

**Table of Contents**

**Audit Manager**

[1 How to enable EPMJOB Delete event 3](#_Toc48056602)

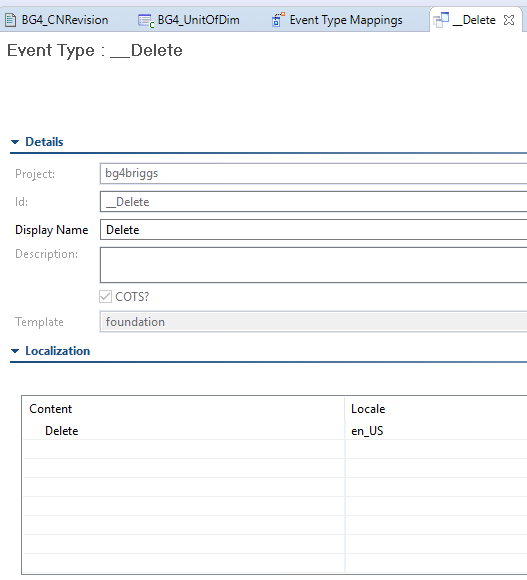
[2 Custom log extension example 9](#_Toc48056603)

[3 How to enable Item Revision Delete event 13](#_Toc48056604)

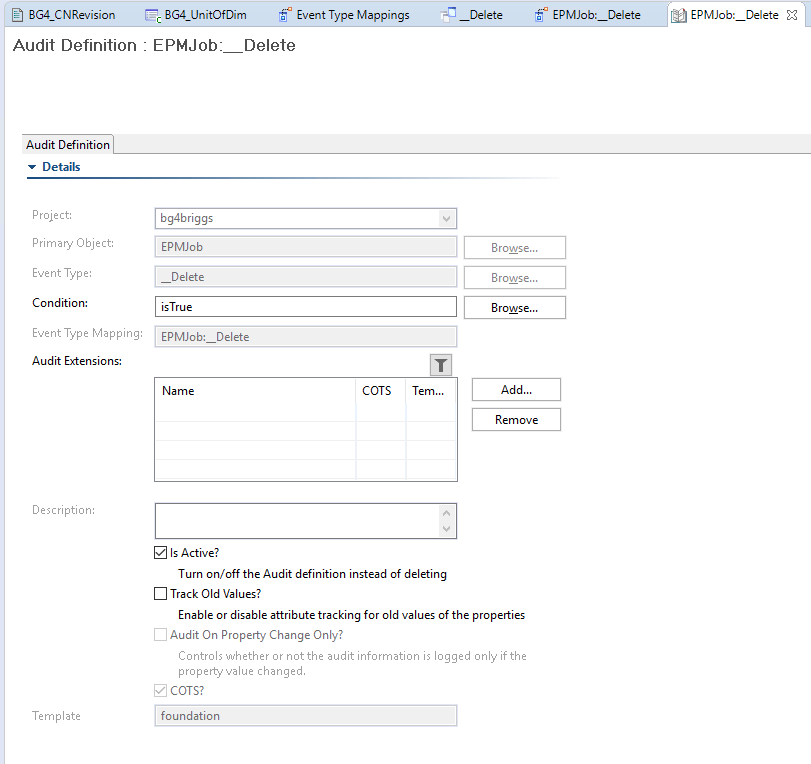
[4 Item Revision delete audit Report. 14](#_Toc48056605)

# How to enable EPMJOB Delete event

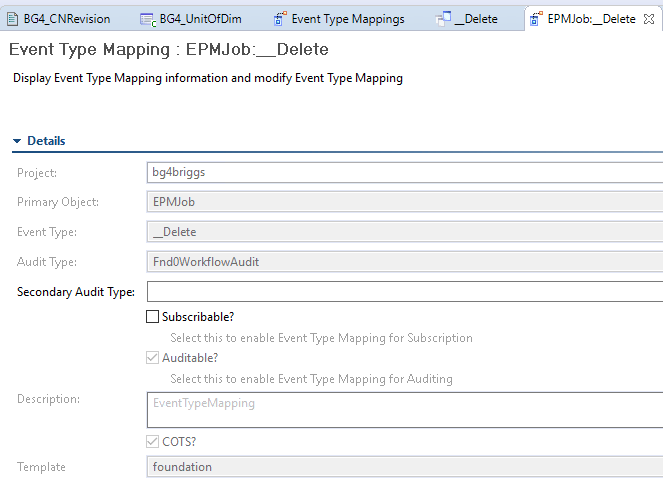
1. Event Type:

