
ORACLE®

# Introduction to ATG Nucleus

# 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

- Nucleus
- Nucleus Component

# Learning Objectives

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

- Understand what Nucleus does
- Create Nucleus components
- Configure Nucleus components
- Reference Nucleus components
- Display and update component properties in the Admin console (Dynamo Administration/ACC)

# Section 1:
# Nucleus and Components

# Nucleus

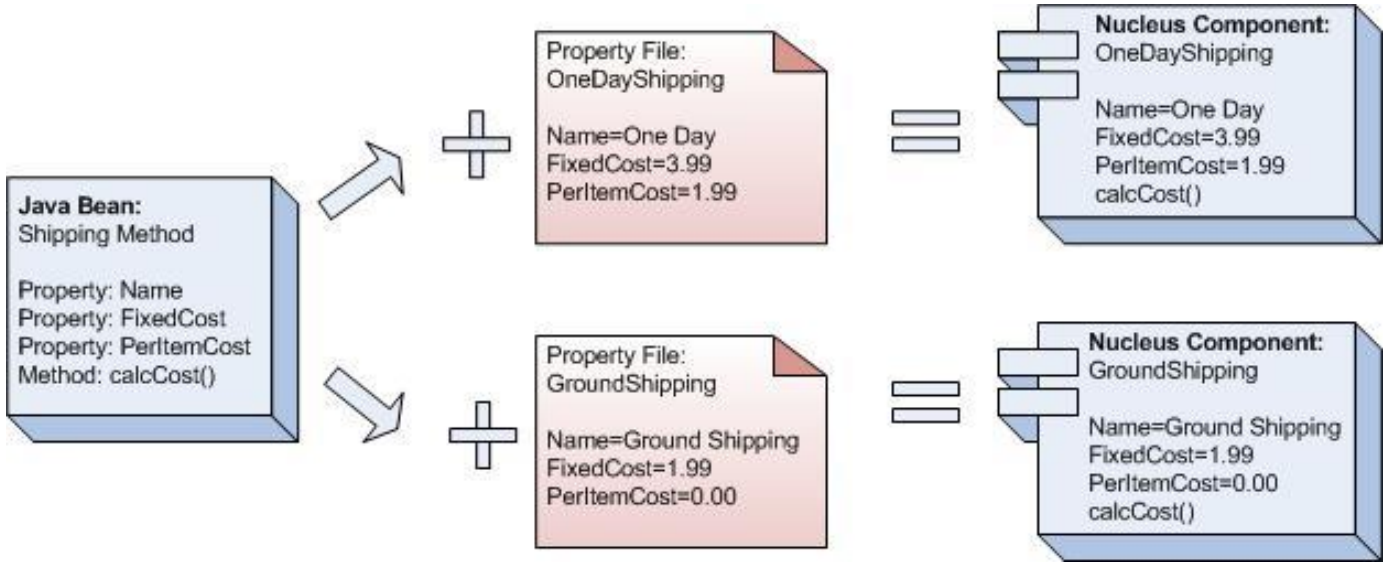
- Nucleus is ATG's base component model for building applications from Java Beans.
- Nucleus by itself provides no application-specific functions. Java Bean components implement the functions.
- Nucleus organizes application components into a hierarchy.
- Each nucleus component has a unique full name so that pages and other nucleus components can reference it via its pathname.
- Nucleus manages component scope.
- Nucleus manages the task of creating and initializing nuclues components based on configuration files. It does so when the application is first initialized (i.e. server startup).
- Any simple JavaBean can be a nucleus component. By implementing various interfaces, components can take advantage of Nucleus services and notifications.

# The Features of Nucleus

- ATG Nucleus performs one basic operation: **Resolving Component Names**
- Given a fully qualified name, Nucleus looks at the path, then finds and creates a nucleus component.
- Nucleus also creates nucleus components and links them up to other components automatically when configured.
- Nucleus depends on a list of configuration directories, called the CONFIGPATH a system variable, to find the nucleus component.
- Nucleus components are configured through configuration text files i.e. ".properties" file. This provides a powerful **decoupling** mechanism.

