NICE Designer
Self-Paced Automation Essentials

NICE Advanced Process Automation

Exercise Book

For Automation Developers

Education Services

Table of Contents

Physical Objects Layer: Screen Elements

Sandbox Exercise

1. Screen Elements may be captured in order to capture an event, collect information or perform a function. For each of the following scenarios:

   i. Capture the relevant screen elements

   ii. Give each captured element a meaningful name

   iii. Make relevant hidden captures visible, and rename them appropriately

   iv. Verify connectivity using the Locate function

   - In the Calc.exe application:
     - Capture **3 digits**
   - In the CRM Application:
     - When a **New Contact is added**:
       - Check to see if the contact has an **associated account**
     - An **Account is closed**
     - When a **new Account is added**:
       - Add the account's **full billing address** to a list.
   - In Internet Explorer:
     - Enter a **word to search** at www.microsoft.com, and click the **search button**.

- After capturing the search box, change the Screen Element Type from HTML Element to HTML TextBox

NOTE

The Screen Element screenshot in the Designer is only an aid for your reference. It has no effect on the Designer's ability to identify the element. For this, use the 'Locate' function.

- Save and Test your Project using the Monitor Application
    - Verify any textbox value changes in real time
- We wish to deal with differences in Identification.

Tune the Identification of the NICE Website captures such that they can be located after a search has returned valid results.

Business Entities Layer

Sandbox Exercise

1. Create the following Business Entity Types and Properties:
    - Address (Base Type: None)
        - Street (Text)
        - City (Text)
        - State (Text)
        - Zip (Text)
    - Bronze Customer (Base Type: None)
        - Full Name (Text)
        - DOB (DateTime)
        - Home Address (Address)
        - Senior (Boolean)
        - Points (Number)
    - Silver Customer (Base Type: Bronze Customer)
        - Date Upgraded (DateTime)
2. Create an instance of the Bronze and Silver Customers, and set initial values for all their properties.
3. Create lists of the Bronze and Silver Customers
4. Create the following Business Entity Types and Properties, and make an instance of each:
    - CRM Account (Base Type: None)
        - Account Name (Text)
        - Account Source (Text)
        - Active (Boolean)
        - New Status (Boolean)
        - Billing Address (Address)
    - CRM Contact (Base Type: None)
        - Title (Text)
        - First Name (Text)
        - Last Name (Text)
        - Account (Text)
        - Phone (Text)
        - Email (Text)
        - Home Address (Address)
    - Timer (Library type – make Instance only)
5. Capture the screen elements (from the **Accounts** tab) that relate to the CRM Account properties, and set them to be Data Sources of the relevant properties.

6. Capture the screen elements (from the **Contacts** tab) that relate to the CRM Contact properties, and set them to be Data Targets of the relevant properties.

7. Set the appropriate life cycle for all BEs created in steps 1 and 4.

8. Test your BEs using the Monitor Application:

   - Change the Account fields in the Training CRM and see how the appropriate values change in the Monitor Application.
   - Change the CRM Account properties' values in the Monitor Application. Do the values in the fields of the Accounts tab (in the CRM Application) change? Why?
   - Change the CRM Contact properties' values in the Monitor Application and see how the appropriate fields change in the Contacts tab (in the CRM Application).
   - Change the Contact fields in the Training CRM. Do the CRM Contact properties' values in the Monitor Application change? Why?

NOTE

Although "Data Target" is a means for a Screen Element to hold a value stored in a Business Entity, it is strongly recommended not to use the "Data Target" and instead assign values explicitly to a Screen Element upon demand as part of the process' flow.

Business Entities Layer: Functions and Events

Business Request #1

When a new Account is added in the CRM application,
add the account's full address to a list

Physical Objects Layer – Screen Elements

1. Create a new project called "Training PO Layer"

   Which Screen Element(s) do you need to capture in order to know when a new account has been added?

