Advanced ATG Personalization

Presenter's Name
Presenter's Title

Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions.
The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Oracle Training Materials – Usage Agreement

Use of this Site ("Site") or Materials constitutes agreement with the following terms and conditions:

1. Oracle Corporation ("Oracle") is pleased to allow its business partner ("Partner") to download and copy the information, documents, and the online training courses (collectively, "Materials") found on this Site. The use of the Materials is restricted to the non-commercial, internal training of the Partner's employees only. The Materials may not be used for training, promotion, or sales to customers or other partners or third parties.

2. All the Materials are trademarks of Oracle and are proprietary information of Oracle. Partner or other third party at no time has any right to resell, redistribute or create derivative works from the Materials.

3. Oracle disclaims any warranties or representations as to the accuracy or completeness of any Materials. Materials are provided "as is" without warranty of any kind, either express or implied, including without limitation warranties of merchantability, fitness for a particular purpose, and non-infringement.

4. Under no circumstances shall Oracle or the Oracle Authorized Delivery Partner be liable for any loss, damage, liability or expense incurred or suffered which is claimed to have resulted from use of this Site of Materials. As a condition of use of the Materials, Partner agrees to indemnify Oracle from and against any and all actions, claims, losses, damages, liabilities and expenses (including reasonable attorneys' fees) arising out of Partner's use of the Materials.

5. Reference materials including but not limited to those identified in the Boot Camp manifest can not be redistributed in any format without Oracle written consent.

Agenda

- Overview of Custom Scenario Events, Actions, and Conditions
- Overview of Collection Filtering
- Using Profile Markers
- Overview of Business Processes
- Workflows Overview

Learning Objectives

At the end of this lesson you should be able to:

- Understand the use of custom scenario events, actions and conditions
- Use collection filters to filter items in a collection
- Use profile markers to mark specific user profiles
- Perform actions based on profile markers on the profile
- Learn about business processes support in ATG
- Understand workflows

Section 1

Overview of Custom Scenario Events, Actions, and Conditions

Custom Events, Actions, and Conditions

- Events are instances of site visitor behavior that the business manager wants the scenario to watch for:
  - A visitor to register at the site or to go to a specific page. “Registers” and “Visits page” are both scenario events.
- Actions are instances of system behavior - often what the system does in response to an event.
  - A scenario might watch for visitors to register, and then send an email. “Send Email” is an example of a scenario action.
- Conditions are filters that qualify the previous event element in a scenario.
  - You could follow the event “Views any item” with the condition “Browser’s type is Netscape Navigator.”
- If events, actions, and conditions do not meet your requirements, you can add your own.

Adding Custom Events

- To add your custom event to the scenarios module, do the following:
  1. Create a JavaBean class that represents your custom JMS message.
  2. Create a Nucleus component implementing MessageSource that will generate your custom JMS message.
  3. Add your custom event to the scenario event registry.
  4. Add your message and message source to the appropriate Dynamo Message System (DMS) configuration file.
  5. Configure the Scenario Manager to receive your message.

Adding Custom Actions

- To add your custom action to the scenarios module, do the following:
  1. Add the new action to the Scenario Manager configuration file.
  2. Implement the action interface (atg.process.action.Action).
  3. Update the configuration file to include it, then the action appears in the Scenarios area of the ACC.

Adding Custom Conditions

- To add your custom condition to the scenarios module, do the following:
  1. Add the new condition to the condition registry in the Scenario Manager configuration file.
  2. Extend the abstract class atg.process.filter.ExpressionFilter for your new condition.
  3. Optionally, extend the scenario module's grammar expression editor to provide more elegant handling of the new custom condition in the ACC.

Section 1 
Check Your Understanding

Why would you want to add custom actions, events, and conditions?

Answer: If events, actions, and conditions do not meet your requirements, you can create your own.

Section 1 
Check Your Understanding

What interface needs to be implemented for generating your custom JMS message?

Answer: MessageSource interface.

Section 1
Check Your Understanding

What interface needs to be implemented for an action?

Answer: Implement the atg.process.action.Action interface.

Section 1 
Check Your Understanding

Which abstract class needs to be extended to implement a custom condition?

Answer: Extend the abstract class atg.process.filter.ExpressionFilter for your new condition.

Summary

- If the ATG OOTB events, actions, and conditions do not meet your requirements, you can create your own.
- Custom Event creation includes creating the javaBean, MessageSource, and registering the event with ATG.
- Adding a custom action involves adding a new action to the scenario manager configuration, implementing the Action interface, and updating the ACC configuration file.
- To add a custom condition, you should add the condition in the registry, extend the ExpressionFilter class.