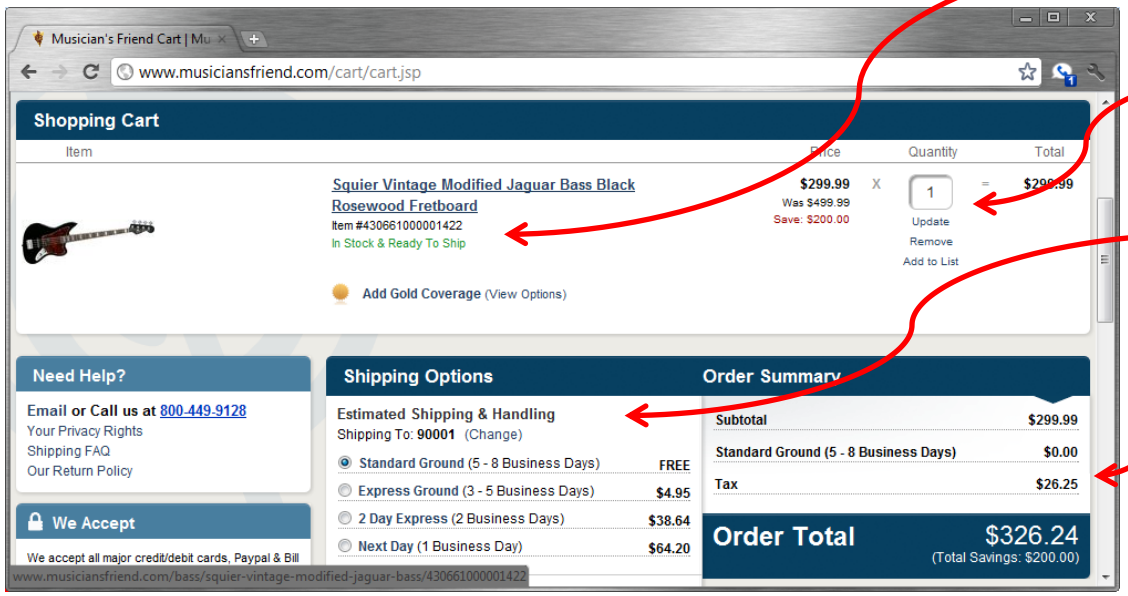
# What is a Nucleus Component?

- Nucleus Components are standard JavaBeans.
- The nucleus component is defined by a ".properties" file that stores configuration information for use during its instantiation.
- Multiple Nucleus Components can be created from the same JavaBean class.



# Example of Nucleus Component

- All business requirements can be modeled as Nucleus Components.
- Some examples of the Nucleus Components on the page: form handlers, droplets, service components.



# Sample Nucleus Component

- Define the Class

```
public class ShippingMethod {
    String name;
    float fixedCost;
    public ShippingMethod() {}
    public String getName () { return name; }
    public void setName (String name) { this.name = name; }
    public float getFixedCost () { return fixedCost; }
    public void setFixedCost (float fixedCost) {
       this.fixedCost = fixedCost;
    }
}
```

- Configure a Component

```
$class=ShippingMethod
name=One Day
fixedCost=3.99
```