   _____

   _____

   Which Screen Element(s) do you need capture in order to obtain the relevant account information?

   _____

   _____

2. For each of the Screen Elements you have listed:
   a. Capture the Screen Element
   b. Name it appropriately
   c. Create a logically organized hierarchy
   d. Ensure that you can locate it

3. Test your captures using the Monitor application.

4. Save your project.

Business Entities Layer

1. Create a new project called "Training BE Layer" and reference the "Training PO Layer" project. Verify that you can see the Screen Elements you captured in the Physical Objects Layer.

2. Create an Instance and call it "List of Addresses" of Type **List of Text**.

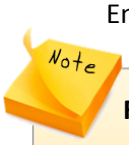
3. Create an "Address" Type with *Street*, *City*, *State* and *Zip* Properties.

4. Create a Function under the Address Type called "Add Full Address to List" which concatenates the address values into a single line and adds it to the "List of Addresses".

5. Create an "Account" Type that includes:
   - "Billing Address" Property (of type Address)
   - "New Account Created " Event

6. Create an Instance of the Account Type, call it "New Account", and define the logical data flow and life cycles for each Property and Event.

7. Save and Test your solution with the Monitor Application. At this stage, just verify that the text entered in the CRM Application can be seen in the appropriate Business Entities.



REMINDER

In the Business Entities Layer, we are only *Creating* and *Defining* Functions and Events.
That is, we are defining what we want to happen and when we want things to happen.
BUT we are not yet associating specific Functions with specific Events.
That happens in the Business Logic Layer, which we will cover next.

9. BONUS: Create another Function under the Address Type called "Add Full Address to List (Conditional)" which concatenates the address values into a single line and adds it to the "List of Addresses", but only if all of the address fields are populated.

Business Logic Layer

Business Request #1

When a new Account is added in the CRM application,
add the account's full address to a list

Business Logic Layer – Event Handlers

1. Create a new project called "Training BL Layer" and reference the "Training BE Layer" project. Verify that you can see the BEs you created in the BE Layer.
2. Create an Event Handler that runs the "Add Full Address to List" function when the "New" event you defined is raised.
3. Save and Test your solution by entering data into the relevant fields in the CRM application and watching the list grow in the Monitor application.
4. BONUS: Add/Modify instructions in your Event Handler so that it does exactly what it is currently doing, but only if all of the Billing Address fields are populated.

Business Request #2

When the agent is updating the Personal Information in the Contacts tab update the customer premium according to the table below

Gender	Age	Previous claims	Premium
Male	20-30	Yes	$200.00
Male	20-30	No	$150.00
Male	31-40	Yes	$130.00
Male	31-40	No	$100.00
Female	20-30	Yes	$170.00
Female	20-30	No	$140.00
Female	31-40	Yes	$125.00
Female	31-40	No	$90.00

- **Disable the Premium field as part of the solution**

Physical Objects Layer – Screen Elements

1. Capture the relevant Screen Element for this business request.

2. As before:
   a. Name it appropriately
   b. Create a logically organized hierarchy
   c. Ensure that you can locate it
   d. Test your capture using the Monitor application

Business Entities Layer

1. In the **Training BE Layer** project, create a "Customer" Type to hold the different customer details:
   - A Number for Age
   - A Text for Gender
   - A Boolean for Previous Claims
   - A Decimal for Premium

2. Create an instance of the type you have just created.

Business Logic Layer - Rules

1. In the **Training BL Layer** project, create a rules tree to calculate the customer premium according to the table above. The rules should be nested, and the lowest rules (branches) should assign a different value to the Premium variable.

2. Add an Event Handler and a function to assign the calculated value of Premium to the relevant SE

3. Save and Test your solution by seeing if changing the values (Gender, Age and Previous Claims) in the turns on the appropriate Rules at the appropriate times and changes the Premium value accordingly.