Section 2

Overview of Collection Filtering

Collection Filtering Overview

- Collection filtering is the process of reducing objects in a collection based on a specified condition.
- When a collection filtering component receives a collection, it determines which objects satisfy the condition by having matching criteria.
- All matching objects become part of the result set collection while the remaining objects are “filtered out” or discarded.
- When you want to filter a collection based on several conditions, you can create one complex filter or create a filter chain.

How Collection Filtering Works for Scenario

- This example describes how a collection filter limits the contents in a slot using a scenario.

How Collection Filtering Works Using Droplets

- This example uses a JSP to access a filled slot and filter the contents within it:

```
<dsp:droplet
      name="/atg/collections/filter/droplet/StartEndDateDroplet">
  <dsp:param name="collection" beanvalue="Articles"/>
  <dsp:param name="collectionIdentifierKey" value="date"/>
  <dsp:oparam name="output">
    Featured Articles:<p>
    <dsp:droplet name="/atg/targeting/TargetingForEach">
      <dsp:param bean="/atg/registry/Slots/Articles‟ name="targeter"/>
      <dsp:oparam name="output">
        <dsp:valueof param="filteredCollection"/>
      </dsp:oparam>
    </dsp:droplet>
  </dsp:oparam>
  <dsp:oparam name="empty"> There are no articles today.</dsp:oparam>
</dsp:droplet>
```

Using Collection Filtering Classes - CachedCollectionFilter

- Each collection filtering class relies on the base abstract class *atg.service.collections.filter.CachedCollectionFilter* for the ability to filter collections.
- This class is designed to receive a collection, remove items from the collection, and create a result set.
- It's possible to pass a profile to this class so that subclasses can use profile property values as filter conditions.
- This class is capable of caching the filtered collection.

OOTB Subclasses of CachedCollectionFilter

- ATG comes with several subclasses of CachedCollectionFilter to filter objects in a collection:
  - atg.service.collections.filter.StartEndDateFilter limits a collection by removing objects that are inactive.
  - atg.service.collections.filter.ChainedFilter lets you chain several filters together so the resultant collection contains items that meet the conditions of all filters in the chain.
  - atg.commerce.gifts.GiftlistSiteFilter filters gift lists and gift items in a multisite environment.

Caching Filtered Content

- When you filter a collection in a JSP, you can cache the filtered identical results.
- A filter that uses caching:
  - Attempts to reuse content cached by a previous invocation that filtered an identical collection.
  - When no cached content can be reused, the filter executes and saves its resultant collection to the cache.
- You can even cache for chained collection filters.
- Caching is most effective for content that requires expensive, re-usable resources or computations to generate it.
- Caching is a waste of resources for collection filtering components when the input collection changes frequently.

Implementing Custom Collection Filters

- When creating collection filters, you may want to keep them simple and modular by limiting them to work with one or two conditions only.
- You can then link as many filters together as you like in a chain.
- To create custom collection filters, complete the following tasks:
  1. Create a collection filtering component.
  2. Configure your collection filtering component.
  3. Create a resource that can access the collection filtering component.

Section 2 
Check Your Understanding

How do you filter out content from a result set?

Answer: Using collection filters.

Section 2
Check Your Understanding

How do you enforce multiple complex filter conditions?

Answer: You can chain collection filters to achieve this.

Section 2 
Check Your Understanding

What is the base abstract class that collection filters rely on?

Answer: atg.service.collections.filter.CachedCollectionFilter .

Section 2 
Check Your Understanding

Name a few sub classes of collection filter.

Answer: StartEndDateFilter, ChainedFilter, GiftlistSiteFilter, etc.

Section 2 
Check Your Understanding

How does a filter that uses caching deal with collections?

Answer: It attempts to locate a previously cached collection. If not, it creates and caches the collection.

Summary

- Collection filtering is the process of reducing objects in a collection based on a specified condition.
- When you want to filter a collection based on several conditions, you can create one complex filter or create a filter chain.
- Collection filters can be used in scenarios and using droplets.
- The abstract class atg.service.collections.filter.CachedCollectionFilter should be extended to implement collection filters.
- ATG provides several sub classes of CacheCollectionFilter for extension.