# What can be specified in the properties file

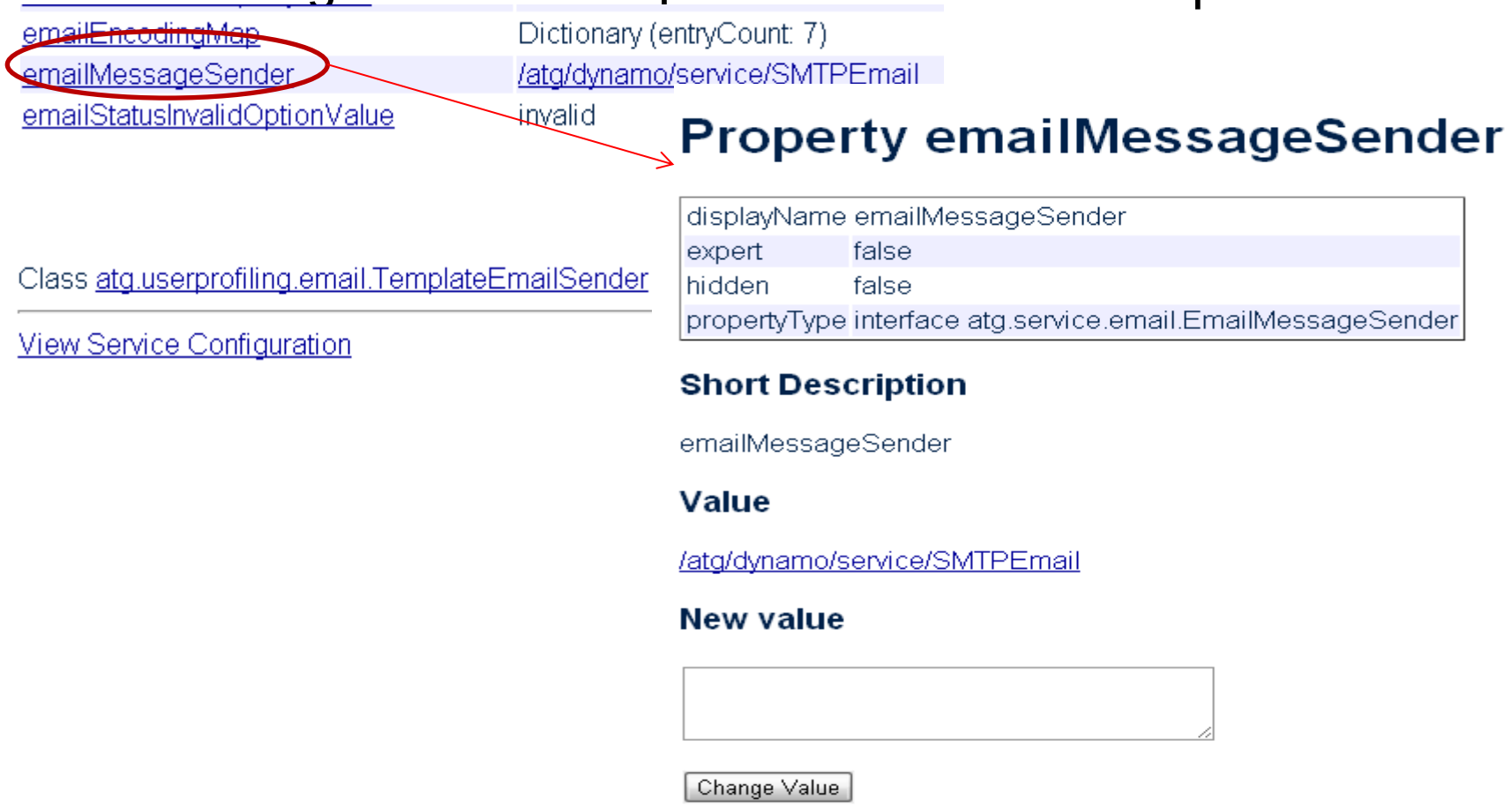
- The java class that the nucleus component represents
- Scope of the component (request/session/global/window/prototype)
- The properties of the nucleus component with basic java data types like: int, float, String etc
- Data Collections for use by the nucleus component with java utils like map, list, set etc
- References to other nucleus components. This is how nucleus provides linkage between components. Nucleus will instantiate the referenced component when needed and the reference component itself will have its own property file with all of the above specified.

# Configuring A Component

- The admin console allows you to view and update initial values for the properties defined in the Java class.

