

A Web-based Survey Using Nouveau Alliance

By Larry Smith, Co-founder, Nouveau Systems

Contents

Introduction	2
Problem Statement	2
Previous Options	2
Our Solution	2
Implementation	3
Installation	10
Demonstration	11
Customization	12
Summary	12

Introduction

This white paper describes a web-based survey that is used to solicit attendees for a lunch event.

Problem Statement

The goal of this white paper is to demonstrate how to use Nouveau Alliance Workflow Service to build and deploy user surveys. A survey consists of a set of questions, usually given in-sequence, to be presented of a designated group of people. When the survey is complete, the results are stored in a database for subsequent analysis.

Previous Options

Currently, a person conducts a survey manually via the telephone or electronically by sending a list of questions via electronic mail. In either case, the results must be tabulated manually with the results re-communicated to the survey takers via the same medium.

Our Solution

This paper presents an example of using Nouveau Alliance to create and present a survey. You use the Nouveau Alliance Workflow Service to specify a workflow to define and conduct the survey. The workflow stores and retrieves the survey data using an embedded Derby database. The survey is then presented to the initiator and survey takers via the web using the Nouveau Alliance servlet interface.

Powerful Modeling

Nouveau Systems' FlowSpace provides a library of primitives that allow you to specify a survey in a simple, yet powerful way. If some customization is needed, you can add to or modify these primitives via Java or a myriad of supported scripting languages. When complete, these

customizations can be added to the library of built-in primitives to define new surveys.

Integrated RDMS Database

You can store and retrieve relational data from a full-featured, fully integrated RDBMS. Using the database access primitives of FlowSpace, you can seamlessly integrate the database access in your survey.

Web-based Servlet Interface

You can enter your application logic graphically by defining workflows with FlowSpace. You then can use a Workflow Servlet to create a web application. This enables domain experts to define and modify business logic without extensive knowledge of web programming.

Implementation

A survey of co-workers as to their lunch choices provides an example of how to create and deploy a survey. This Lunch Survey asks each person if they are interested in joining a lunch group and what their preferences are for food type, restaurant choice and when they are available. After a period of time specified by the surveyor, the survey results are sent to the participants.

The implementation of the Lunch Survey is best described by its Process, Database and Workflow models.

Process Model

The Lunch Survey is implemented by a single workflow with multiple execution threads.

When a Lunch Survey is started, independent threads are started to:

- Capture the survey parameters (who, what and when)
- Launch a monitor for the surveyor to view the surveys as they are completed
- Start a delayed “sweeper” to take care of surveys that are not completed by the specified time.
- Start a survey for each user to capture their responses.

The following figure shows these N+2 threads, where N equals the number of survey users.

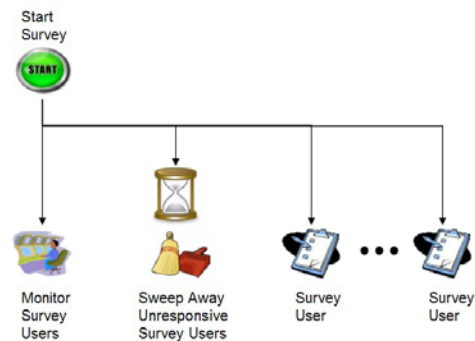


Figure 1 – Survey Threads

Each of the Survey User threads are suspended until each user resumes them to answer the survey questions. The “sweeper” and “monitor” threads are completed by time expiration and manual intervention, respectively.

Database Model

The persistence of the Lunch Survey is provided by the embedded use of Derby/DB RDBMS in Nouveau Alliance. There are two database tables used for this purpose.

Restaurants

This table specifies the choice of restaurants for the survey. Each restaurant has a type characteristic that allows for the selection by the type of food served by the restaurant.

restaurantID (INTEGER, GENERATED)

The unique identifier of each restaurant used as a primary key.

restaurantType (VARCHAR(50))

The type of food served by the restaurant (American, Chinese, Italian,...).

restaurantName (VARCHAR(50))

The name of the restaurant.

