# Oracle® iMarketing

Concepts and Procedures

Release 11*i*

August 2000

Part No. A86140-02

Oracle iMarketing Concepts and Procedures, Release 11*i*

Part No. A86140-02

# Contents

## Implementing Oracle iMarketing

# Send Us Your Comments

**Oracle iMarketing Concepts and Procedures, Release 11*i***

**Part No. A86140-02**

Oracle Corporation welcomes your comments and suggestions on the quality and usefulness of this document. Your input is an important part of the information used for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most?

If you find any errors or have any other suggestions for improvement, please indicate the document title and part number, and the chapter, section, and page number (if available). You can send comments to us via the postal service.

Oracle Corporation
CRM Content Development Manager
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

If you would like a reply, please give your name, address, telephone number, and (optionally) electronic mail address.

-----------------------------------------------------------------------------------------------------------------------------------

-----------------------------------------------------------------------------------------------------------------------------------

-----------------------------------------------------------------------------------------------------------------------------------

If you have problems with the software, please contact your local Oracle Support Services.

# Preface

Welcome to the Oracle iMarketing, Release 11i, suite of applications.

This *Concepts and Procedures Guide* provides information and instructions to help you work effectively with Oracle iMarketing.

This preface explains how this *Concepts and Procedures Guide* is organized and introduces other sources of information that can help you.

## Intended Audience

This guide is aimed at the following users:

- Technical Service Representatives (TSR)
- Customer Service Representatives (CSR)
- System Administrators (SA), and others with similar responsibility
- Web Site Managers

This guide assumes you have the following pre-requisites:

- Understanding of the company business processes
- Knowledge of products and services as defined by your marketing policies
- Basic understanding of Oracle and Developer/2000

## Structure

This manual contains the following sections:

“Understanding Oracle iMarketing" provides overviews of the application and its components, explanations of key concepts, features, and functions, as well as the application’s relationships to other Oracle or third-party applications.

“Using Oracle iMarketing” provides process-oriented, task-based procedures for using the application to perform essential business tasks.

## Related Documents

For more information, see the latest versions of the following manuals:

- *Oracle Marketing Online Concepts and Procedures*
- *Oracle iStore and iMarketing Implementation Guide*
- *Oracle Marketing Online Implementation Guide*
- *Discoverer 3i User Guide*

# Understanding Oracle iMarketing

This topic group provides overviews of the application and its components, explanations of key concepts, features, and functions, as well as the relationship of the application to other Oracle or third-party applications.

## Overview of Oracle iMarketing

Oracle iMarketing is a part of the Oracle Applications family, an integrated suite of applications that include both ERP (Enterprise Resource Planning) and CRM (Customer Relationship Management) applications. Oracle Applications are designed to transform traditional businesses into e-businesses.

Oracle iMarketing is an internet marketing application designed to increase online revenues, build customer loyalty, and extend company brand presence to other sites.

iMarketing is a tightly integrated add-on module to Oracle iStore and is not being sold stand alone in this release.

iMarkerting works like a personalization engine in the Oracle e-business suite. Merchants can use iMarketing to create and execute marketing campaigns targeted at customer segments on their storefronts.

Oracle iMarketing is integrated with Oracle Marketing Online. It leverages its functionality and shares its schema.

Oracle iMarketing helps merchants compete on the online marketplace. Merchants typically use Oracle iMarketing to

- turn visitors into buyers
- grow each customer to maximum value for the merchant

- build and solidify customer loyalty into a solid, long-lasting competitive advantage

# Overview of Oracle iMarketing Functionality

iMarketing provides the following functionality:

- Segment Management
- Campaign Management
- Personalization of the Visitor’s Experience
- Affiliate Management

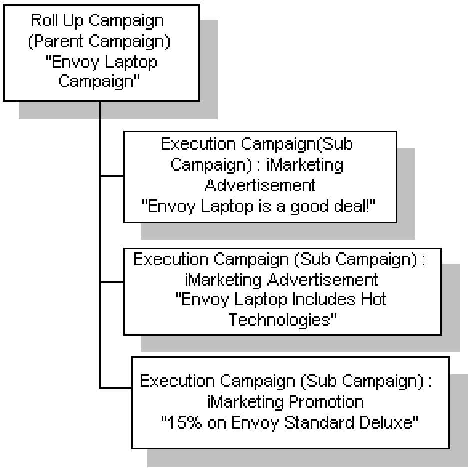
## Segment Management

In the present market, the focus has shifted from mass marketing to one-to-one marketing. This new level of customer intimacy is possible due to cheaper and faster data management, interactive media, and the ability to customize marketing campaigns to the individual. The result is the marketer knowing the customer better.

