in PDF format). To download this document, visit the HP manual site at:

http://h20230.www2.hp.com/selfsolve/manuals

Task 2: Configure LW-SSO in RC

To enable LW-SSO for the RC integration, configure RC to support LW-SSO. For detailed steps, see the following topic in the Service Manager 9.20 help:

- *Configure LW-SSO in Release Control*

NOTE: The last step in this topic is to create an RC user that has the same user account and password that are used to log on to Service Manager so that the Single Sign-on works.

Task 3: Configure the RC adapter

The RC solution requires an adapter configured in RC for Service Manager. For detailed steps, see the following topic in the Service Manager 9.20 help:

- *Configure the RC adapter*

NOTE: Pay attention to the following screen prompts when running SdiConfigurer.bat:

- Is HP's Lightweight Single Sign-On (LW-SSO) used?

  Type **y** and press ENTER. If you give a false answer here, you will need to start over by running the SdiConfigurer.bat file again.

- Is https required in order to access wsdl?

  Type **n** and press ENTER.

Task 4: Add and enable an RC integration

To use the enhanced RC integration, you must add and enable an RC integration instance in Service Manager.

For detailed steps, see the following topic in the Service Manager 9.20 help:

- *Add a Release Control integration*

NOTE: The CCRM integation package adds two new parameters to the out-of-box RC integration template (SMtoRC) in Service Manager 9.20. Set the parameters as described in the following table.

Parameter	Value	Example
rc.username	User name of the RC user account that manages multitenancy in RC.	admin
rc.password	Password of the RC user account that manages multitenancy in RC.	admin

Task 5: Test the environment setup

After you have completed the above configuration tasks, you may want to test your environment to see if it is correctly set up for the RC integration. This task includes:
- Subtask 1: Create test user accounts
- Subtask 2: Test LW-SSO from Service Manager to RC
- Subtask 3: Test LW-SSO from RC to Service Manager

Subtask 1: Create test user accounts

1. Create a new operator in Service Manager with the same username as a user account in RC (for example, admin), and grant enough rights to this account so that this account can create RFCs.

2. Create a new user (for example, **approver**) in Service Manager with the necessary rights to approve change requests.

3. Add the user (for example, **approver**) to the Approval list.

   a. Go to **Change Management** > **Maintenance** > **Approvals**.

   b. Type **Approval** in the Name field. Add the new operator with approval rights to the list, or replace the existing one.

4. Create a new user in RC with the same username (for example, **approver**) and grant it rights to approve change requests.

Subtask 2: Test LW-SSO from Service Manager to RC

1. Log on to the Service Manager Web client with user account **admin**.

2. Create a change request, and change its phase to Change Approval.

   NOTE: Make sure that the Affected CI of the change is from HP Universal CMDB (UCMDB).

3. Click **Change Management** > **Change Calendar**.

   If the RC Change Calendar displays with the RC log-in prompt bypassed, LW-SSO is successfully enabled for integration from Service Manager to RC.

Subtask 3: Test LW-SSO from RC to Service Manager

1. Log on to RC as **approver**.

2. Approve the new change request synchronized from Service Manager.

   If the change request can be approved with the Service Manager log-in prompt bypassed, LW-SSO is successfully enabled for integration from RC to Service Manager.

Setting up the OO integration

If you are planning to use both the RC and OO integrations in your system, you are recommended to set up the RC integration first and then the OO integration. This is because the OO integration also needs SSL configuration, which means both integrations should share the same trust store in Service Manager. Perform the following tasks to set up the OO integration in your production system:

- Task 1: Enable SSL in Operations Orchestration and Service Manager
- (Optional)Task 2: Configure Service Manager and OO to support LW-SSO
- Task 3: Add and enable an OO integration

   After you have completed these tasks, the enhanced OO integration is ready for use.

Task 1: Enable SSL in Operations Orchestration and Service Manager

When the Service Manager server is accessing OO as a client, OO requires that the servers communicate over Secure Sockets Layer (SSL) to ensure data security. You must create a root CA and self-signed certificate in OO and then import them to Service Manager.