Users

This table is used by the Lunch Survey to track the survey takers. The contents of the table are determined by (1) the surveyor when he/she specifies a list of survey taker email addresses, (2) the survey taker when they make a restaurant selection and (3) by the “sweeper” thread when it tags the “participation” field with “DNR” (Did Not Respond) if a survey taker did not reply to the survey email.

userID (INTEGER, GENERATED)

The unique identifier of the table used as a primary key.

email (VARCHAR(80))

The email address of the survey taker. This address is used to send an email to each participant that has a link to launch the survey in the user’s browser.

participation (CHAR(3))

A code specifying the acceptance state of the survey taker (‘Yes’, ‘No’ and ‘DNR’. A value of ‘ ’ signifies that the survey is pending.

restaurantType (VARCHAR(50))

The survey taker’s food preference (Chinese, Italian, ...).

restaurantName(VARCHAR(50))

The survey taker’s restaurant selection.

timeFrame (VARCHAR(50))

A string representing the survey taker’s availability for lunch time (e.g., 11:30-12:30).

comments (VARCHAR(256))

Any optional comments to be appended to the survey taker’s response.

At the end of the survey, a report is generated from this table and mailed to the participants and the survey initiator.

Workflow Model

The Lunch Survey consists of two primary workflows representing the survey management and the actual survey sequence.

Lunch Survey Workflow

The Lunch Survey workflow is responsible for capturing and monitoring the survey, launching the survey, and cleaning up after the survey period has elapsed. The following is an “overview” diagram of the workflow.

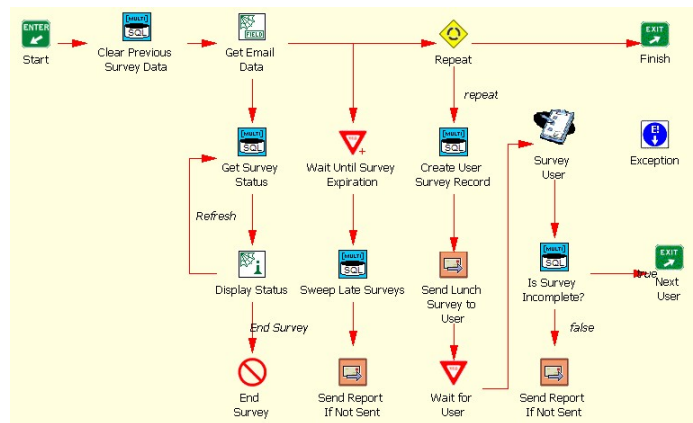


Figure 2 – Lunch Survey Workflow

For a better understanding, the workflow has been broken into two parts, with a listing of its constituent workflow nodes, activities and properties.

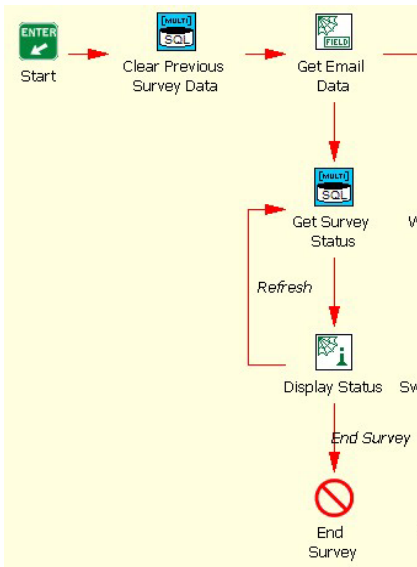


Figure 3 – Lunch Survey Workflow (capture/monitor)



Clear Previous Survey Data

This node clears the Users Table populated by a prior survey. A variable to track if the final report has been emailed to survey takers is also reset.

```
DBConnectActivityImpl
driver=org.apache.derby.jdbc.EmbeddedDriver
connectionProperty=INTERPRETER.DB_CONNECTI
ON
connectionURL=jdbc:derby:$WORKSPACE/defaultDB
;create=true
connectionUser=APP
connectionPassword=APP
```

```
DBExecuteActivityImpl
executeString=DELETE FROM Users
connectionProperty=$INTERPRETER.DB_CONNECTI
ON
returnStatusProperty=INTERPRETER.DB_RETURN_
STATUS
```

```
DBCloseActivityImpl
connectionProperty=$INTERPRETER.DB_CONNECTI
ON
```

```
SetPropertyActivityImpl
INSTANCE.reportSent=0
```

Note: From this point on the parameterization for the database connect and close activities is the same. They will be abbreviated by “...” to simplify the presentation.