A campaign is refined by targeting people considered most likely to buy or be receptive to a promoted message or product. Within that target population (segment), the campaign can be further tailored to include personalized offers specific to the sub-population.

Segments are created within iMarketing. This segmentation is used by the iMarketing execution engine to personalize the relationship with the visitor.

## Campaign Management


Roll Up Campaign (Parent Campaign) "Envoy Laptop Campaign"
Execution Campaign(Sub Campaign) : iMarketing Advertisement "Envoy Laptop is a good deal!"
Execution Campaign (Sub Campaign) : iMarketing Advertisement "Envoy Laptop Includes Hot Technologies"
Execution Campaign (Sub Campaign) : iMarketing Promotion "15% on Envoy Standard Deluxe"

You can define campaigns using iMarketing. These campaigns are executed in iStore. The hierarchical structure of campaigns is described in the following paragraphs.

- **Roll-up Campaigns** is the highest level of campaign in the campaign hierarchy. The roll-up campaign is not executed. It contains the general and higher level information about the campaign. An example of a Roll-up campaign in the campaign hierarchy figure is the *Envoy Laptop campaign*.
- **Execution Campaigns:** You can define campaigns which promote or sell concepts, products, or devices to prospective audiences in the web store using iMarketing. In the campaign hierarchy figure, *Envoy Laptop is a good deal* is an execution campaign targeted to price sensitive visitors and *Envoy Laptop includes hot technologies* is an execution campaign targeted to technology savvy visitors. The iMarketing execution engine displays execution campaigns (advertisements or promotions) in iStore.

  For promotions, the merchant defines one or more offers applied during the purchase in iStore. For example:

  - Product discounts (% off, $ off)
  - Volume discounts

For more detail on Campaign Management and offers see the latest version of *Oracle Marketing Online Concepts and Procedures*.

## Personalization of the Visitor's Experience

By using a rule based technology, iMarketing personalizes the visitor's experience and recommends campaigns for the visitor. The merchant defines her own business rules to display in iStore the right campaign at the right moment to the right customer.

This rule based technology targets the right marketing message to the right visitor at the right time. For example, a campaign can be targeted to deliver a specific message to price sensitive visitors and another one to deliver a specific message to technology-savvy visitors.

As a visitor browses through a site, iMarketing dynamically displays the most appropriate campaign from all campaigns eligible for that customer segment.

This dynamic selection of campaign(s) is done on every web page that is generated, and is sensitive to the context of the customer's browsing session.

For example, the most appropriate time to present a promotion on golf balls is when a visitor is browsing at a golf bag or when a customer adds a golf bag to his/her shopping cart.

The merchant can also create rules which select products depending on items being viewed/in the shopping cart.

See Understanding Postings and Understanding Rule Sets for more details.

## Affiliate Management

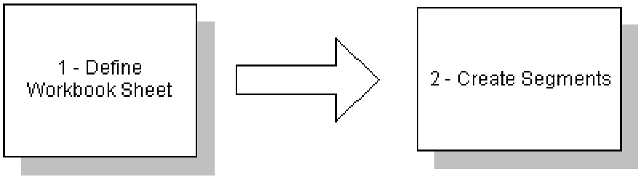
Oracle iMarketing enables merchants to register affiliates and post content on their site.

# Oracle iMarketing Process Flow

The main iMarketing functionality are the following:

- Defining an Audience/Segment
- Defining a Campaign
- Defining a Posting and Rule Sets
- Defining an Affiliate

## Defining an Audience/Segment

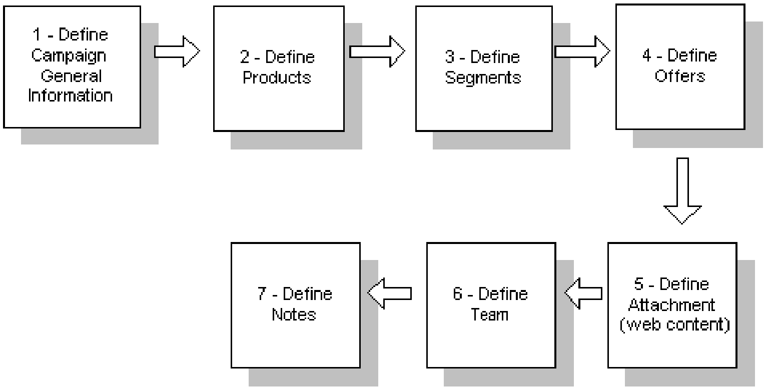


iMarketing provides segmentation functionality as a subset of Marketing Online's audience management functionality.