Section 3

Using Profile Markers

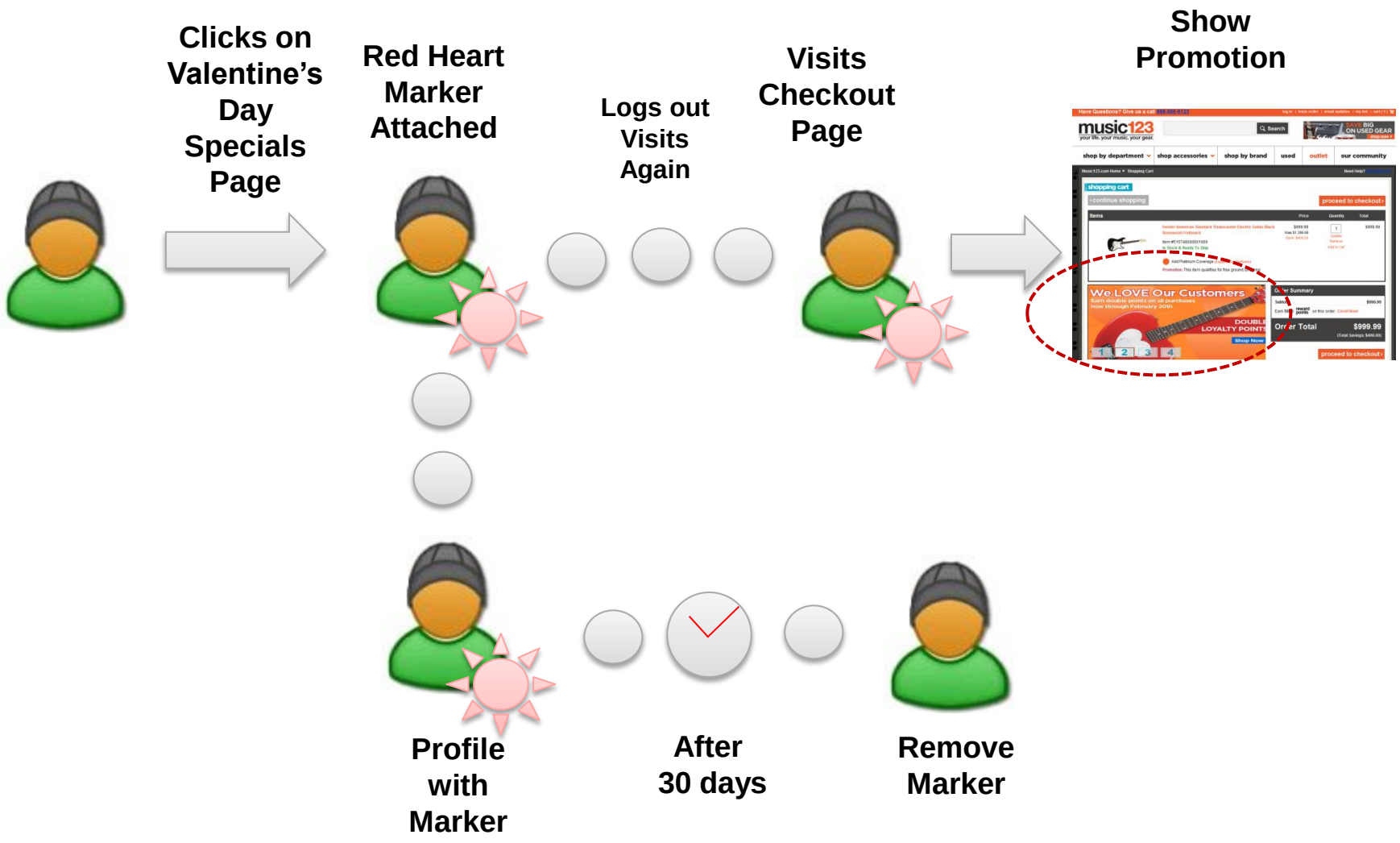
Profile Markers Overview

- A profile marker is a marker RepositoryItem that attaches to a user profile.
- Use profile markers when you want to “mark” a user’s profile with some information.
- You can also gather data using profile markers.
- You can design your sites to change their appearance or behavior based on the profile markers in a user profile
- Profile markers are flexible and contain key, value, and data fields.
- ATG ships with a wide variety of tools to use profile markers using droplets and in scenarios.