Get Email Data

This node creates a web page from a template. The page defines a form which prompts for a list of survey taker email

addresses, the surveyor email address, the text for the survey invitation mail’s subject and body, and the time period which the survey takers must respond by.

The last Set Property Activity sets a variable with the “respond by” date to be used in the mail message sent to each survey taker.

```
SetPropertyActivityImpl
INTERPRETER.message=Please fill out the following form and
click the Send Survey button to send out the lunch invitations.
```

```
SetPropertyFromTextFileActivityImpl
URLspecifier=$INSTANCE.website/doc/examples/LunchSurvey
/templates/invitation_template.html
targetPropertyName=unformattedText
```

```
SetPropertyActivityImpl
formattedText="$unformattedText"
SERVLET.RESPONSE.print=$formattedText
close=$SERVLET.RESPONSE.close
```

```
WaitActivityImpl
waitTime=NULL
```

```
SetPropertyActivityImpl
INSTANCE.absTime=$(SYSTEM.currentTimeMillis +
$INSTANCE.timeout)
```

Note: The set property sequence of “SERVLET.RESPONSE.println and \$SERVLET.RESPONSE.close, followed by the indefinite Wait, causes the templated page to be displayed in the surveyor’s default web browser. From this point on, this sequence of activities will be abbreviated by the aggregate activity label “DISPLAY_WEB_PAGE_ACTIVITIES” and their custom property settings.



Get Survey Status

This node calls the sub-workflow “GetSurveyReport” to extract the current Survey Taker status. This workflow extracts data from the Users table, formats the data as a HTML table element, and returns the formatted text in an INTERPRETER property named “results”.

```
EnterActivityImpl
TOPOLOGY_STATE=BRANCH
```

```
SubWorkflowActivityImpl
workflowOID=c73d173e-f36f-46d6-95bb-abe74f3c8859.667"
```

Note: The EnterActivityImpl TOPOLOGY_STATE, in this case, must be biased to BRANCH in order to properly handle feedback loops. If not specified in this case, FORK inputs are assumed and the workflow pauses versus continuing with the loop.



Display Status

This node displays a web page with the results of the Survey Takers query. The web page’s sets the “answer” property based upon the selection of the

“Refresh” button or the “End Survey” button. This property is used to set the Exit Activity’s “label” property which determines the node that is traversed next.

```
SetPropertyActivityImpl
  INTERPRETER.message=<p>The following is a table
  of surveyed users:</p>\n$INTERPRETER.results
```

```
DISPLAY_WEB_PAGE_ACTIVITIES
  URLspecifier=$$INSTANCE.website/doc/examples/
  LunchSurvey/templates/info_template.html
```

```
ExitActivityImpl
  label=$INSTANCE.answer
```

End Survey

This node terminates any running thread of the Lunch Survey. It is intended to be used in the rare case where you want to interrupt the survey.

```
AbortActivityImpl
  reason=Survey Terminated by the originator.
```

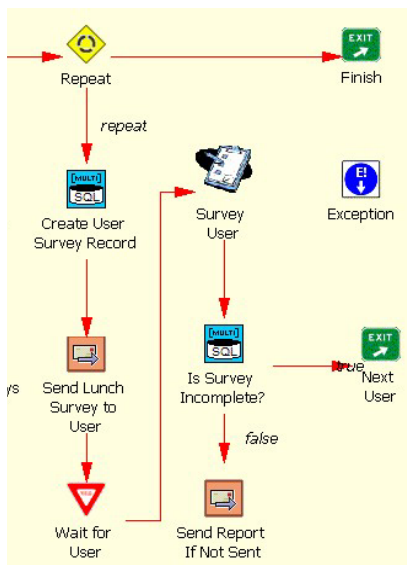


Figure 4 – Lunch Survey Workflow (launch)