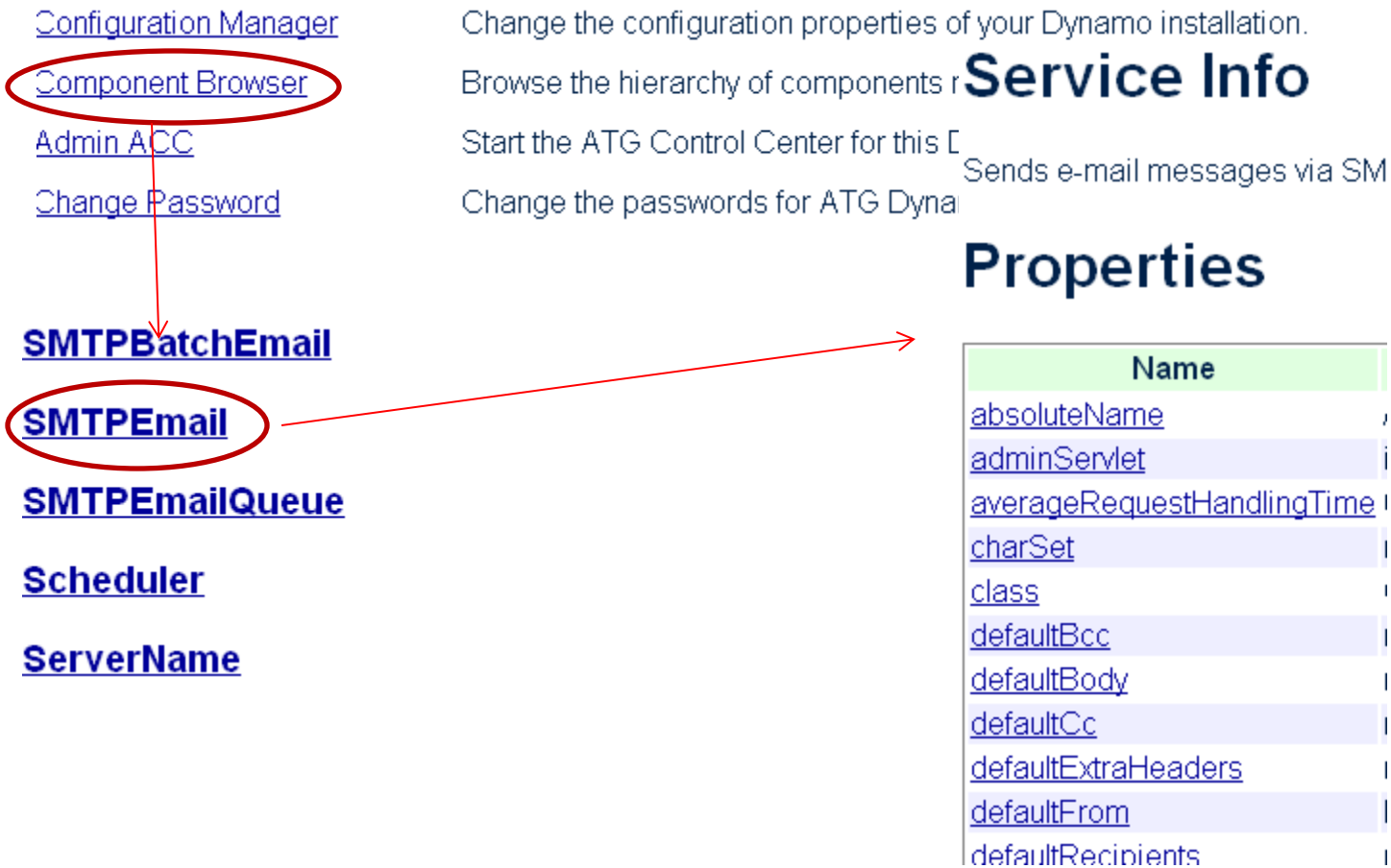
# Linking to Another Component

- You can link a property to another component by referencing the nucleus path of the other component.