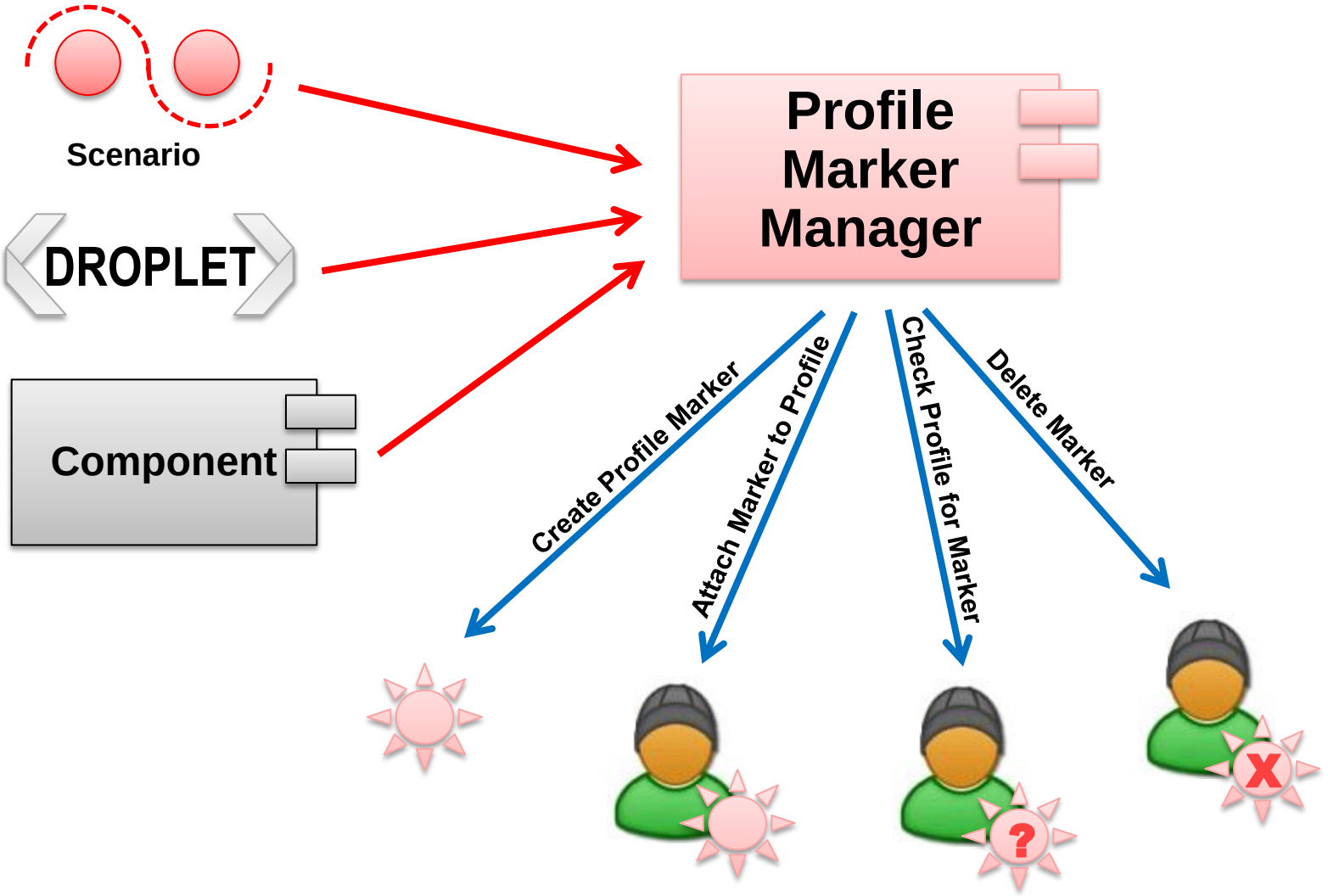
How to Use Profile Markers

- Profile Marker Manager facilitates all interaction between markers and profiles, such as creating, removing, and querying the profiles.
- You can create one or more markers and attach them to profiles either through a scenario, a JSP page using a droplet, or Java code.
- Your application can check profiles for specific markers in Java code, using droplets on JSP pages, or using scenarios.
- To optimize performance, you should remove a marker when no longer needed using Java code, droplet or scenario.

Using Profile Markers (1)



Using Profile Markers (2)



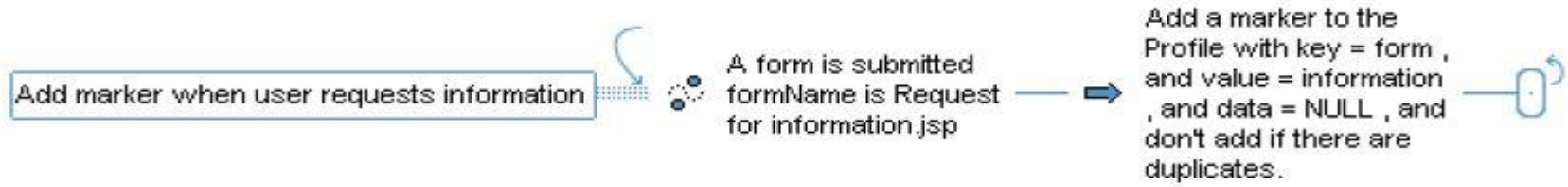
Profile Marker Manager

- Profile Marker Manager is a component that manages how markers interact with profiles:
  - Adds a new marker to a user profile.
  - Queries for information about the markers on the marked profile.
  - Removes an existing marker that's no longer used.
  - Applies validation and duplication rules.