Repeat

This node controls the invitation iteration sequence for the survey. The survey takers, defined by a comma-separated list of email addresses, are processed in parallel by this Repeat node.

```
RepeatActivityImpl
  repeatTopology=PARALLEL
  repeatCount=0
  repeatLabel=repeat
  repeatIndex=CONTEXT.NODE.repeat
  repeatWorkspaceIDs=NULL
  executionWorkspaceIDs=NULL
```

```
emailTarget=$INSTANCE.emailCSVList
```



Create User Survey Record

This node creates a skeletal entry in the Users table for a survey taker each time through the repeat loop. The user record is seeded by the survey taker’s email address. The default participation field is initialized to a blank character signifying that the survey is pending.

```
DBConnectActivityImpl
```

```
...
```

```
SetPropertyActivityImpl
  insertStatement="INSERT INTO Users
  (email,participation,restaurantType,restaurantName,timeframe,co
  mments) VALUES('$CONTEXT.emailTarget',' ',' ',' ')"
```

```
DBExecuteActivityImpl
  executeString=$insertStatement
```

```
...
```

```
DBExecuteQueryActivityImpl
  executeString="SELECT userID FROM Users WHERE
  email='$CONTEXT.emailTarget' ORDER BY userID DESC"
```

```
...
```

```
DBResultSetActivityImpl ...
```

```
DBCLOSEActivityImpl ...
```



Send Lunch Survey to User

This node sends a template-derived email message to each survey taker. The email message contains a form with a button the user can select to start their survey.

```
SetPropertyActivityImpl
  INTERPRETER.respondByMessage=Please answer this survey
  by $INSTANCE.absTime.toDate.toString
```

```
SetPropertyFromTextFileActivityImpl
  URLspecifier=$$INSTANCE.website/doc/examples/LunchSurvey/
  templates/email_invitation_template.html
  targetPropertyName=unformattedText
```

```
SetPropertyActivityImpl
  emailBody="$$unformattedText"
  SendEmailActivityImpl
  EmailRecipient=$CONTEXT.emailTarget
  EmailSender=$INSTANCE.emailSender
  EmailSubject=$INSTANCE.emailSubject
  EmailMessage=$emailBody
```



Wait for User

This node pauses the workflow thread for each survey taker. Each survey taker’s thread is resumed when the survey taker clicks the button in their email invitation.

```
WaitActivityImpl
  waitTime=NULL
```



Survey User

This node runs the “User Survey” sub-workflow. This workflow conducts the interaction of the survey. The definition of this workflow is presented below.

```
SubWorkflowActivityImpl
  workflowOID=c73d173e-f36f-46d6-95bb-
  abe74f3c8859.160"
```



Is Survey Incomplete

This node determines whether all survey takers have responded. It checks the Users table to see if any participation values are still in their initialized state (a blank value).

```
DBConnectActivityImpl
...
DBExecuteQueryActivityImpl
  executeString="SELECT userID FROM Users WHERE
  participation=" "
...
DBResultSetActivityImpl
...
  returnStatusProperty=TARGET
...
DBCLOSEActivityImpl
...
```



Send Report If Not Sent

After each survey taker responds to the survey, this node runs the “Send Report If Not Sent” sub-workflow to check if the final survey report has already been distributed and email it to all survey invitees if it hasn’t been sent. The final report is sent when all survey takers have responded or when the “respond by” time has expired.

```
SubWorkflowActivityImpl
  workflowOID=c73d173e-f36f-46d6-95bb-
  abe74f3c8859.3114"
```



Next User

This node provides a topological alternative to sending the final report. It does not have any functional content.