# View Component in Admin Console

- You can view components in the Dynamo admin console.



# Section 1 ✔
# Check Your Understanding

What is the operation provided by the Nucleus?

a. Provide core business functionality
b. Provide a basic customizable shopping cart
c. Resolving component names
d. Provide IOC, ORM and Personalization
e. Nucleus provides the base classes for an ATG implmentation

Answer: c

# Section 1
# Check Your Understanding

How does Nucleus provide linkage between components?

a. Nucleus determines the inter dependency and injects the component automatically when needed
b. A IOC framework provides linkages. Nucleus does not do that.
c. Nucleus provides the linkage between components by configuring a reference to that component.
d. The calling code can request nucleus to another component. This process will link the two components together
e. Nuclues contains a configured map, which contains the component as key and the linked component as the value.

Answer: c

# Section 1
# Check Your Understanding

How is decoupling achieved in ATG?

a. Using the spring framework in addition to ATG, decoupling can be achieved.
b. Using an MVC framework such as struts in addition to ATG, decoupling can be acheived.
c. By using property files and linking components, decoupling of code and configuration can be achieved.
d. By using the scenario engine to instantiate components when needed, decoupling of view from business logic can be achieved.

Answer: c

# Section 1
# Check Your Understanding

In addition to class, scope and links to other components, what other attribute values can be specified in a property file?

a. No attribute values can be specified in the property file.
b. Only attributes values for primitive data types such as int, float, etc.
c. Attribute values such as int, float, string, map, list, etc.
d. Attribute values such as int, float, string, map, list and class names for attributes that take object values.

Answer: c

# Section 1
# Check Your Understanding

What is the minimum requirement for creating a Nucleus component?

a. Any class can be a nucleus component
b. Any JavaBean can be a nucleus component
c. A class extending GenericService is required
d. A class implementing NucluesComponent is required
e. A class should either extend GenericService or implement NucleusComponent to be a nuclues component

Answer: b

# Summary : Nucleus

- Nucleus is a key ATG feature for building component-based web applications.
- Nucleus allows for decoupling of code and configuration.
- Multiple components can and usually are created using the same JavaBean class.
- Property files should have JavaBean references and attributes required for initialization.
- Property files may have references (links) to other Nucleus components.
- A dynamic web site is made up of multiple nucleus components which encapsulate the business logic.

# Section 2:
# Creating Nucleus Components

# Steps to create a Nucleus Component

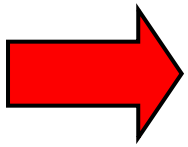
1. Identify the business requirements that the component will fulfill.
2. Identify other Components that could provide functionality to support the requirement.
3. Design and create a JavaBean class.
4. Identify and create the property file in the configuration path.
5. Author the property file.
6. Link to other components for additional functionality as identified.
7. Ensure that the property file is in the CONFIGPATH.

# What is a JavaBean?

- JavaBean is a regular java class that forms the basis of a nucleus component.
- A nucleus component must be a JavaBean.
- A JavaBean is a Java class with:
  - a no-argument constructor
  - class must be public
  - properties implemented by **`get`** and **`set`** methods

methods:

```
public String getSomething () …
public void setSomething (String)
…
```