For detailed steps to enable SSL in OO and Service Manager, see the following topic in the Service Manager 9.20 help:
- *Example: Enabling SSL in Operations Orchestration and Service Manager*
NOTES:
- Follow the steps in this topic to configure SSL in OO.
- If you are using the OO integration separately, follow the steps in this topic to configure SSL in Service Manager.
- If you have already enabled SSL and LW-SSO for the RC integration in your system, make the following adjustment to the steps in the "configure SSL in Service Manager" section as described in the following table:

Parameter	Value
<keystore_file>	ca.keystore
	Note: When running the commands, replace <keystore_file> with this value. This is the server trust store file you used when setting up the RC integration. You need to import the root CA and self-signed certificate from OO to this keystore, instead of a new trust store file.
<store password for keystore_file>	<root CA keystore pass phrase>
	Note: When running the commands, replace <store password for keystore_file> with the pass phrase you specified for ca.keystore when setting up the RC integration.

(Optional) Task 2: Configure Service Manager and OO to support LW-SSO

This task is optional for the OO integration. You can configure OO and Service Manager to support Lightweight Single Sign-On (LW-SSO) so that the OO log-in prompts are bypassed when a Service Manager user attempts to view OO flow execution reports in OO.

For information about how to configure LW-SSO in OO and Service Manager, see the following topics in the Service Manager 9.20 help:

- *Configure LW-SSO in Operations Orchestration*
- *Configure LW-SSO in Service Manager*

Task 3: Add and enable an OO integration

After you have completed the required SSL and the optional LW-SSO configuration tasks, you are ready to add and enable an OO integration instance in Service Manager.

For detailed steps, see the following topic in the Service Manager 9.20 help:

- *Add an Operations Orchestration integration*

NOTE: The CCRM integration package adds a new parameter **OOCM** to the out-of-box OO integration template (named SMOO) in Service Manager 9.20. Make sure to set it to true.

Parameter Name	Description
OOCM	Set to true/false to enable/disable the visibility of the OO integration instance in Change Management (default: true).

The enhanced OO integration is now set up in your system and ready for use.

Administration

This section describes how System Administrators can manage multitenancy provided by the CCRM integration after the installation and configuration tasks are completed.

The SMtoRC integration enhancement supports multitenancy, which allows users to see change records in RC for their authorized tenants only. For example, when a user from Company A logs on to Service Manager, the user can only view (from changes, interactions, incidents, problems, or the System Navigator) the data of Company A's account in RC through Change Calendar.

> **NOTE**: In Service Manager, some users may be able to view data of more than one company, but in Change Calendar they can view the data of only one company at one time, which is determined by the user's Default Company defined in Service Manager.

To manage multitenancy for the RC integration, you may need to perform the following tasks:

- Task 1: Enable multitenancy for Service Manager and UCMDB
- Task 2: Add or update a company record and deactivate tenancy
- Task 3: Re-synchronize a company record with RC

Task 1: Enable multitenancy for Service Manager and UCMDB

1. Log on to Service Manager as a System Administrator.
2. Go to **System Administration** > **Base System Configuration** > **Miscellaneous** > **System Information Record**.
3. Select the **Run in Multi-Company Mode** check box under the **General** tab.
4. Select the **HP Universal CMDB** check box under the **Active Integrations** tab.
5. Provide values for the following fields:

Field	Description	Example
UCMDB webservice URL	Type the URL to the UCMDB Web Service API. The URL has the following format: `<UCMDB server name>:<port>/axis2/services/ucmdbSMService` Replace `<UCMDB server name>` with the host name of your UCMDB server, and replace `<port>` with the communications port your UCMDB server uses.	`<UCMDB.hp.com>:8080/axis2/services/ucmdbSMService`
UserId	The user ID of the user account that has permission to access the UCMDB Web Service API.	admin
Password	The password of the user account that has permission to access the UCMDB Web Service API.	admin
Multi-	Type the URL to the UCMDB Web Service	`<UCMDB.hp.com>:<8080>/axis2/services/ucmdbManagementSe`

tenant UCMDB webservice URL	API. The URL has the following format: *<UCMDB server name>*`:`*<port>*`/axis2/services/uc mdbManagementService` Replace *<UCMDB server name>* with the host name of your UCMDB server, and replace *<port>* with the communications port your UCMDB server uses.*<UCMDB server name>*`:`*<port>*`/axis2/services/uc mdbManagementService`	`rvice`
UserId	The user ID of the user account that has permission to access the multitenant UCMDB Web Service API.	sysadmin
Password	The password of the user account that has permission the access the multitenant UCMDB Web Service API.	sysadmin