Exception

This node is used to capture any exceptions that might occur during the execution of the survey. If an exception arises, the surveyor is sent email with context information related to the exception.

```
SendEmailActivityImpl
  EmailRecipient=$INSTANCE.emailSender
  EmailSender=$INSTANCE.emailSender
  EmailSubject=Exception Thrown in running the Lunch Survey
  EmailMessage=Workflow: $WORKFLOW.name\nNode:
  $exceptionCursor.currentWorkflowNodeReference.label\nActivi
  ty#: $exceptionCursor.currentActivityIndex\nException:
  $exception.toString\nException
  StackTrace:\n$exception.stackTrace.toList.toString
```



Figure 5 – Lunch Survey Workflow (sweep)



Wait Until Survey Expiration

This node waits until the absolute date of expiration of the survey has expired. This node is preferred over a basic Wait node in that it provides the expected results if the workflow is resumed after a Nouveau Alliance server is restarted after a failure.

```
SetPropertyActivityImpl
  totalWait=$INSTANCE.timeout
  absWait=$((SYSTEM.currentTimeMillis + $totalWait)
```

```
WaitActivityImpl
  waitTime=$((totalWait * 10%)
```

```
CompareActivityImpl
  value1=0
  value2=$((max($absWait - SYSTEM.currentTimeMillis,0))
  comparison=1
  skipCount=-1
```




Sweep Late Surveys

After the survey time period has expired, this node finds and marks the participation fields of survey takers who have yet to respond.

DBConnectActivityImpl

...

DBExecuteActivityImpl

```
executeString="UPDATE Users SET
participation=DNR' WHERE participation=" "
```

...

DBCcloseActivityImpl

...



Send Report If Not Sent

This node is a duplicate of the node of the same name in previously described in the “launch” thread.

User Survey Sub-Workflow

The User Survey sub-workflow represents the survey presented to the customer. The following is an “overview” diagram of the workflow.

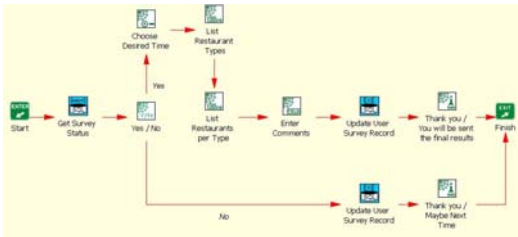


Figure 6 – User Survey Workflow

For a better understanding, the workflow has been broken into two parts, with a listing of its constituent workflow nodes, activities and properties.

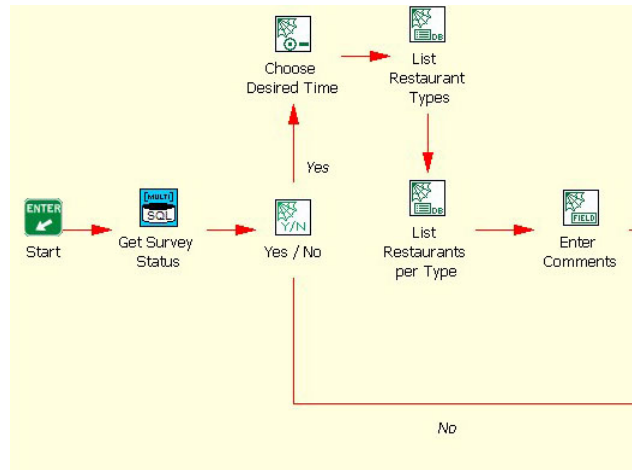


Figure 7 – User Survey Workflow (front)



Get Survey Status

This node is a duplicate of the node of the same name in previously described in the “capture / monitor” thread.



Yes / No

This node asks the user via a web page if they would like to participate in the survey. If they select the “yes” button (setting the INSTANCE “answer” property”), the next survey question appears in the browser. If the “No” button is selected, then a “Thank you” message is displayed and their Users record is updated with their negative participation intent.