- Default profile marker manager:
  - atg/markers/userprofiling/ProfileMarkerManager .
  - You may choose to create additional profile marker manager components.

Marking a User Profile

- Using scenario:



- Using droplet:

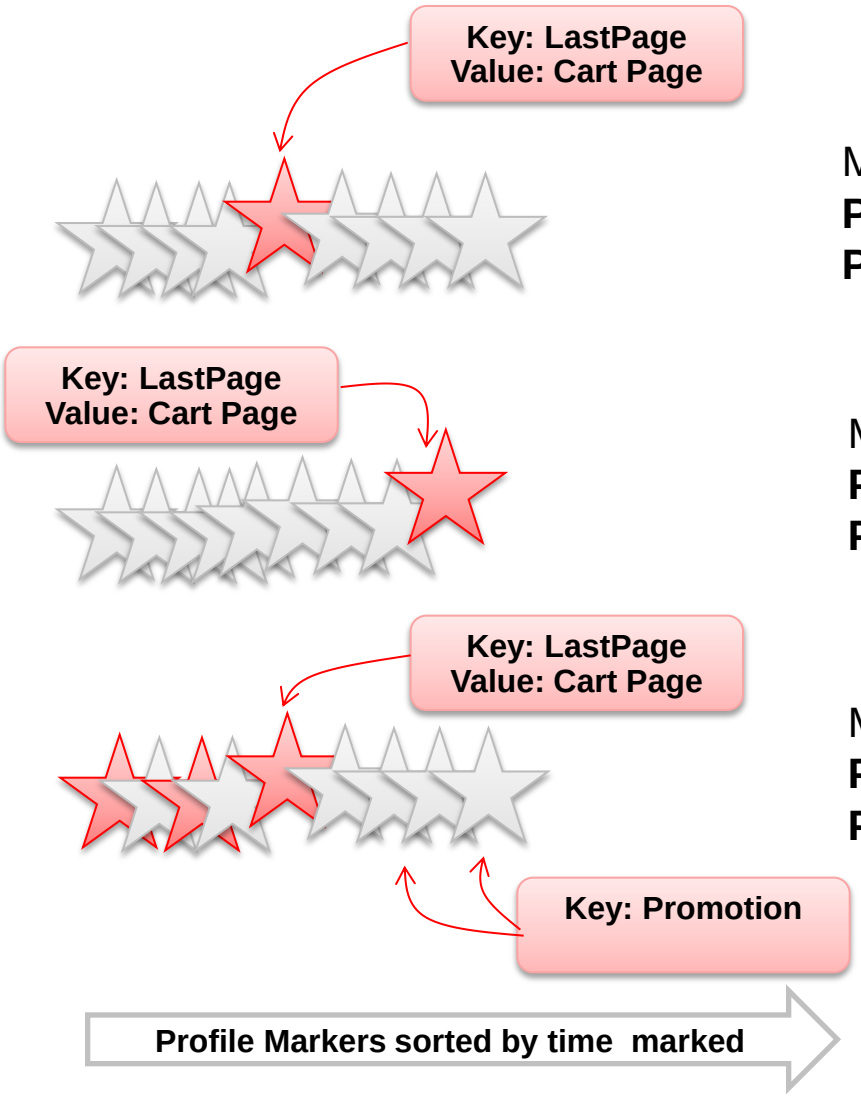
```
<dsp:droplet
      name="/atg/markers/userprofiling/droplet/AddMarkerToProfileDroplet">
   <dsp:param name="key" value="form"/>
   <dsp:param name="value" value="information"/>
   <dsp:param name="duplicationMode" value="NO_DUPLICATES"/>
</dsp:droplet>
```

Using Droplet and Scenarios to Find Markers

- When you use servlet beans, you can specify an input parameter that indicates the profile you want to work with. If you choose not to, the active user profile is used.
  - ProfileHasMarkerDroplet
  - ProfileHasLastMarkerDroplet
  - ProfileHasLastMarkerWithKeyDroplet

- Use profile marker conditions in scenarios when you want the scenario to advance only if the active user has specific markers:
  - Profile has a marker condition,
  - Profile's last marker condition,
  - Profile's last marker with key condition.