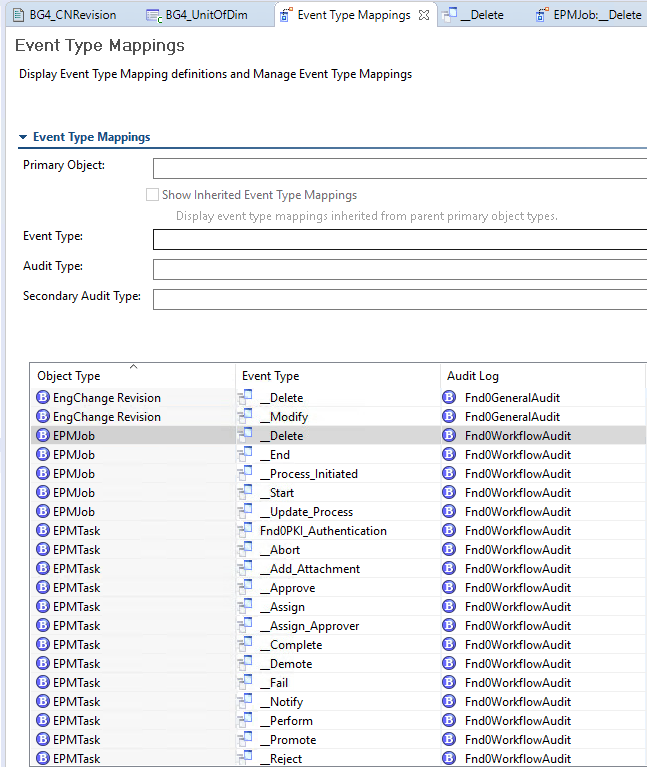


1. Audit Definitions:



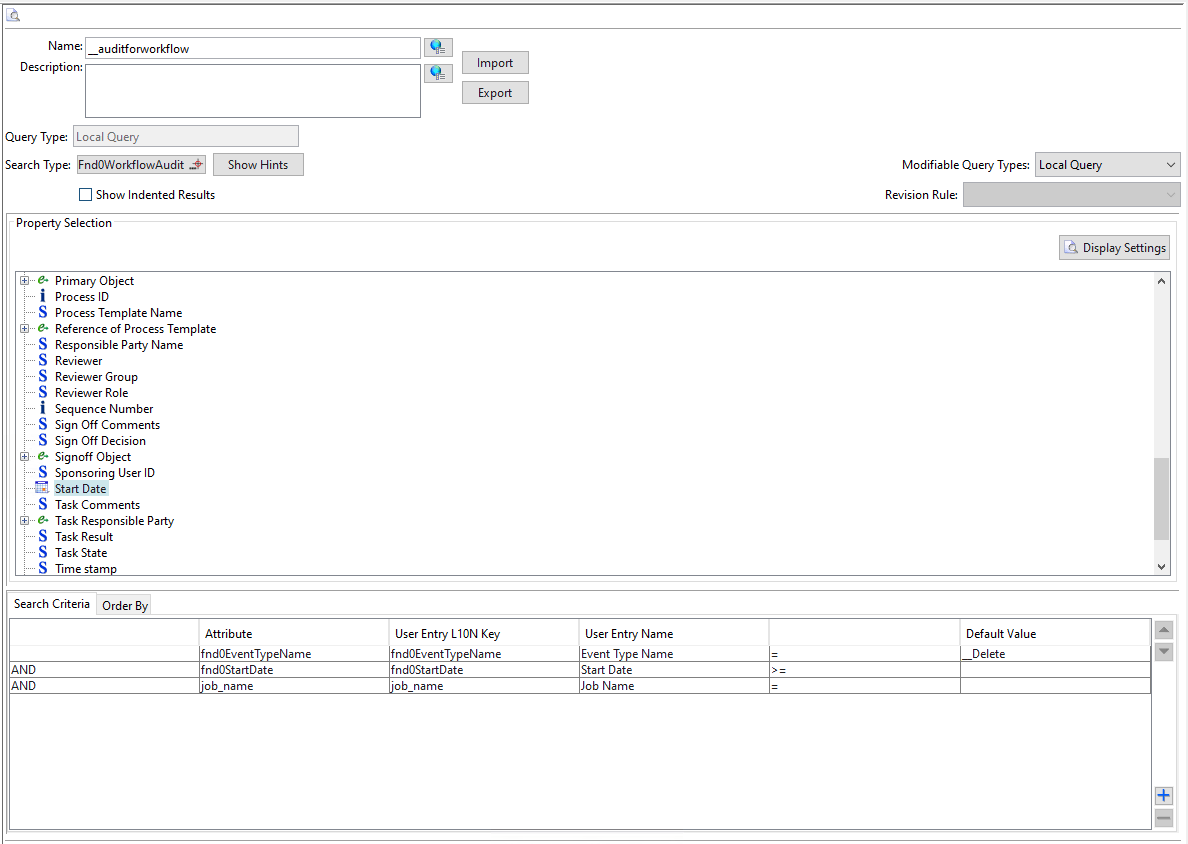
1. Event Type Mapping:



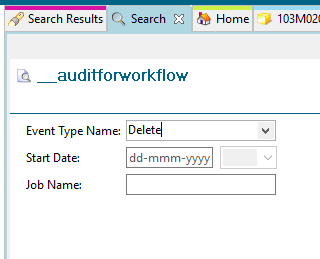


**Extract EPMJob delete report/log file.**

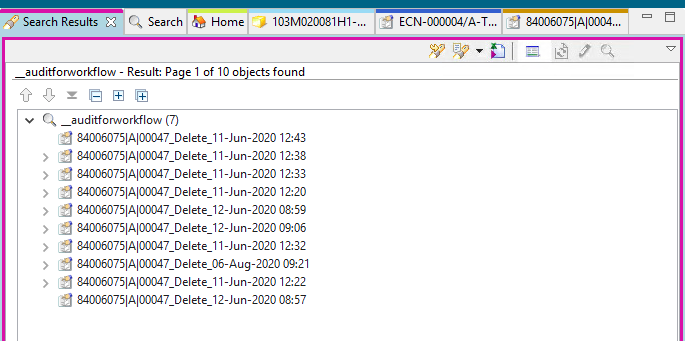
1.Create custom query as shown in below image.

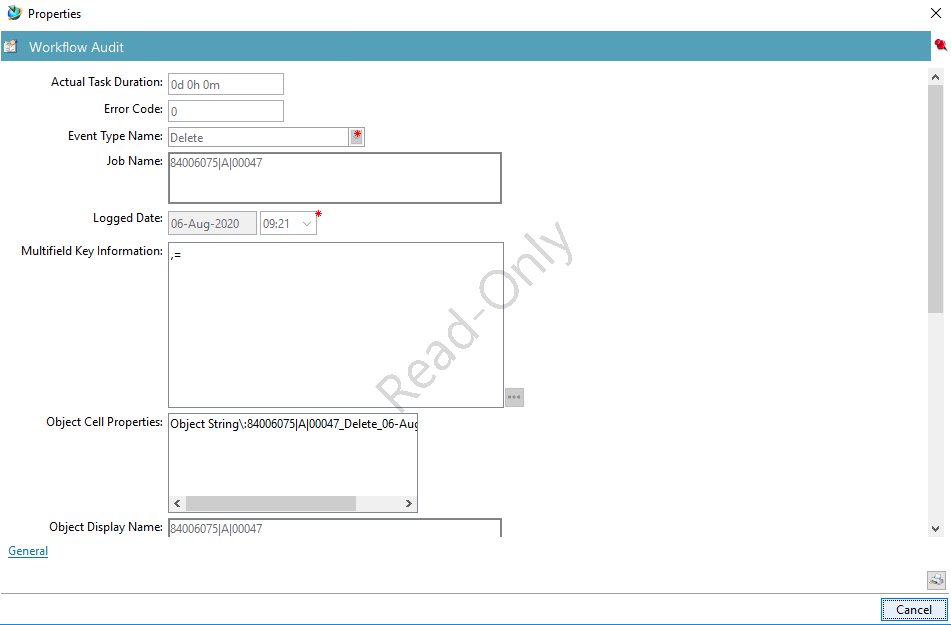


2.Open the custom query to execute in RAC.



3.Enter the JOB name to find all EPMJob delete report.





# Custom log extension example

In this example, a log extension named **BG4\_CustomAuditLog** is attached to the **EPMTask:\_Delete:isTrue** audit definition object. You can create log extensions, using Business Modeler IDE.

1. In Business Modeler IDE, create a template project if you have not already created one.
2. Define the **BG4\_CustomAuditLog** extension.
   1. [Create a library](https://docs.sw.siemens.com/en-US/product/282219420/doc/PL20200109161503476.itk/html/add_a_new_library) named **BG4\_BRIGGS**.
   2. Expand the project and the **Rules**→**Extensions** folders.
   3. Right-click the **Extensions** folder and choose **New Extension Definition**.

The New Extension Definition wizard runs.

