AUTOMATION OF HCM EXTRACT

INTRODUCTION

HCM Extract is a powerful tool within the HCM application that is used for reporting and outbound integration. It generates raw XML as well as providing the capability to integrate with BI Publisher to generate formatted output in a variety of supported formats.

When used as part of an integrated process it may be required to invoke the HCM Extract automatically as part of an automated flow. In addition once the extract is complete it is likely that the output needs to be retrieved for subsequent processing.

This document provides details of how this automation can be achieved using existing web services that are available in R7.

HCM EXTRACT WITHIN THE PAYROLL FLOW ENGINE

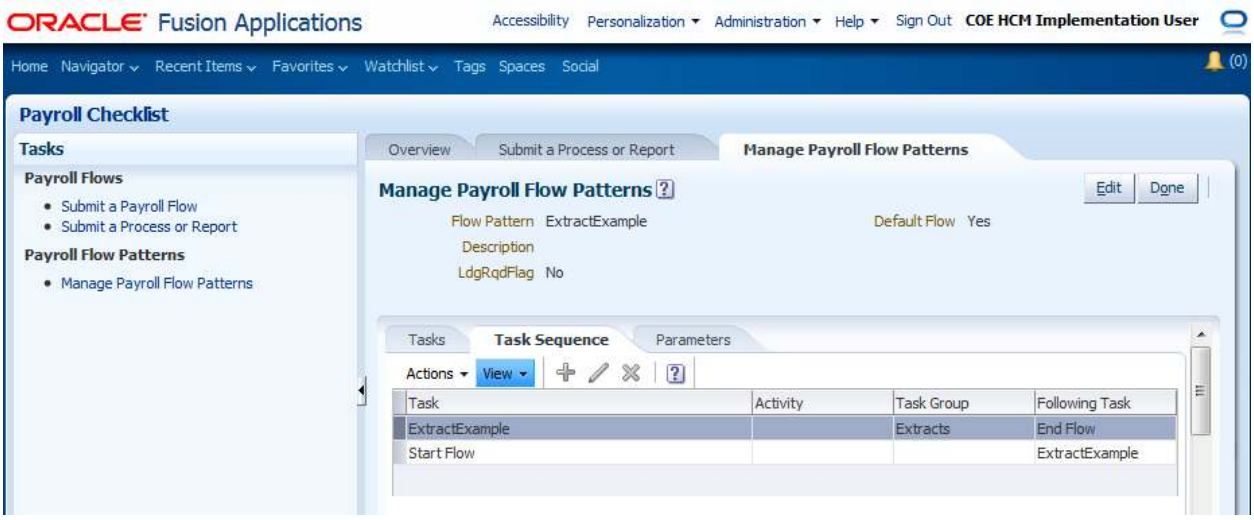
The Payroll Flow Engine is a generic processing engine that has the capability to invoke a wide range of different types of processes. Examples of its use include various processes for payroll accounting, payroll calculations, payments, statutory reporting and inbound data loading. In addition it is the mechanism by which HCM Extracts are invoked.

When an HCM Extract is defined it generates a Flow object in the Payroll Flow data model with the same name as given to the HCM Extract itself. This flow is used to control the processing of the HCM extract including setting any parameters and handling the individual steps within the flow. By default there is just one step (Task) within the flow, but if required additional tasks can be added. When a specific HCM Extract flow is invoked a Flow Instance is created.

The ability to manage the invocation of a Flow and the Tasks within it is the key to being able to automate a process through the Payroll Flow Engine, and hence the way HCM Extract can be automated. The next few sections will discuss this in more detail.

For the purposes of this discussion consider a simple HCM Extract definition called **ExtractExample.**

When **ExtractExample** was saved it created a Payroll Flow of type EXTRACT called **ExtractExample.** In addition a Task by the same name was created and was included in the Task Sequence for the flow as shown below:



INVOKING HCM EXTRACT

A Payroll Flow can be invoked from outside of Fusion using the FlowActionsService, hence this will be the mechanism by which HCM Extracts can be automated.

The method to invoke the HCM Extract is FlowActionsService.submitFlow.