Finding Markers



Marker being evaluated for:
Profile has Marker Droplet
Profile has Marker Condition

Marker being evaluated for:
Profile has Last Marker Droplet
Profile has Last Marker Condition

Marker being evaluated for:
Profile has Last Marker with Key Droplet
Profile has Marker with Key Condition

Finding Markers (2)

- ProfileHasMarkerDroplet *and* Profile marker condition.
  - Only users with a specific marker with key, value, and data qualify.
  - The marker could have been placed on the profile at any time.

- ProfileHasLastMarkerDroplet *and* Profile's last marker condition.
  - Only users whose last marker has the key, value, and data values qualify.
  - Only the last marker placed on the profile chronologically is evaluated.
  - Even if the profile has the required marker but it was not the last one, the profile does not qualify.

Finding Markers (3)

- ProfileHasLastMarkerWithKeyDroplet *and* Profile's last marker with key condition.
  - Only users whose last marker with a particular key have the value and data values specified qualify.
  - The last marker on the profile with the specified key is evaluated.
  - This is the most commonly used variant.
  - It protects the developer from getting affected by a change that inadvertently places another marker on the profile.
- For all droplets and conditions, the value and data are optional. They can take an ‘any’ value.

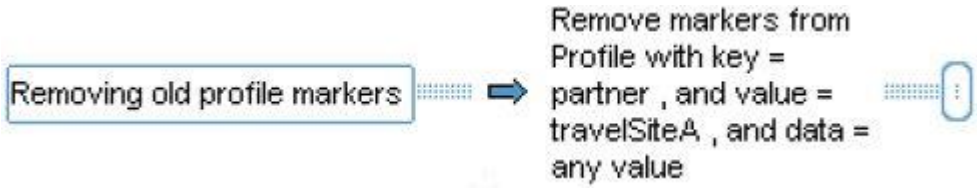
Advancing a Scenario Based on Profile Marker Events

- There are three scenario elements that listen for the following events:
  - **Profile marker added**: A marker was added to a profile.
  - **Profile marker removed**: A marker was removed from a profile.
  - **Profile marker replaced**: One marker was replaced on a profile with another marker.

- You can create a scenario that advances when it receives a profile marker event.

Removing Profile Markers

- Remove the unused markers to ensure performance.
  - Remove specific markers from a given profile:



  - Remove all markers on a profile:

- Profile Markers can also be removed using droplets and Java code.

When to Use Profile Markers

- Custom Profile Attributes and Profile Markers generally provide a similar functionality.
- Custom Profile Attributes are:
  - A build time activity. They need to be designed and implemented by developers.
  - They are lighter on the database as they are columns in a table.
- Profile Markers are:
  - A run time activity. They can be created at will.
  - They are heavier on the database as they are rows in a table.
  - Profile markers should be removed and archived from the DB.
- When a large subset of your users have an attribute, use Custom Profile Attributes.
- When a small subset of your users have an attribute, use Profile Markers.

Section 3
Check Your Understanding

What type of an object is a profile marker?

Answer: Profile marker is a repository item.

Section 3 
Check Your Understanding

What are the various ways to create and attach a marker to a profile?

Answer: Using a JSP page, Java component, and droplet.

Section 3 
Check Your Understanding

Name a few droplets that ATG comes with to handle profile markers.

Answer: ProfileHasMarkerDroplet, ProfileHasLastMarkerDroplet, and ProfileHasLastMarkerWithKeyDroplet.

Section 3 
Check Your Understanding

List a few scenario conditions that ATG ships with.

Answer: Profile has a marker condition, Profile's last marker condition, and Profile's last marker with key condition.

Section 3 
Check Your Understanding

What three scenario elements listen for events?

Answer: Profile marker added, profile marker removed, and profile marker replaced events.

Summary

- A profile marker is a marker RepositoryItem that attaches to a user profile.
- Profile markers are flexible and contain key, value, and data field.
- Profile Marker Manager facilitates all interaction between markers and profiles, such as creating, removin,g and querying the profiles.
- You can create a profile marker, attach it to profile, check if a profile has marker, and delete a marker from either a Java component, droplet, or scenario.
- Profile Markers provide a more flexible approach than custom profile attributes to deal with requirements.

Section 4

Overview of Business Processes

Defining And Tracking Business Processes

- ATG's Adaptive Customer Engine includes business tracking features.
- It lets you:
  - Define a business process as a series of stages.
  - Track activity within the business process.
  - Report on the activity for a specified time frame.
- It enables the system to track progress or failure to progress in the business process.
- You can personalize a user's experience based on where they are in a business process.