### Steps

1. Create a workbook sheet using Oracle Discoverer. A workbook sheet allows you to define a list of visitors. This list includes the people who are most likely to buy or be receptive to a campaign.
2. Create segments.

## Defining a Campaign



iMarketing provides campaign management functionality as a subset of Marketing Online’s campaign management functionality.

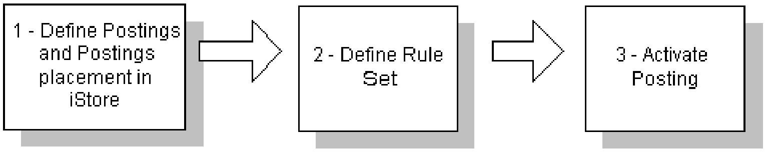
With iMarketing you can create campaigns of set-up type Internet Advertisement or Internet Promotions. These two types of campaigns can be displayed on iStore or on an Affiliate’s site by the iMarketing execution engine.

### Steps

1. Define **General** information such as the campaign name, description, source code, status (new, inactive, active...).
2. Define **Products** towards which the campaign is targeted. This information is used by the iMarketing execution engine. (See Understanding Rule Sets).
3. Define **Market** or Segments towards whom the campaign is targeted. This information is used by the iMarketing execution engine. (See Understanding Rule Sets).
4. Define **Offers**. An offer is an incentive such as a product or an order discount. This step exists only for iMarketing promotion campaigns.
5. Define **Attachments** i.e, the web content attached to the Campaign. The web content consists of images and texts. This web content is used by the iMarketing engine to display the campaign on the merchant’s web site.
6. Define the **Team** working on this campaign. For more information, see the latest version of *Oracle Marketing Online Concepts and Procedures.*

7. Define **Notes** for this Campaign. See the latest version of *Oracle Marketing Online Concepts and Procedures*.

### Defining a Posting and a Rule Set



Postings are personalized content place holders used in your store or displayed on an affiliate's web site. Postings display personalized campaigns on the web store.

#### Steps

1. Define a posting on your web store. You can define inactive postings and place these postings on iStore templates. See Posting API Details in the latest version of *Oracle iStore and iMarketing Implementation Guide* for more information. This step needs to be done only once. iStore has an out of the box posting location for the home page, shopping cart page, section page, and product detail page. See Understanding Postings for more information.
2. Define rule sets. See Understanding Rule Sets for more information.
3. Activate a posting by assigning a rule set and a style sheet to the posting and changing its status to active.

For more information on an affiliate posting, see Understanding Affiliate.

# How Oracle iMarketing Relates to Other CRM Products

iMarketing is integrated with Oracle Marketing Online and Oracle iStore. Merchants who buy both Oracle Marketing Online and iMarketing can define their campaigns from a single user interface.

## Integration with Oracle Marketing Online

Oracle iMarketing integrates tightly with Oracle Marketing Online to allow companies to have consistent marketing practices. Oracle Marketing Online and Oracle iMarketing use common functionality and user interface to create campaigns and segments. Customer segments defined in one application can also be used for campaigns of the other application.

Features shared with Marketing Online (campaign management) are integrated with other CRM and ERP modules such as Pricing Engine and Inventory. Refer to the latest versions of *Oracle Marketing Online Concepts and Procedures* and *Oracle Marketing Online Implementation Guide* for more details.

Oracle iMarketing uses the following information to evaluate segments:

- Customer profile
- Information about the organization the visitor belongs to (in a business-to-business interaction)

## Integration with Oracle iStore

Oracle iMarketing integrates with Oracle iStore to deliver personalized content to the iStore user interface and to provide merchandising functionality. These applications are integrated through Java/JSP API calls and have to be installed on the same application server.