This has the following parameters:

Parameter Name	Description
Flow Name	The name of the HCM Extract
Flow Instance Name	The identifier of the particular instance of the extract to be run
Legislative Data Group Name	The name of the legislative data group
Recurring Flag	Set to “false”
Parameters	Name value pairs of parameters. For default HCM Extract only Effective Date is required. See example below.

EXAMPLE

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:typ="http://xmlns.oracle.com/apps/hcm/processFlows/core/flowActionsService/types/"
xmlns:flow="http://xmlns.oracle.com/apps/hcm/processFlows/core/flowControllerService/">
  <soapenv:Header/>
  <soapenv:Body>
    <typ:submitFlow>
      <typ:flowName>ExtractExample</typ:flowName>
      <typ:flowInstanceName>ExtractExample_Run1</typ:flowInstanceName>
      <typ:legislativeDataGroupName>US Legislative Data Group</typ:legislativeDataGroupName>
      <typ:recurringFlag>false</typ:recurringFlag>
      <typ:parameterValues>
        <flow:ParameterName>Effective Date</flow:ParameterName>
        <flow:ParameterValue>2014-01-01</flow:ParameterValue>
      </typ:parameterValues>
```

```
    </typ:submitFlow>
  </soapenv:Body>
</soapenv:Envelope>
```

MONITORING HCM EXTRACT

When the HCM Extract is invoked it executes each of the Tasks according to the Flow task sequence. In this case there is only one task to perform. Whilst the task is being performed its status is maintained by the Payroll Flow Engine and it is possible to monitor the changes in the status using the FlowActionsService.getFlowTaskInstanceStatus method.

The parameters for this method are as follows:

Parameter Name	Description
Flow Instance Name	The identifier of the particular instance of the extract to be run
Flow Task Instance Name	The name of the specific task to monitor
Legislative Data Group Name	The name of the legislative data group

The possible status values that could be returned are as follows:

Status Returned	Meaning
COMPLETED	Complete
ERRORED	Error
ESS_CHILD_JOB_COMPLETED	ESS Child Job Completed
ESS_CHILD_JOB_NOT_FOUND	ESS child job not found
ESS_CHILD_JOB_SUBMITTED	ESS child job has been submitted
ESS_CHILD_JOB_SUB_ERROR	ESS Child Job submission error
ESS_MT_SERV_NOT_FOUND	ESS Metadata service not found
ESS_PARENT_JOB_SUBMITTED	ESS Parent Job Submitted
ESS_PARENT_JOB_SUB_ERROR	ESS Parent Job Submission error
ESS_RT_SERV_NOT_FOUND	ESS run time service not found
EXECUTION_STARTED	Flow Task Execution Started
FUNCTIONAL_ERROR	Flow task functional error
IN_PROGRESS	In Progress
IN_PROGRESS_IDLE	In Progress With Manual Intervention Required
MARKED_FOR_RETRY	Marked for Recalculation
NOT_STARTED	Not Started
PUBLISHED	Published
ROLLEDBACK	Rolled Back
SCHEDULED	Scheduled
SKIPPED	Skipped

EXAMPLE

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:typ="http://xmlns.oracle.com/apps/hcm/processFlows/core/flowActionsService/types/">
  <soapenv:Header/>
  <soapenv:Body>
    <typ:getFlowTaskInstanceStatus>
      <typ:flowInstanceName>ExtractExample_Run1</typ:flowInstanceName>
      <typ:flowTaskInstanceName>ExtractExample</typ:flowTaskInstanceName>
      <typ:legislativeDataGroupName>US Legislative Data Group</typ:legislativeDataGroupName>
    </typ:getFlowTaskInstanceStatus>
```

```
  </soapenv:Body>
</soapenv:Envelope>
```

RETRIEVING HCM EXTRACT XML

Once the HCM Extract flow reaches COMPLETED status the raw XML can be downloaded. We recommend that this feature is only used for relatively small extracts as the XML data set is verbose and therefore can be costly to process.

The payrollProcessingActionService.fetchExtractOutput method can be used to download the raw XML.