How Business Process Tracking Works

- A business process should be defined as a series of stages applied to a business object.
- The business object is defined as a repository item type:
  - A profile or an order.
- As the business object reaches a new stage, the corresponding repository item is marked to indicate the process and stage reached.
- A message is sent identifying the business process, the repository item, and the new business process stage being reached.

Defining A Business Process

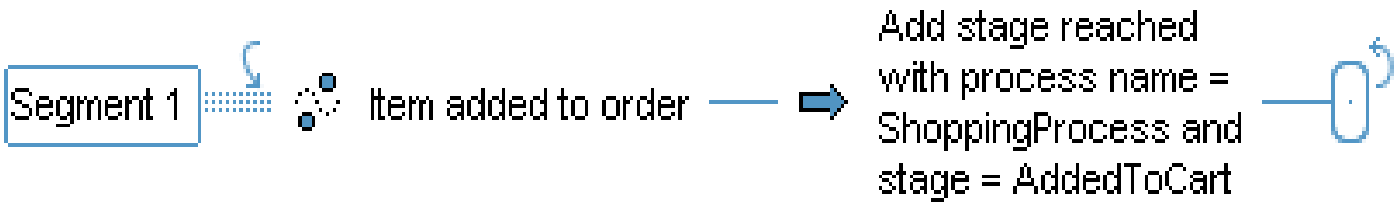
- Define a business process and track the progress of items through following business process:
  - Identify the object of the business process (a profile, order, or other type of repository item).
  - Create and configure a BusinessProcessConfiguration component to define the business process.
  - Add the BusinessProcessConfiguration component to the businessProcessConfigurations property of the BusinessProcessManager.
  - Set up a process for adding business process stage markers to your business process object.
  - Create a scenario recorder to record business process stage reached events, together with reports derived from the dataset.

Marking Business Process Stages

- Page-based and scenario-based tools can add, remove, and check for business process stages.
  - Page-based:

```
<dsp:droplet name="AddBusinessProcessStage">
  <dsp:param name="businessProcessName"
  value="ShoppingProcess"/>
  <dsp:param name="businessProcessStage" value="AddedToCart"/>
</dsp:droplet>
```

  - Scenario-based:



Deleting Business Process Content

- ATG provides two API methods in the BusinessProcessManager class for business process content deletion.
  - RepositoryItem-Based Deletion:
    - Delete business process content from a specific RepositoryItem .

```
removeBusinessProcessStage("MyBusinessProcess", "StageA")
```

  - Business Process-Based Deletion:
    - Delete content associated with stages from a given business process on every affected RepositoryItem .

```
deleteBusinessProcessMarkers("MyBusinessProcess", "StageB")
```

Reporting on Business Processes

- Tracking business process allows you to generate, study, and learn from reports about what happens at different stages of the business process.
- To generate reports about business processes:
  - Create a scenario recorder, including a scenario, data mapper, and dataset to store business process information.
  - Create reports based on the dataset.
- Using these reports you can detect when users are dropping out of a business process and take steps to encourage them to resume the business process.

Section 4 
Check Your Understanding

What is a business object in the context of a business process?

Answer: It is a repository item type such as profile or order.

Section 4 
Check Your Understanding

Which component should you create and configure to define the business process?

Answer: Create and configure a BusinessProcessConfiguration component to define the business process.

Section 4

Check Your Understanding

Which component and property holds a list of business processes?

Answer: BusinessProcessConfigurations property of the BusinessProcessManager holds the list..

Section 4 
Check Your Understanding

How can you mark a stage as reached on a JSP Page?

Answer: Using the AddBusinessProcessStage Droplet.

Section 4 
Check Your Understanding

What are the two ways to remove business process content?

Answer: Either for a specific repository item or for all repository items associated with a state.

Summary

- ATG's Adaptive Customer Engine includes business tracking features.
- It enable the system to track progress or failure to progress in the business process.
- To define a business process, identify the object, configure the BusinessProcessConfiguration, and set up a process for adding stage markers to object.
- You can use a droplet or a scenario to mark stages as reached.
- ATG provides mechanisms to report on business processes.

Section 5

Workflows Overview

Workflows Overview

- The scenarios module includes a mechanism for modeling business processes called a workflow.
- They are based on ATG's process engine architecture like scenarios.
- Most workflow classes are analogous to scenario classes.
- Major difference between scenarios and workflows:
  - There is only a single scenario type for which you can create custom events and actions.
  - You can create any number of different workflow types with the workflow classes, each with its own actions and tasks, repository items, and database tables.