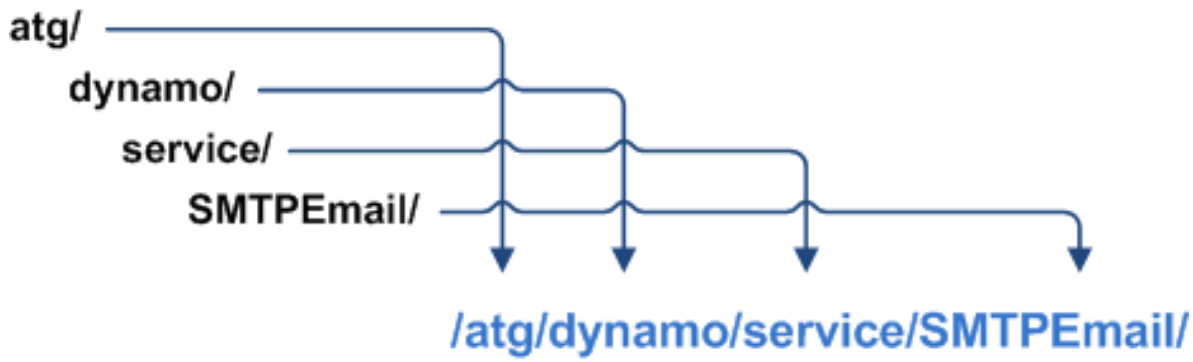


property:
something

- Nucleus components can and often do extend other helper JavaBean classes to gain additional functionality.

# Component Names

- Full component names are unique, containing folder names followed by the simple component name, separated by forward slashes.




- The fully qualified component path is how ATG's framework API is referenced.
- A component called EmailSender in the above directory structure would be referred to as:

  /atg/dynamo/service/SMTPEmail/EmailSender
- The property file in the respective module would be located at:

  /atg/dynamo/service/SMTPEmail/EmailSender.properties

# Java Properties File

- Name/value pairs are separated by = or :
- Values can span multiple lines if terminated by “\”
- White space is ignored at
  - Beginning of line
  - Between property and value (a = 2 and a=2 are the same)
- ! or # at the beginning of the line signifies a comment. ATG specific properties start with $

```
# Example of a property file

myProperty=2
Email.address=test@test.com
emailText=Hello,\
This is an email from my property\
file.
```

# ATG Component Properties File

- Nucleus-specific properties like class name, scope type are prefixed by the $ character.
- White space that follows the property value is treated as part of the property value.
- Certain characters and strings are given special treatment, as described in the following table:

| | |
|---|---|
| ! # | If placed at the beginning of a line, comments out the line. |
| \n | Newline character |
| \r | Carriage return |
| \t | Tab |
| \\ | Inserts a backslash character. For example: path=c:\\docs\\doc1 |
| \u | Prefixes a UNICODE character—for example, \u002c |

# Example of creating a simple Nucleus Component

- To create a nucleus component called ShippingOption
  - Create a JavaBean called com.site.shipping.ShippingOption.java.
  - Define, get, and set methods for properties, i.e. name and price.
  - Create a property file in /site/shipping/OneDayShip.properties.
  - Define the java class, name, and price properties.
- The property file is shown as below.

```
$class=com.site.shipping.ShippingOption
$scope=global
name=One Day
price=2.99
```

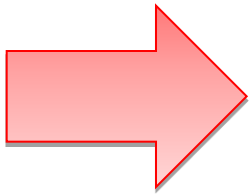
# Simple Properties Types Configuration

- Nucleus components can have configurations for simple properties.
- Simple properties include java primitive type and primitive wrapper type, etc.

If your class has:

```
// Class Person
public String getName() {
  return name;
}

public void
  setName(String name) {
   this.name = name;
}
```



The property file should have:

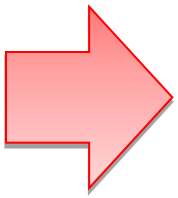
```
$class=Person
name=Tom
```

# Array Properties Types Configuration

- Nucleus components can have configurations for array properties.
- Array Properties include java utils like Array, List, and Set.

If your class has :

```
// Class Friends
public String[] getNames() {
   return names;
}
public void
 setNames(String[] names){
   this.names = names;
}
public void
  setCities(String[] cities) {
    this.cities = cities;
}
```



The property file should have:

```
$class=Friends
names=Allen,\
      Tom,\
      Terry
cities=LA,SB,SD
```

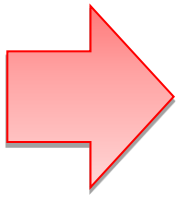
# Map Properties Types Configuration

- Nucleus components can have configurations for map properties.
- Map Properties include java utils like Map and Hashtable.