The parameters for this method are as follows:

Parameter Name	Description
Flow Instance Name	The identifier of the particular instance of the extract to be run
Flow Task Name	The name of the specific task to monitor
Mode	Set to "FLOW"

EXAMPLE

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:typ="http://xmlns.oracle.com/apps/hcm/batchProcesses/core/batchProcessesCoreService/types/">
  <soapenv:Header/>
  <soapenv:Body>
    <typ:fetchExtractOutput>
      <typ:instanceName>ExtractExample_Run1</typ:instanceName>
      <typ:taskName>ExtractExample</typ:taskName>
      <typ:mode>FLOW</typ:mode>
    </typ:fetchExtractOutput>
  </soapenv:Body>
</soapenv:Envelope>
```

This service call returns a String **result** attribute that contains the full XML generated by the extract.

HCM EXTRACT BI REPORT OUTPUT LOCATION

When an HCM Extract is associated with a BI template then the Payroll Flow will ensure that the BI report is generated and written to the BI Server.

There is a payroll action parameter named BI_OUTPUT_SIZE that determines whether the generated BI report is written as an Attachment to WebCenter Server (UCM). The action parameter value is the maximum number of bytes that file can be before it is no longer stored in UCM. If the file is stored in UCM it appears in the Payroll Checklist Results UI and can be viewed in downloaded from this location. Otherwise a link to the location of the file in the BI server is shown in the Payroll Checklist Results UI.

If the BI_OUTPUT_SIZE action parameter is not present (the default) then a file of 10Mb or less will be stored in UCM and rendered in the Payroll Checklist Results UI. Files that are larger than 10Mb are rendered as a link to the file in the BI repository.

DERIVING PAYROLL FLOW INFORMATION

When the HCM Extract is processed by the Payroll Flow Engine it generates a set of records in PAY_REQUESTS and these can be interrogated to derive information required for subsequent processing.

If the HCM Extract was set up to generate a BI report then one of the PAY_REQUEST records will relate to the BI invocation. The Call_Id value is required to be able to run the BI ScheduleService to retrieve the file.

Using the BI ReportService the following SQL can be run to retrieve the PAY_REQUEST.CALL_ID:

```
select r.call_id
from pay_requests r,
     pay_flow_Instances fi
where r.flow_instance_id = fi.flow_instance_id
and instance_name = 'ExtractExample_Run1' -- the name of the Flow Instance
and r.call_type = 'BI_OP';
```

RETRIEVING BI REPORT GENERATED BY HCM EXTRACT

If the HCM Extract was set up to generate a BI report then the output will have been written to the BI Server from where it can be retrieved using the BI ScheduleService - http://bi-host/xmlpserver/services/v2/ScheduleService?wsdl

By generating a web proxy for the BI ScheduleService the following code can be used to retrieve the BI report:

```
public class ScheduleServiceClient {
@WebServiceRef
private static ScheduleService_Service scheduleService_Service;

static String username;
static String password;
static String outputFile;
static String called;

public static void main(String[] args) {
scheduleService_Service = new ScheduleService_Service();
ScheduleService scheduleService =
scheduleService_Service.getScheduleService();
// Add your code to call the desired methods.

username = System.getProperty("username", "c");
password = System.getProperty("password", "PWD");

outputFile = "OUTPUT.csv";
callId = "1234";  // CallId from PAY_REQUESTS query

try {
// Set the Job Filter to restrict to the file generated by the HCM Extract based on
CallId
JobFilterProperties jfp = new JobFilterProperties();
jfp.setJobName(callId); // CallId from PAY_REQUESTS
jfp.setJobNameOperator("Equals");

// Retrieve the JobInfo for the BI Report
JobInfosList jil = scheduleService.getAllScheduledReportHistory(jfp, 1, username,
password);

// Derive the JobId
JobInfo j = ((jil.getJobInfoList()).getItem()).get(0);
Long jobId = j.getJobId();

// Retrieve the JobOutput
```

```
JobOutputsList jol = scheduleService.getScheduledReportOutputInfo(jobId.toString(),
username, password);

// Retrieve the Job OutputId
JobOutput jo = (jol.getJobOutputList()).getItem().get(0);
Long outputId = jo.getOutputId();