The main integration points are:

- Oracle iMarketing postings are invoked on iStore Java Server Pages. While generating a page, the JSP engine makes a posting call to iMarketing. The visitor profile is used to display the personalized content in the visitor's browser.
- iStore provides contextual information to iMarketing. The information includes: the products being viewed, products in the shopping cart, specialty store, and the customer identifier. As a result, iMarketing generates the HTML for displaying the posting.

# Understanding Segmentation

iMarketing leverages the segmentation functionality of Oracle Marketing Online to enable iMarketing users to define and manage segments. A segment is a group of related visitors. For example, visitors who live in California can belong to the "CA Segment." Oracle iMarketing campaigns can be associated with the segments defined by the merchants. Each campaign can be associated with one or more segments.

Merchants define segments in Oracle Discoverer. Visitors belonging to those segments become eligible for campaigns targeted towards those segments.

Segment structure is hierarchical, and a merchant can create sub-segments within a particular segment.

Each campaign can be associated with one or more segments. Campaigns are then targeted only at segments that they are associated with.

Once logged in, the designated segments apply to the visitor throughout the session.
From eligible campaigns that are available for the customer segment, a campaign is selected for display to the customer depending on the session context i.e, the page that the customer is currently browsing.

The segment condition applies throughout the session, although different campaigns might be selected, depending on the session context.

---

**Note:** You can create an Anonymous Visitor segment to target campaigns to visitors not registered in iStore. All anonymous visitors share the same PARTY ID. The name attribute of this party is IBEGUEST.

---

## Guidelines on Creating a Workbook in Oracle Discoverer

Oracle Discoverer is a business intelligence tool that allows you to retrieve and analyze data without having to understand how to access databases.

### Relationship Between Segments and Workbooks

While creating a segment, you must associate a Discoverer workbook sheet with the segment

You can use a Discoverer workbook sheet to define a list of customers (PARTIES of type PERSON).

### Segmentation Process

A batch program (Refresh Party Market Segments) managed by the concurrent manager populates the association between PARTIES (of type PEOPLE) and Segments. The PARTIES - SEGMENTS association (segmentation) is used by Oracle iMarketing during the personalization process to retrieve the list of segments that a visitor belongs to. Based on the list of segments associated with a particular visitor, the list of campaigns that target this specific visitor is dynamically retrieved.

### Workbook Creation

An Oracle iMarketing user who creates a segment must also create a workbook sheet using Oracle Discoverer. For more details, refer to the latest version of *Discoverer 3i User Guide*.

1. To launch Oracle Discoverer, click the **Web Discoverer** link found in the Create Segment screen, or Segment Details screen.
2. To create a workbook sheet, use the Oracle Marketing business area. This business area contains folders needed to complete workbook creation. The workbook sheets are used later for the segmentation process. While creating a workbook, ensure that:
   - The Table Display is used to select data
   - At least one item called PARTY ID is included in the list of selected items
   - The workbook is saved in the database

### Business Area Tips and Hints

Oracle Marketing Online provides a Discoverer business area. Using the Oracle Marketing Online business area with associated folders, a marketer can choose from a wide range of selection criteria based on

- business
- business history
- customer demographic
- previous responses to solicitations

### Business to Consumer Segmentation Guidelines

- Select the PARTY ID item from the folder containing PARTY of type PERSON.
- Apply selection on the party’s profile (customer’s profile) to generate the query.

### Business to Business Segmentation Guidelines

- Select the PARTY ID item from the folder containing PARTY of type PERSON.
- If needed, select any column from the folder related to the ORGANIZATION (for the selected PARTIES of type organization). In this case, the PARTY of type PERSON is an employee of the Organization.
- You can apply selection criteria on the customer’s profile and on the organization’s profile to generate this query.

## Understanding Campaigns

A campaign consists of a set of messages delivered during a defined period to an audience. A campaign in iMarketing can be displayed on a merchant’s own site or on an affiliate site.

Some campaigns may have target products associated with it, while others might not have target products associated with it, e.g, a brand awareness campaign. Target products can be assigned to any kind of campaigns.

Campaigns can be:

- **Advertisements**: campaigns that present content to customers, but do not have an offer associated with it.
- **Promotions**: campaigns in which offers can be attached. An offer might be:
  - a discount on products
  - a discount on the total order
  - a free product Y when you buy two of product X

When you create an iMarketing campaign the following information is considered by the iMarketing execution engine.