* 1. Perform the following in the **Extension** dialog box:
     1. In the **Name** box, type **BG4\_CustomAuditLog**.
     2. In the **Language** box, select **CPlusPlus**.
     3. In the **Library** box, select **BG4\_BRIGGS**.
     4. Click **Add** to the right of the **Availability** table and perform the following in the **Extension availability** dialog box:
        1. In the **Business Object Name** box, select **Fnd0AuditDefinition**.
        2. In the **Operation Name** box, select **fnd0writeAuditLog**.
        3. In the **Extension Point** box, select **PostAction**.
        4. Click **Finish** in the **Extension availability** dialog box

1. Attach the **BG4\_CustomAuditLog** extension to the **EPMTask:\_Delete:isTrue** audit definition object.
   1. From the **Extensions** view, open **Audit Manager**→**Audit Definitions**. Right-click the **EPMTask:\_Delete:isTrue** audit definition and choose **Open**.

The **Audit Definition: EPMTask:\_Delete:isTrue** view is displayed.

* 1. Click **Add** to the right of the **Audit Extensions** table and select the **BG4\_CustomAuditLog** extension from the **Find an Audit Extension** dialog box.
  2. To save the changes to the data model, choose **BMIDE**→**Save Data Model**.

1. Implement the **BG4\_CustomAuditLog** extension.
   1. Right-click the **BG4\_CustomAuditLog** extension and choose **Generate extension code**.

The extension boilerplate code is generated into a **BG4\_CustomAuditLog.cxx** C++ file and a **BG4\_CustomAuditLog.hxx** header file. To see these files, open the project in the **Navigator** view and browse to the **src\server\BG4\_BRIGGS** directory.

Note:

You might need to right-click in the view and choose **Refresh** to see the files that were generated.

* 1. Open the **BG4\_CustomAuditLog.cxx** file in a C/C++ editor and add your custom business logic.

Generated code has method signature as **int BG4\_CustomAuditLog ( METHOD\_message\_t \*msg, va\_list args )**

Replace the method signature as below in **BG4\_CustomAuditLog.cxx** and **BG4\_CustomAuditLog.hxx**

int BG4\_CustomAuditLog  
( tag\_t  /\*targetObjTag\*/,  
int /\*secondaryObjectCount\*/,  
tag\_t\* /\*secondaryObjectTags\*/,  
char\*\* /\*secondaryQualifiers\*/,  
Char\* /\*eventType\*/,  
int paramCount,  
char\*\* paramNames,  
char\*\* paramValues,  
int /\*errorCode\*/,  
const char\* /\*errorMessage\*/,  
tag\_t primaryAuditBOTag )

The following is a sample file:

int BG4\_CustomAuditLog(tag\_t targetObjTag,

int secondaryObjectCount,

tag\_t\* secondaryObjectTags,

char\*\* secondaryQualifiers,

char\* eventType,

int paramCount,

char\*\* paramNames,

char\*\* paramValues,

int /\*errorCode\*/,

const char\* /\*errorMessage\*/,

tag\_t primaryAuditBOTag)

{

TC\_write\_syslog(" Entering BG4\_CustomAuditLog\n");

int ifail = ITK\_ok;

//tag\_t job\_tag = NULLTAG;

char\* signoff = NULL;

tag\_t signoffTag = NULLTAG;

tag\_t auditClassId = NULLTAG;

tag\_t signoffClassId = NULLTAG;

char\* audit\_class\_name = NULL;

char\* signoff\_class\_name = NULL;

logical isProp = false;

char\* propVal = 0;

char\* targetObjClassName = NULL;

tag\_t auditClassId1 = NULLTAG;

TC\_write\_syslog(" paramCount : %d\n", paramCount);

TC\_write\_syslog(" Secondary Object count : %d\n", secondaryObjectCount);

if (paramCount == 0)

{

//return ITK\_ok;

}

for (int i = 0; i < paramCount; i++)

{

TC\_write\_syslog(" paramNames : %s\n", paramNames[i]);

if (strcmp(paramNames[i], "signoff") == 0)

{

signoff = paramValues[i];

break;

}

}

char\* uid = NULL;

if (targetObjTag!=NULLTAG)

{

ifail = POM\_class\_of\_instance(targetObjTag, &auditClassId1);

ifail = POM\_name\_of\_class(auditClassId1, &targetObjClassName);

TC\_write\_syslog(" Target Object Class Name : %s\n", targetObjClassName);

ITK\_CALL(POM\_tag\_to\_uid(targetObjTag, &uid));

TC\_write\_syslog(" Target Object UID : %s\n", uid);

tag\_t roottask = NULLTAG;

char\* taskUI = NULL;

EPM\_ask\_root\_task(targetObjTag,&roottask);

char\* objName = NULL;

char\* objString = NULL;