Physical Objects Layer: Databases

Business Request

- **Load the following columns from the "X Sell" table into your Business Entities.**
  - **First Name**
  - **Last Name**
  - **Phone**
  - **Email**
- **When a New Contact is added in the Training CRM Application:**
  - **Add the contact's First Name, Last Name, Phone and Email to the "X Sell" table**
  - **Load the same columns from the Updated Table into your Business Entity**

Presentation Layer

Business Request

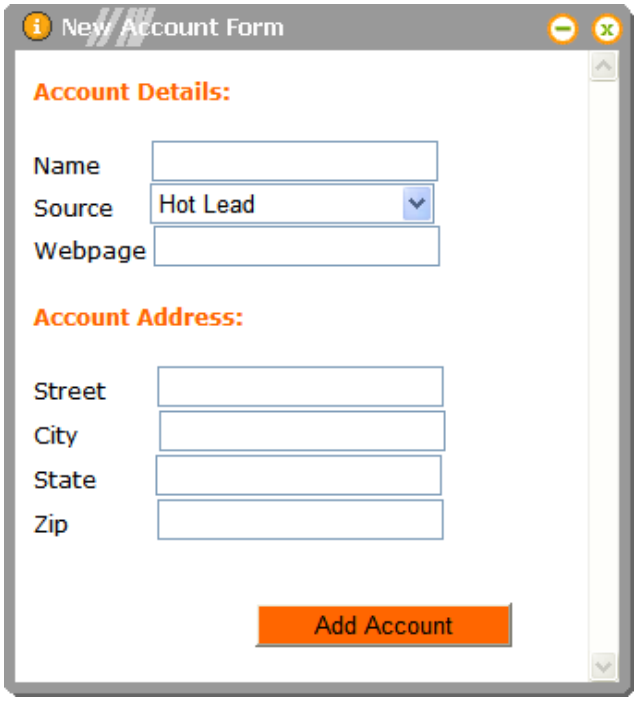
When an Account is closed in the CRM application:

a. **Have a message appear on the agent's screen to allow the agent to select the reason for closing the account: Service / Product / Other**
b. **Have a textbox appear when "Other" is selected**
c. **Have a textbox for the user to enter the account webpage**
d. **Add an "OK" button which, when clicked, will close the callout and launch the account's webpage in IE**

Automation

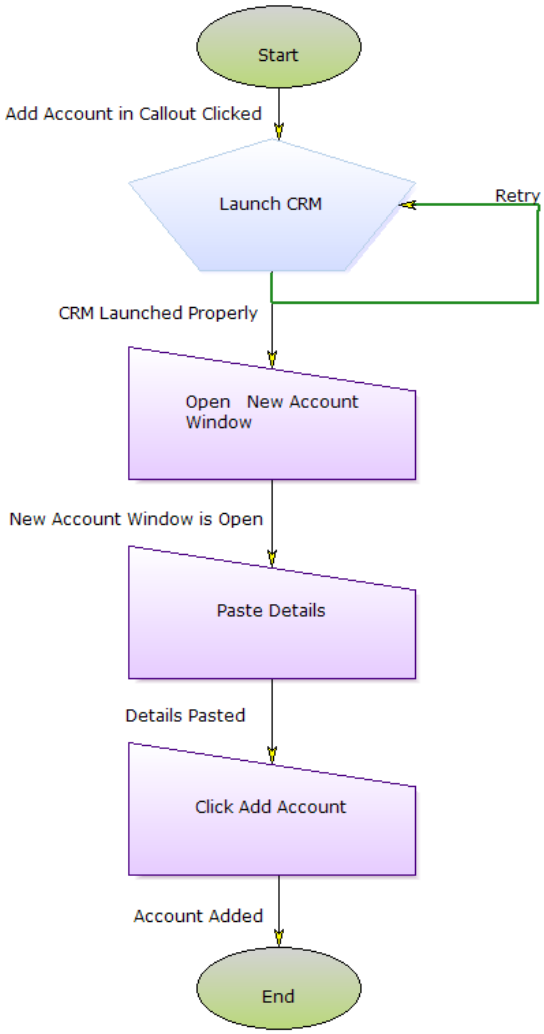
Business Request:

Create an Automated Workflow for adding a new account in the Training CRM Application. The user will be presented a form with the appropriate fields corresponding with the New Account window (an example can be seen below).