1. **General Information About a Campaign** such as the campaign name, description, source code, status (new, inactive, active...). The iMarketing execution engine uses the start and end date and the status of the campaign for processing. Only active campaigns with a valid start and end date are displayed in iMarketing.

2. **Products** towards which the campaign is targeted. This information is used by the iMarketing execution engine.
3. **Segments** with which the campaign is associated. This information is used by the iMarketing execution engine (see Understanding Rule Sets).
4. **Attachments** define web content attached to the campaign (images and texts). See the latest version of *Oracle Marketing Online Concepts and Procedures* for details.

## Understanding Advertisements

An advertisement is a campaign type without an associated offer. For example, a brand awareness campaign or a product advertisement. Advertisements may or may not be linked to products.

An Advertisement appears on local and affiliate sites. An advertisement can link to

- various site locations
- a product page

An advertisement can also perform actions like

- add an item to the shopping cart
- display the cart

## Understanding Promotions

A promotion is a specific type of campaign that offers a product discount. It impacts the price of an item purchased or the order amount during the checkout process. A promotion is associated with one or more offers.

You may want to promote offers that have the same marketing message. For doing this, you may create a promotion of an internet campaign type with multiple offers attached to it. If you want to promote multiple offers that have different marketing messages, then you should create one internet promotion campaign per offer. For example:

- create one promotion campaign: *5% on Standard Envoy Laptop* and define under this campaign one offer: *5% discount on the Standard Envoy Laptop.*
- define one promotion campaign: *Great discount on all Envoy Laptops* and define under this campaign two offers: *a 10% discount on the Standard Envoy Laptop and a 5% discount on the Deluxe Envoy Laptop.*

Offer types include:

- **Product Discount**: linked with a target product to reduce product price
- **Add Free Product:** linked with a target product to add one or more free products to the order
- **Order Discount:** reduces the order's total cost

## Understanding Content

Content, consisting of images and text can be created for campaigns from iMarketing's Create Content screen. Content rendering is the way in which content is presented on a specific location of the merchant's or on an affiliate's web site.

Content rendering processing is done by the Oracle XML parser which applies a XSL style sheet to the XML structure returned by iMarketing.

> **Note:** XSL is an eXtended StyLe style sheet that is defined at the posting level. This file is stored on the local file system of the application.

You can associate content (text, image) to a campaign from the Create Content screen. Enter data in the following fields:

- **File Name (Image Only)** is the name of the file containing the content. All content, including textual content is contained in files.
- **Location (Image Only)** is the Location of the file. If this field is left blank, Location points to the/OA_MEDIA/ virtual directory. If the user enters a path, the file is retrieved from the web server virtual directory specified in the path. If the user enters a URL, content is retrieved from this URL.
- **Type** is the type of content. Oracle iMarketing supports text and image content. The content type string gives details on the type of content (text, image).
- **Sub Type** includes semantics associated with content. For **text**, use the following:
  - **Short Description**: of the campaign, 40 characters or less.The user decides the length, no control is imposed by Oracle iMarketing.

- **Description:** of the campaign, 120 characters or less. The user decides length, no control is imposed by Oracle iMarketing.
- **Long Description:** of the campaign. The user decides length, control is imposed by Oracle iMarketing.
- **For Images,** use the types specified below. Banner names and sizes come from the IAB/CASIE Advertising banner size standard. Refer to http://www.iab.net for IAB/CASIE Advertising banner size standard.

  The following table gives the image details for content.

*Image Details*

| Content Type | Description | Size |
|---|---|---|
| Image | Full Banner | Banner of 468 by 60 pixels. |
| Image | Half Banner | Banner of 234 by 60 pixels. |
| Image | Vertical Banner | Banner of 120 by 240 pixels. |
| Image | Button1 | Banner of 120 by 90 pixels. |
| Image | Button2 | Banner of 120 by 60 pixels. |
| Image | Square Button | Banner of 125 by 125 pixels. |
| Image | Micro Button | Banner of 88 by 31 pixels. |

- **Width and Height** (Image Only) For image content, width and height are in pixels.
- **Language** The content is associated to a specific language.
- The preview button enables the user to preview content.

### How Content is Retrieved and Displayed in a Style Sheet

When a posting is executed, the posting selects the correct content by selecting the appropriate content type defined for the user language.