SetPropertyActivityImpl

```
INTERPRETER.message=<p>I am conducting a survey for those
interesting in going out to lunch today.</p><p>The following is
the current status of this
survey:</p>${INTERPRETER.results}<p>Are you interested in
joining us?</p>
```

DISPLAY_WEB_PAGE_ACTIVITIES

```
URLspecifier=${INSTANCE.website/doc/examples/LunchSurvey/
templates/yesno_template.html
```

ExitActivityImpl

```
label=${INSTANCE.answer
```



Choose Desired Time

This node presents a web page with a set of radio buttons to asks the user when they prefer to have lunch. The “getOptions” sub-workflow is used to create the HTML radio buttons from the INTERPRETER displayList and ValueList properties.


```
SetPropertyActivityImpl
  INTERPRETER.displayList="11:30-12:30','12:00-1:00','12:30-1:30"
  INTERPRETER.valueList=$INTERPRETER.displayList
  INTERPRETER.message=Please select a preferred time frame.
```

```
SetPropertyActivityImpl
  INTERPRETER.resultsRow='<input type="radio" name="choice" value="$CONTEXT.value">$CONTEXT.display<br>'
```

```
SubWorkflowActivityImpl
  workflowOID=c73d173e-f36f-46d6-95bb-abe74f3c8859.177
```

```
SetPropertyActivityImpl
  INTERPRETER.options=$INTERPRETER.results
```

```
DISPLAY_WEB_PAGE_ACTIVITIES
  URLspecifier=$$INSTANCE.website/doc/examples/LunchSurvey/templates/radiobutton_template.html
```

```
SetPropertyActivityImpl
  INTERPRETER.timeFrame=$INSTANCE.choice
```



List Restaurant Types

This node presents a web page with a list of restaurant food types (e.g., Chinese, Italian ...) to ask the user for their preference. The “GetOptionsFromDB” sub-workflow is called to get a list of restaurant types from the Restaurants database table and converts them to HTML select options.

```
SetPropertyActivityImpl
  INTERPRETER.selectStatement=SELECT DISTINCT restaurantType FROM Restaurants ORDER BY restaurantType
  INTERPRETER.selectProps=INTERPRETER.restaurantType
  INTERPRETER.resultsRow='<option value="$INTERPRETER.restaurantType">$INTERPRETER.restaurantType</option>'
  INTERPRETER.results=""
  INTERPRETER.message=Select the type of restaurant that you want.
```

```
SubWorkflowActivityImpl
  workflowOID=c73d173e-f36f-46d6-95bb-abe74f3c8859.175
```

```
SetPropertyActivityImpl
  INTERPRETER.options=$INTERPRETER.results
```

```
DISPLAY_WEB_PAGE_ACTIVITIES
  URLspecifier=$$INSTANCE.website/doc/examples/LunchSurvey/templates/list_template.html
```

```
SetPropertyActivityImpl
  INTERPRETER.restaurantType=$INSTANCE.option
```



List Restaurants per Type

This node presents a web page with a list of restaurants for the restaurant type selected in the previous node. The “GetOptionsFromDB” sub-workflow is called to get a list of restaurants for the type from the Restaurants database table and converts them to HTML select options. Any

single quotes in the restaurant name are properly escaped for the subsequent SQL INSERT statement.

```
SetPropertyActivityImpl
  INTERPRETER.selectStatement="SELECT restaurantName FROM Restaurants where restaurantType=$INTERPRETER.restaurantType"
  INTERPRETER.selectProps=INTERPRETER.display
  INTERPRETER.resultsRow='<option value="$INTERPRETER.display">$INTERPRETER.display</option>'
  INTERPRETER.results=""
  INTERPRETER.message=Please select your desired restaurant.
```

```
SubWorkflowActivityImpl
  workflowOID=c73d173e-f36f-46d6-95bb-abe74f3c8859.175
```

```
SetPropertyActivityImpl
  INTERPRETER.options=$INTERPRETER.results
```

```
DISPLAY_WEB_PAGE_ACTIVITIES
  URLspecifier=$$INSTANCE.website/doc/examples/LunchSurvey/templates/list2_template.html
```

```
SetPropertyActivityImpl
  INTERPRETER.restaurantName=$INSTANCE.option
```

```
ReplaceStringActivityImpl
  REPLACE-STRING='\ '
  WITH-STRING='\'
  USE-REGEXP=false
  INTERPRETER.restaurantName=$INTERPRETER.restaurantName
```