Once the "Add Account" button is clicked in the form, the Automation Workflow will launch the Training CRM, open the New Account window, paste the details and click the "Add" button.

An **example** workflow diagram can be seen below:



1. Make your Workflow foolproof:
   - Add timeout to necessary Steps.
   - Add transition loops if necessary
   - Add notification Callouts to the workflow.
   - Clear the form's fields when the Workflow ends.

2. Test your Workflow.
   - Save your Project.
   - Run the RT Client, open the Monitor Tool and test the flow.

Robotic Automation

Business Request: Automate Activity Entry

Invoker:

- **When a New Account is added:**
  - **Send an invocation which includes the Account Name and Web Page of the New Account**

Robotic Workflow:

- **Click on the "Activities" Tab**
- **Click on the "New" Button**
- **Paste the following text into the Notes section:**
  "New Account Created: [Account Name]"
- **Open the Webpage specified when the Account was added, in a new browser window (have the webpage open for 5 seconds before completing the workflow)**

Decision Support: Data Collection

Business Request #1 – Extension of Business Logic Exercise

The customer would like to collect data on the Billing Addresses of every New Account created.
Have the data collection begin when the New Account window appears, and have the data collection complete when the account has been created.

Client-to-Client

Sandbox Exercise – Simple Message Service

This exercise will utilize the Designer machine and the Client machine as 2 different Clients communicating with each other. The Designer machine will be the sender and the Client machine is the receiver.

Create a *C2C Receiver* Project:

1. Create a Text Business Entity:
   - "Message Received"
2. Create a Client-to-Client Remote Channel
   - Name it "My Channel"
   - Enter your Designer machine name as the Remote Host
   - Enter port numbers for the Remote and Local Machines
   - Verify that the Local Port defined matches the Remote Port defined on the sender machine. Available ports are 5000 a 5001.
3. Create a Client-to-Client Assignment
   - Name it "My Assignment"
   - Create a unique Key (this will be used by the sender)
   - Assign this key to the "Message Received" BE
4. Publish your solution and assign it to the NICE_A team

Create a *C2C Sender* Project:

1. Create a Text Business Entity:
   - "Message to Send"
2. Create a Client-to-Client Remote Channel
   - Name it "My Channel"
   - Enter your Client machine name as the Remote Host
   - Enter port numbers for the Remote and Local Machines
   - Verify that the Local Port defined matches the Remote Port defined on the receiver machine. Available ports are 5000 a 5001.
3. Create an Event Handler
   - Event: When the value of "Message to Send" is modified
   - Action: Send "Message to Send" across the "My Channel" channel. Use the Key defined in Assignment you have created on the Client machine.
4. Test your solution
   - Run the Client from both the Designer and Client machines
   - Open the Monitors
   - Enter a message in your "Message to Send" Business Entity and see it appear in the "Message Received" BE

Bonus: Final RPA Exercise

Create a full Robotic Automation solution.

Similar to the business request in the *Fundamentals Review and Presentation Layer* exercise, the organization requires a solution involving creating a customer's new account and storing whether the address is according to shipping policy. The process includes entering the new account data, Launching Google Maps, verifying the distance and storing the data.