NOTE: In the **UCMDB webservice URL** and **Multi-tenant UCMDB webservice URL** fields, you do not need to enter `http://` or any trailing slashes. You only need to provide the host name, port, and the Web Services API to the HP Universal CMDB server.

6. Click **Save**.

Task 2: Add or update a company record and deactivate tenancy

1. Log out, and then log on to Service Manager again as a System Administrator after finishing Task 1.

2. To add a company record into tenant:

    a. Go to **System Administration** > **Base System Configuration** > **Companies**.

    b. Provide values for the fields as necessary.

    c. Select **Yes** in the Show Company in Multi-Company Lists field.

    d. Provide a value for the UCMDB Customer ID field.

    e. Click **Add**. A dialog box opens, asking "Do you want to synchronize with UCMDB and Release Control?"

    • If you select **Yes**, a company record synchronized with UCMDB and RC is added. On the Company Information page, you can see the read-only checkboxes of **Synched with UCMDB** and **Synched with RC** are selected.

    • If you select **No**, a company record is added but without being synchronized with UCMDB and RC. You can synchronize the record later.

    NOTES:

- Each added company record has a unique Customer ID. A new company record given an existing Customer ID cannot be successfully added.

- The company information synchronized with RC includes Customer ID and Company Code. Once synchronized, the Company Code of the record becomes read-only.

- If you selected **Yes** and RC failed to synchronize the record due to some reasons, you can see the **Synched with RC** check box is unselected on the Company Information page. To synchronize the record to RC, you need to update and save the record.

3. To update a company record:
   a. Search the company record list and select a record you want to update.
   b. Update the company information as necessary.
   c. If you want to enable multitenancy for this company record, make sure to select **Yes** in the **Show Company in Multi-Company Lists** field.
   d. Click **Save**. The result could be one of the following:
      - If both **Synched with UCMDB** and **Synched with RC** are unselected, a dialog box opens, asking "Do you want to synchronize with UCMDB and Release Control?" Click **Yes**. The record is then automatically synchronized with UCMDB and RC.
      - If **Synched with UCMDB** is selected while **Synched with RC** is unselected, no dialog box opens. The record is automatically synchronized with RC if it is not successfully synchronized with RC yet.
      - If both **Synched with UCMDB** and **Synched with RC** are selected, the record is repeatedly synchronized to RC and no dialog box opens. You can see the modification you made to the record in Service Manager. In this case, since the Company Code is read-only, the record is not updated in RC.

   NOTES:
   - To synchronize a record with RC, it is mandatory to click **Yes** in the Question dialog box.
   - You can view the synchronizing results of company data in a log file. For the location of the file, go to **Tailoring** > **Integration Manager** and select the SMtoRC integration. The Log File Directory field on the **Integration Instance Information** page indicates the location of the log file.
   - It may take some time before you can see the check boxes selected, as it takes time to implement a schedule.

4. To deactivate a company's tenancy:
   a. Search the company record list and select a company record you want to deactivate.
   b. Select **No** in the **Show Company in Multi-Company Lists** field.
   c. Click **Save**.
   d. Click **Yes** in the question dialog box.

Task 3: Re-synchronize a company record with RC

In some cases, you need to re-synchronize a company record with RC. For example, if a company record was synchronized to RC and was deleted later in RC, you can re-synchronize the record to generate a new record in RC; if the tenants information in Service Manager and RC does not match, you can re-synchronize the information to RC.

1. Open a company record that was synchronized with RC.
2. Click **Re-Synch** next to the **Synched with RC** check box. The company record is re-synchronized.

   NOTE: The **Re-Synch** button for the **Synched with RC** check box is disabled unless you selected **Yes** in the **Show Company in Multi-Company Lists** field and the record was successfully synchronized with RC.

June 2010