Enter Comments

This node presents a web page with the HTML textures to capture comments from the user. Once again, any single quotes in the comments are escaped for the subsequent SQL INSERT statement.

```
SetPropertyActivityImpl
  INTERPRETER.message=Enter any comments you would like to accompany your response (optional):
```

```
DISPLAY_WEB_PAGE_ACTIVITIES
  URLspecifier=$$INSTANCE.website/doc/examples/LunchSurvey/templates/textfield_template.html
```

```
ReplaceStringActivityImpl
  REPLACE-STRING='\ '
  WITH-STRING='\'
  USE-REGEXP=false
  INTERPRETER.comments=$INSTANCE.text
```

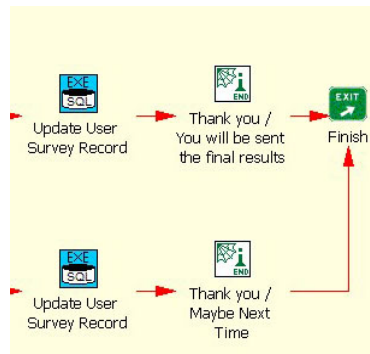


Figure 8 – User Survey Workflow (back)



Update User Survey Record (top)

This node updates the survey taker's Users database record to reflect the survey responses.

DBConnectActivityImpl

...

DBExecuteActivityImpl

```
executeString="UPDATE Users SET participation='Yes',
restaurantType='$INTERPRETER.restaurantType',
restaurantName='$INTERPRETER.restaurantName',
timeFrame='$INTERPRETER.timeFrame',
comments='$INTERPRETER.comments'
WHERE userID=$INTERPRETER.userID"
```

...

DBCcloseActivityImpl

...



Update User Survey Record (bottom)

This node updates the survey taker's Users database record to reflect a "No" participation choice

DBConnectActivityImpl

...

DBExecuteActivityImpl

```
executeString="UPDATE Users set participation='No'
WHERE userID=$INTERPRETER.userID"
```

...

DBCcloseActivityImpl

...



Thank you / You will be sent the final results

This node presents a web page (without any wait for response) to thank the survey taker for participating.

SetPropertyActivityImpl

```
INTERPRETER.message=Thank you for completing
the survey. You will be mailed the final results
shortly.
```

DISPLAY_WEB_PAGE_ACTIVITIES (no wait)

```
URLspecifier=$INSTANCE.website/doc/examples/Lu
nchSurvey/info_end_template.html
```



Thank you / Maybe Next Time

This node presents a web page (without any wait for response) to thank the survey taker for NOT participating.

SetPropertyActivityImpl

```
INTERPRETER.message=Thank you for your
response. Perhaps you will join us for lunch at
another time.
```

DISPLAY_WEB_PAGE_ACTIVITIES (no wait)
URLspecifier=\$INSTANCE.website/doc/examples/LunchSurvey/info_end_template.html

Installation

The Nouveau Alliance Lunch Survey example can be located in three locations:

- A pre-installed workspace called "examples".
- A workflow **template** called "examples".
- A ZIP archive file named LunchSurvey.zip that is available from the Nouveau Systems website.

The specific installation instructions depend on the location.

Pre-installed Workspace

If the "examples" workspace has already been installed, with workflow definitions (LunchSurvey, UserSurvey and LoadDatabase), no additional installation is required. If these workflow definitions do not exist, please follow the ZIP Archive instructions presented below.

Workspace Template

To install the "examples" workspace from a template, start FlowSpace and select the New Workspace option from the File menu and select the "examples" template. Once the workspace is created, confirm that the Lunch Survey workflows are present (see list above). If not, then follow the ZIP Archive instructions presented below.

ZIP Archive

The Lunch Survey ZIP archive file can be downloaded from here:

<http://download.nouveausystems.com/whitepapers/LunchSurvey.zip>

To install this zip file:

1. Change directory to your Alliance root directory

2. Unzip the ZIP archive file in this directory. The output of the ZIP file will be placed in the following directories:

doc/examples/ LunchSurvey
etc/servlets

3. Start FlowSpace and open or create a workspace called “examples”.
4. Into this workspace, import the workflow definition archive files in this Lunch Survey xml directory:

LunchSurvey.zip
LoadDatabase.zip

These XML-archive files specify workflow definitions that contain all of the Lunch Survey operations.