Creating a Workflow Type

- Choose the workflow subject type:
  - The subject type is a repository item that represents the main element in the process that will be modeled by the workflow.
  - An example is a registration request repository item.
- Create the repository items and database schema:
  - Each workflow type must define its own data schema and repository items.
- Configure the workflow components:
  - Each workflow type requires configuration of several nucleus components and XML files.
- Create the workflow definition in the ACC:
  - For each workflow type the workflow editor is customized to expose the relevant item type and its properties.
  - Only those custom expressions and actions that are relevant are exposed.

Registration Workflow Example(1)

- The EcoVida registration workflow type show a typical workflow type.
- This workflow type models the process of registering a new partner (builder or reseller) in a portal community.
  - Step 1: The prospective reseller comes to the EcoVida partner portal and submits an application.
  - Step 2: The EcoVida portal initiates an approval workflow.
  - Step 3: Because the partner is in the East region, the application is routed to the Eastern regional channel manager for review. Mandy, the channel manager, claims the task.
  - Step 4: Mandy completes the reseller approval task.

Registration Workflow Example(2)

- Registering a new partner (builder or reseller) in a portal community.
  - Step 5: The workflow then routes the registration request to a finance manager to perform a credit check. Frank, a finance manager, claims the task and does a credit check.
  - Step 6: Frank approves the reseller.
  - Step 7: Mandy is notified that the partner has passed the credit check and been approved.
  - Step 8: A personalized email is sent to the partner informing them of their approval.
  - Step 9: The workflow ends.

Workflow Classes

- The following are the classes and interfaces that you use to work with workflows:

Classes and Interfaces	Description
atg.workflow.WorkflowManager	The primary interface that exposes features of the workflow system.
atg.workflow.WorkflowView	Exposes workflow features to a user of the workflow system.
atg.workflow classes and interfaces	Used to expose various workflow-related information.
atg.workflow.servlet classes	Include the workflow servlet bean and form handler classes.
atg.process.ProcessManager	This interface includes several methods that provide workflow support.

Workflow Servlet Beans

- The workflow API includes servlet beans that you can use to create custom UIs for manipulating workflows.
  - WorkflowTaskQueryDroplet
    - Perform workflow task queries.
  - WorkflowInstanceQueryDroplet
    - View all existing workflow instances.
  - ItemLookupDroplet
    - Looking up properties of the workflow subject given the subject ID.
  - GetDirectoryPrincipal
    - Resolving user directory principals.

Workflow Task Form Handler

- WorkflowTaskFormHandler
  - Used to perform workflow task operations such as setting a task's owner and firing outcomes.
  - The submit methods are:

Method	Description
setTaskPriority	Sets the task's priority.
setTaskOwner	Sets the task's owner.
claimTask	Claims the task on behalf of the current user's view.
releaseTask	Releases the task on behalf of the current user's view.
fireOutcome	Fires the task outcome identified.

Examples of Workflows

- The ATG Content Administration provides a workflow that manages the lifecycle of a publishing project.
- ATG provides an example workflow for EcoVida that you can use as a starting point.
- Many other different types of business processes can be workflows.
  - Commerce order fulfillment.
  - Managing customer support calls.

Section 5 
Check Your Understanding

How many scenario types and workflow types can you have in your application?

Answer: You can have only one single scenario type. There can be multiple workflow types.

Section 5
Check Your Understanding

What is a subject type in the workflow?

Answer: The subject type is a repository item that represents the main element in the process that will be modeled by the workflow.

Section 5

Check Your Understanding

Where are the workflow objects stored in the data schema?

Answer: Each workflow type must define its own data schema and repository items.

Section 5 
Check Your Understanding

What is the primary interface that exposes features of the workflow system?

Answer: The WorkflowManger.

Section 5 
Check Your Understanding

List a few servlet beans available to work with workflows.

Answer: WorkflowTaskQueryDroplet, WorkflowInstanceQueryDroplet, ItemLookupDroplet, etc.

Section 5
Check Your Understanding

What is the purpose of the WorkflowTaskFormHandler?

Answer: Used to perform workflow task operations such as setting a task's owner and firing outcomes.

Summary

- The scenarios module includes a mechanism for modeling business processes called a workflow.
- Workflows are similar to scenarios, but can be applied to a wider range of processes.
- You can create any number of different workflow types with the workflow classes, each with its own actions and tasks, repository items, and database tables.
- To create a workflow choose a subject type, create a repository, configure workflow components, and create the workflow definition.
- ATG provides workflow servlet beans and form handlers.

ORACLE IS THE INFORMATION COMPANY


ORACLE®