// Retrieve the BI Report Document
byte[] out = scheduleService.getDocumentData(outputId.toString(), username, password);

// Write the byte stream to the output file
FileOutputStream o = new FileOutputStream(outputFile);
o.write(out);

} catch (Exception e) {
e.printStackTrace();
}
}
}
```

GENERATE BI REPORT FROM HCM EXTRACT

If an HCM Extract has not been set up with a BI Template then it is still possible to use a service to generate the required BI Report.

In order to do this the BI Template will need to be created based on the delivered data model:
/Shared Folders/Human Capital Management/Payroll/Data Models/globalReportsDataModel

Once the BI template is in place the following service call to the BI ReportService can be made:
- http://bi-host/xmlpserver/services/v2/ReportService?wsdl

This requires the **instanceName** parameter to be setup.
In the example below the output byte[] is simply written to System.out. In an orchestrated flow this would have to be streamed to wherever the next part of the process was to use it.

```
public class ReportServiceClient {
@WebServiceRef
private static ReportService_Service reportService_Service;

static String username;
static String password;
static String reportAbsolutePath;
static String instanceName;

public static void main(String[] args) {
reportService_Service = new ReportService_Service();
ReportService reportService = reportService_Service.getReportService();
// Add your code to call the desired methods.

username = System.getProperty("username", "USER");
password = System.getProperty("password", "PWD");
reportAbsolutePath =
System.getProperty("reportAbsolutePath",
      "/Custom/Human Capital Management/ExtractExample.xdo"); // The xdo in BI Catalog
instanceName = System.getProperty("instanceName", "ExtractExample_Run1");

try {
ReportRequest req = new ReportRequest();
```

```
ReportResponse res = new ReportResponse();

req.setReportAbsolutePath(reportAbsolutePath);
req.setSizeOfDataChunkDownload(-1);
req.setAttributeFormat("xml");  // The output format of the template
req.setAttributeLocale("en-US");
req.setAttributeTemplate("Simple");

ArrayOfParamNameValue arrayOfParamNameValue =
new ArrayOfParamNameValue();
ParamNameValues paramNameValues = new ParamNameValues();
ParamNameValue paramNameValue = new ParamNameValue();
ArrayOfString arrayOfString = new ArrayOfString();

paramNameValue.setName("instanceName");
arrayOfString.getItem().add(instanceName);
paramNameValue.setValues(arrayOfString);
arrayOfParamNameValue.getItem().add(paramNameValue);

paramNameValues.setListOfParamNameValues(arrayOfParamNameValue);

req.setParameterNameValues(paramNameValues);

res = reportService.runReport(req, username, password);

byte[] baReport = res.getReportBytes();

System.out.write(baReport);

} catch (Exception e) {
e.printStackTrace();
}

}
}
```

RETRIEVING EXTRACT FROM UCM

If HCM Extract is associated with a BI Template and the generated report is within the maximum file size defined by the BI_OUTPUT_SIZE action parameter, then the generated report will be written to UCM from where it can be downloaded.

The presence of the file in UCM can be derived using the following SQL:

```
select fi.instance_name,fd.datatype_code,nvl(fd.dm_document_id,fd.url) as
contentIdOrURL
from fnd_attached_documents fad,
fnd_documents_vl fd,
pay_requests pr,
pay_flow_instances fi
where fad.entity_name = 'FLOW_BI_OUTPUT'
and fad.pk1_value = pr.pay_request_id
and fad.document_id = fd.document_id
and fi.flow_instance_id = pr.flow_instance_id
and fi.instance_name like ' ExtractExample_Run1';
```

The DATATYPE_CODE indicates whether the file is available as a FILE or WEB_PAGE.

If the report is stored as a FILE in UCM then it can be downloaded using the standard UCM export utilities.

For example using the DownloadTool:

```
java -cp "oracle.ucm.fa_client_11.1.1.jar" oracle.ucm.client.DownloadTool --
url="https://{host}/cs/idcplg" --username="USER" --password="PWD" --
dDocName="UCMContentId" --outputFile="./output.csv"
```

V1 - April 2014