To load these definitions into the “examples” workspace, select the “Import New Workflow” option from the FlowSpace section page number pop-up menu in the lower-right corner of the display. You can select more than one xml archive file to import at a time by using the control or shift keys while making the selection.

Required System Software

The Lunch Survey uses the Send Email Activity which relies on the Nouveau Alliance Server being configured to access a local SMTP server that can accept outgoing email messages. This requires having Nouveau Alliance Server’s ‘mail.smtp.host’ and optionally the ‘mail.smtp.port’ settings correctly configured in the server’s “etc/config.ini” file. The Nouveau Alliance Installer attempts to seed this setting during installation.

Required License and Version

A Professional or Department license for FlowSpace version 3.2 and later is required to use the Lunch Survey. If you're currently using Personal Edition, try FlowSpace Professional Edition *free* for 30 days now! Just point your web browser at your Nouveau Alliance server's home page, and click "Get Your Upgrade Evaluation!" to get started.

With the appropriate license, and a running Nouveau Alliance server, you can demonstrate the use of the Lunch Survey.

Demonstration

To demonstrate the Lunch Survey:

1. Make sure that the Nouveau Alliance Server is running.
The workflows and the servlets of the framework require this server process.
2. Display the Lunch Survey’s main page:
<http://yourhost:8081/doc/examples/LunchSurvey/index.html>
This page contains buttons that control the Lunch Survey’s operation (loading data and starting the survey).
3. Load in the demonstration database schema by pressing the Database Initialize button.
This step loads in test survey restaurants.
4. Start the Lunch Survey by pressing the Survey Start button.
This step will display a survey customization page.
5. Complete the form on this page by specifying a comma-separated list of survey taker email addresses in the “To” field. In the “From” field, specify your email address.
You can optionally modify the “Subject” and “Message” fields to customize your lunch invitation.

6. Select a “Response by” time to specify the survey collection time period.
7. To send out the invitations to the survey takers, press the “Send Survey” button.
Electronic mail will be sent to the survey takers and a status report will be displayed.

A “Refresh” button is supplied on this page to allow you to periodically check on the status of the survey and an “End Survey” button is provided to terminate the survey.

Customization

You can replace the Lunch Survey described in this white paper by specifying what data that you want to collect (the Database Schema) and how you want to obtain the data (the User Survey Workflow).

Database Schema

You can populate or extend the database tables which store the list of available restaurants and survey takers (users).

Restaurant Data

You will need to populate the “Restaurant” table with the restaurants in your area. The demonstration contains the basic information about a restaurant (name and type of food). You could Alter the table to include any additional information that would be useful to collect.

User Data

The survey takers in the demonstration data are characterized by an “email” field in the “Users” table, as well as their responses to

the survey questions. To build your own survey, you can populate this table with your user characteristics altering the table to maintain more information about the survey taker (additional contact information or additional survey responses).

Workflow Definitions

Once you establish the list of restaurants to select from and what information is to be obtained from each survey taker, you can then specify how the data is to be gathered. This is done by replacing the User Survey workflow.

User Survey Workflow

You will need to replace the User Survey workflow with your own question sequencing and dependencies. The survey database schema must be synchronized with flow of the workflow.

You can complete the customization by deciding what properties should be specified in the servlet.config file, or specified “in-line” in the workflow node activities. For example, in the lunch survey, for ease of customization, all of the survey questions are defined as INSTANCE properties.

Summary

This white paper describes the construction of a survey taking application using Nouveau Alliance. Some of the technologies used include:

- Integrated Derby/DB RDMS
- Workflow Enabled Servlets
- Flexible Workflow Modeling

The paper describes how to model a survey taking tool from a process, database and workflow perspective. It also includes a demonstration lunch survey to bring the framework to life and provide insight on how to customize this survey for your own use.