Invoker:

- **New Account Callout:**
  - **New Account data fields (CRM)**
  - **Distance to the customer's address**
  - **Button for launching Google Maps (this button will be enabled only once the City is entered)**
  - **Button for sending the request to the queue**

Robotic Automation Workflow:

- **Create New Account in the CRM application with the values received from the Invoker**
- **Evaluate the distance. For values under 100 KM, set the Shippable property to "Shippable", and "Non Shippable" otherwise**
- **insert the information to the Shippable table in the DB**
  - **Account Name**
  - **City**
  - **Shippable**
  - **Missing Data Y/N (for any field that is blank)**
- **If the distance has not been sent from the Invoker (its value is 0), write the following text to the RT Client log file: <Account Name> is missing the distance**

Extra Exercises

Business Request #3

There has been a problem with agents pre-dating the activities they enter.
From now on, if the Date of the current Activity is earlier than the Current Date,
the Activity Date must be immediately reset to the Current Date.
This must also be part of the first solution.

Physical Objects Layer – Screen Elements

3. Capture the relevant Screen Element for this business request.

4. As before:
    e. Name it appropriately
    f. Create a logically organized hierarchy
    g. Ensure that you can locate it
    h. Test your capture using the Monitor application

Business Entities Layer

1. Here, we will be using BEs to absorb the value in the captured SE, as well as to send a value back to the same SE.

Can the same BE be used to perform both functions?

_____

_____

2. Create the BE Type and Property(s) required for this, and define their logical data flow(s).

3. Create another BE called "Today" that is initialized with the **Current Date** (a Library Object). This will be used at the Business Logic Layer to compare with the Activity date.

4. Test these BEs using the Monitor application

Business Logic Layer – Rules

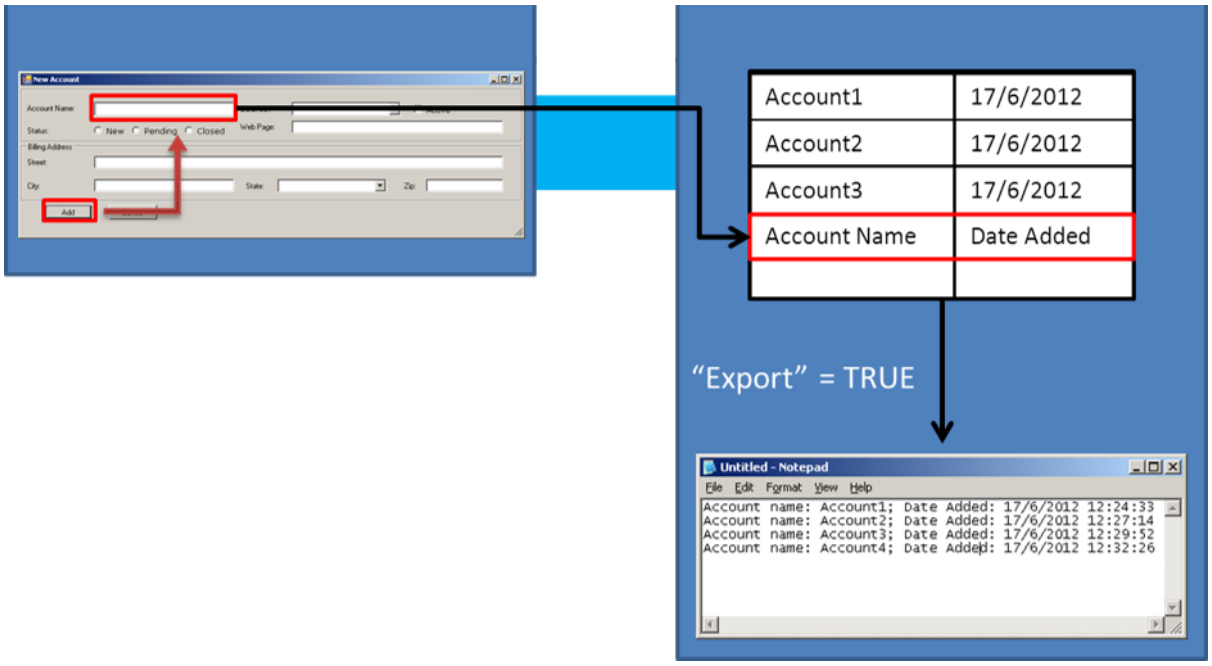
5. Create a Rule that resets the Activity Date to the Current Date if the agent tries to set the Activity Date earlier than the Current Date.