If your class has :

```
// class CityMap
public Map getNamesMap(){
   return namesMap;
}

public void
  setNamesMap(Map namesMap){
    this.namesMap = namesMap;
}
```



The property file should have:

```
$class=CityMap
namesMap=\
   LA=Los Angeles,\
   SB=Santa Barbara,\
   SD=San Diego
```

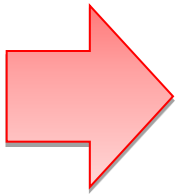
# Properties For Other Component Configuration

- Nucleus component can reference other nucleus components.
- A good design leverages other components when possible.

If your class has :

```
// Class PromoEmailer
Public SMTPEmailSender
   getEmailSender(){
     return emailSender;
}

public void setEmailSender(
  SMTPEmailSender emailSender){
    emailSender = emailSender;
}
```



The property file should have:

```
$class=PromoEmailer

emailSender=\
   /atg/dynamo/\
   service/SMTPEmail
```

# Multiple Components From One Class

- One JavaBean class may be the source of multiple nucleus components.
- Depending on the initialization properties specified, each component can satisfy different business requirements by reusing the same JavaBean class.
- A good design captures the common aspects and reuses classes instead of creating new ones with similar functions.
- As an example, most shipping methods could be a single class with different initialization parameters.
- Reusing classes improves code quality, maintainability, and reduces defects.

# Section 2

# Check Your Understanding

What are the two basic steps for creating a Nucleus Component?

a. Create a JavaBean
b. Create a property file in the ATG configuration path, referencing the Java Bean
c. Register the JavaBean in the Initial.properties file
d. Register the JavaBean by calling the Nucleus.registerComponent method
e. Drop the JavaBean in the Nucleus configuration path

Answer: a, b

# Section 2

# Check Your Understanding

How does the location of the property file for the nucleus component relate to the component's fully qualified name?

a. The location of the property file has no bearing on the component name
b. The relative location from the config folder is the fully qualified name of the component.
c. The property file has to be in the configuration path. The fully qualified name is specified in the property file.
d. The JavaBean name and package name determines the components fully qualified name. The property file must be placed there.

Answer: b

# Section 2

# Check Your Understanding

Which of the following three are true for property files?

a. ATG internal properties start with $.
b. White space is ignored every where in the file
c. ! or # at the beginning of the line signifies a comment
d. Values can only be in a single line. Multiple lines are not allowed.
e. Name/value pairs are separated by : or =

Answer: a, c, e

# Section 2
# Check Your Understanding

Can collections like map, list, or array be in the configuration property file?

a. No. Only primitive data types like int, float are allowed.
b. Primitive data types and arrays are allowed
c. Primitive data types, string and arrays/lists are allowed
d. Primitive types, strings, and collections like arrays, lists and maps are allowed.

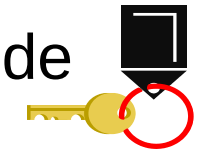
Answer: d

# Summary : Nucleus Component

- A nucleus component consists of a JavaBean class and a property file to create a Nucleus component.
- Nucleus component names contain hierarchical folder followed by the component name.
- Nucleus configurations in the properties file can include java primitive types, java Strings and java collections such as map, list, set, and array.
- Good design leverages other nucleus components by referencing them in the property files.

# Key Points

- Nucleus is a key ATG feature for building component based web applications.
- Nucleus allows for decoupling of code and configuration.
- Simple JavaBeans can be combined with configuration property files to make nucleus components.
- Properties within a configuration file can include java primitive types like int, long etc and java collections like map, list, set etc.
- Using Nucleus components, developers can decouple configuration from code and achieve code reuse.



ORACLE IS THE INFORMATION COMPANY


ORACLE®