AOM\_UIF\_ask\_value(targetObjTag,"object\_name",&objName);

TC\_write\_syslog(" Object Name : %s\n", objName);

AOM\_UIF\_ask\_value(targetObjTag, "object\_string", &objString);

TC\_write\_syslog(" Object String : %s\n", objString);

if (roottask!=NULLTAG)

{

ITK\_CALL(POM\_tag\_to\_uid(roottask, &taskUI));

TC\_write\_syslog(" Root Task Object UID : %s\n", taskUI);

int n\_subtasks = 0;

tag\_t\*subtasks = NULL;

EPM\_ask\_sub\_tasks(roottask, &n\_subtasks, &subtasks);

TC\_write\_syslog(" SubTask(S) Count : %d \n", n\_subtasks);

}

else

{

TC\_write\_syslog("ROOT TASK IS NULL \n");

}

}

if (primaryAuditBOTag != NULLTAG)

{

ifail = POM\_class\_of\_instance(primaryAuditBOTag, &auditClassId);

ifail = POM\_name\_of\_class(auditClassId, &audit\_class\_name);

TC\_write\_syslog(" audit\_class\_name : %s\n", audit\_class\_name);

ifail = AOM\_set\_value\_string(primaryAuditBOTag, "fnd0Reason", uid);

}

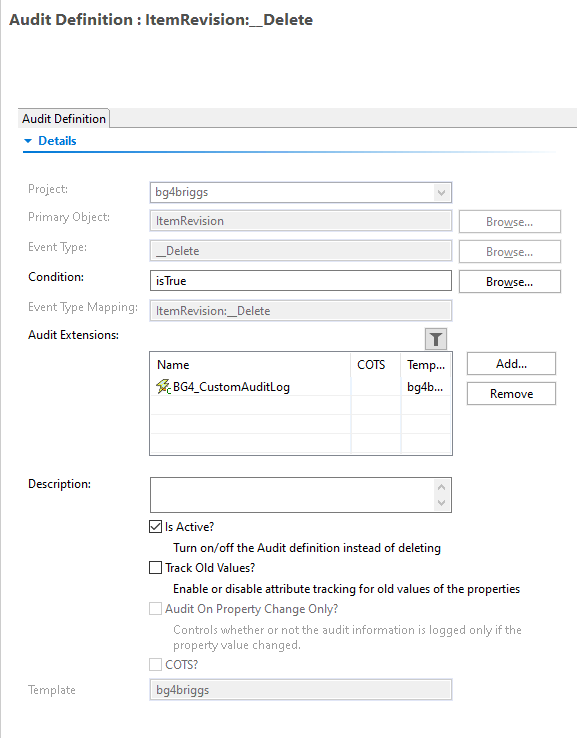
TC\_write\_syslog(" Exiting BG4\_CustomAuditLog\n");

* 1. Open the **BG4\_CustomAuditLog.hxx** file and update it with the header information.

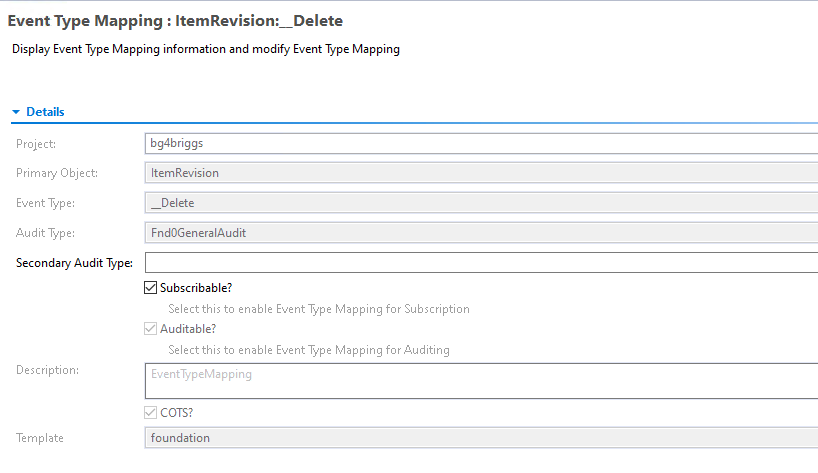
1. Build the library (**libBG4\_Briggs.dll**file) for the **BG4\_CustomAuditLog** extension.
2. (For hot deploy only) Copy the **libBG4\_Briggs.dll**file to the *TC\_ROOT***\bin** directory.

# How to enable Item Revision Delete event

1. Create custom audit definition.



1. OOTB Event type Mapping.



# Item Revision delete audit Report