6. Save and Test your solution by entering data into the relevant field in the CRM application and watching the Activity Date reset when appropriate.

Notepad Presentation

When an Account is added, the Account Name and the Date it was added must be added to a list.

You then must be able to export the entire list to an open Notepad window when a trigger is executed via the monitor.



Automate Activity Entry

- **When a New Account is added, automate the following:**
  - **Click on the "Activities" Tab**
  - **Click on the "New" Button**
  - **Paste the following text into the Notes section:**
    "New Account Created: [Account Name]"

Awarding Prizes

- Create a complete solution in order to accommodate the following:
  - For every 5 new contacts added, randomly select one to be the winner
  - Once the winner has been selected, create a new E-mail as follows:
    - To: the winner's E-mail address
    - subject: Congratulation
    - Content: Dear <Contact's Name>
      You have won a prize

Please note: Even though Outlook might not be configured in your system this exercise can still be tested

Add Button Manipulation

- Create the proper Physical Objects, Business Entities and Business Logic in order to accommodate the following:
  - When the New Contact window is opened, disable the Add button immediately
  - The Add button should be enabled only when all fields in the window are populated with data.

Is Senior Function

- Create a function that receives a person's Date of Birth (a complete date, not only the year) and determines whether the person is senior (above 65) or not.
  - Use the Activity Date on the activities tab from the CRM training App as your input
  - Can this be done without creating a Business Logic Layer?

On-Screen Counter/Timer

- Create a callout that displays an incrementing counter.
- From within the callout, you should also be able to:
  - Change the increment step (it doesn't necessarily have to increment by one)
  - Change the increment speed
  - Reset the counter to Zero
  - Reset the settings to their Default values

Google Search

- **Create a Callout with an input text field and an OK button.**
- **When the agent clicks OK, the process will search the text in www.google.com.**
- **This Callout should be available to the agents from their Quick Menu.**

Database Access Workaround

- **The customer requires a solution involving access to the "Products" table in the Database, but cannot permit you to open the Database directly. You can only access the Database through the Designer.**
  Create a callout, accessible from the Quick menu, which:
  - **Always displays the up-to-date contents of the "Products" table**
  - **Allows you to INSERT, UPDATE and DELETE records from the table at will**

Offer Products Process

- **When a new contact is added in the Training CRM application, a popup window should appear, prompting the agent to enter the contact's age.**
- **A contact is defined as "Senior" if they are above 60 years old. Below is a table outlining the Regular and Seniors prices for the two products offered by the company.**

Product Name	Regular Price	Seniors Price
Home	$200	$180 (10% Discount)
Auto	$500	$400 (20% Discount)

- **Once the agent has entered the contact's age, a callout should appear, containing the names of the two products listed above, and their appropriate prices, which will be offered to the contact.**
- **In the same callout, the agent should be able to specify which product the contact would like to buy, or indicate that the contact is "Not Interested".**
- **Once a decision has been made, using the Concatenate function, add the following text to a list:**
  Full Name: [First Name] [Last Name]; Product: [Product Chosen]
- **Have this list accessible though the Quick Reference callout**

Client-to-Client Callouts

For this exercise:

- ***You must work in pairs. Each member of the pair will create their own project***
- ***All layers may be in the same project***

- **Create a Remote Client-to-Client channel with your partner so that both of you can send and receive data to/from each other.**
- **Create a Callout with:**
  - **Two Input Text fields. Each of these fields will contain a message to be sent to different Business Entities on your partner's machine asynchronously**
  - **Send buttons for each Text field**
  - **An area to display the two messages received from your partner**

Contact us: training@nice.com
Visit us: Technical training on ExtraNICE

NICE®