**Note:** A style sheet can include processing to pick another content if the required content is not defined for the selected campaign.

### Organizing Style Sheets and Content Associated With My Campaigns

Oracle iMarketing does not impose any control over the list of content defined for a campaign. A minimal list of content type should be defined for each campaign to enable multiple style sheets to display the campaign in multiple postings. Examples of content type include:

- Text - Short Description
- Image - Full Banner
- Image - Square Button

The style sheet must be designed to handle cases where a specific content type is not defined for the selected campaign.

## Understanding Campaign Selection

Selecting and presenting campaigns and content that is relevant to a customer segment is important for attracting and retaining customers and for stimulating purchases. As a customer browses through the different screens on a site, iMarketing dynamically selects the campaign to be displayed from all campaigns eligible for the customer's market segment.

The following selection criteria can be used:

- **Specific list of campaigns**: campaigns are picked randomly in the defined list.
- **Campaigns relevant to the shopping context**: iMarketing retrieves campaigns matching the visitor's profile (segments) and the shopping context (products being viewed or products in the shopping cart).

## Guidelines on Selecting Campaigns Relevant to the Shopping Context

The following table describes the various campaign types relevant to the shopping context.

***Campaigns Relevant to the Shopping Context***

| Selected Campaign Type | Selected Shopping Context | How Campaigns Relate to products | String Displayed in the Rule Set detail Page | Description |
|---|---|---|---|---|
| Advertisements | Products being viewed | Promoting | Advertisements promoting products being viewed. | Select advertisements related to the product(s) being viewed and targeted to any of the user's segments. |
| Advertisements | Products being viewed | Cross-selling | Advertisements cross-selling products being viewed. | Select advertisements cross-selling the products being viewed and targeted to the user's segments. |
| Advertisements | Products being viewed | Up-selling | Advertisements up-selling products being viewed. | Select advertisements up selling the products being viewed and targeted to any of the user's segments. |
| Advertisements | Products in the shopping cart | Promoting | Advertisements promoting products in the shopping cart. | Select advertisements promoting the products in the shopping cart and targeted to any of the user's segments. |
| Advertisements | Products in the shopping cart | Cross Selling | Advertisements cross-selling products in the shopping cart. | Select advertisements cross-selling the products in the shopping cart and targeted to any of the user's segments. |

***Campaigns Relevant to the Shopping Context***

| Selected Campaign Type | Selected Shopping Context | How Campaigns Relate to products | String Displayed in the Rule Set detail Page | Description |
|---|---|---|---|---|
| Advertisements | Products in the shopping cart | Up-selling | Advertisements up-selling products in the shopping cart. | Select advertisements up-selling the products in the shopping cart and targeted to any of the user's segments. |
| Advertisements | Any product- all eligible campaigns | N/A | Any eligible advertisement | Selects advertisements targeted to any of the user's segments. |
| Same for Promotions. In Campaign Type, select only Promotions.<br>Advertisements and Promotions In campaign type, Select Advertisements and Promotions. | | | | |

## Understanding Postings

A posting is a personalized placeholder for content. The page on which the posting resides is called the calling page. The posting can reside on your site or on an affiliate site. A posting serves as a container through which a campaign is displayed. The content inside the posting is selected by a rule set.

A posting can be called from any Java Server Page (JSP) template, including an iStore template.

The rule set selected for a posting is responsible for selecting a personalized campaign for display.

Content rendering is the way content (text, images) is rendered on a specific location of the merchant or an affiliate web site for a campaign.

Content rendering processing is done by the Oracle XML parser that applies a XSL style sheet to the XML structure returned by Oracle iMarketing.

The Posting API returns HTML. The posting style sheet is applied to the XML structure to generate this HTML. This is the actual content returned for display to the user.

A posting returns nothing if the posting status is inactive.

**Note:** For information on content, see "Understanding Content."

## Understanding Rule Sets

A rule set consists of multiple rules that are used by the iMarketing execution engine to select a campaign.

A rule can have one or more conditions but only one selection criteria. If a rule has multiple conditions, then each condition must be evaluated to **true** for the selection criteria to be executed.

The rule set is dynamically evaluated when the page is generated for display. A rule set is associated with a posting which can be called from any Java Server Page (JSP) template, including an iStore template.

A rule consists of

- an execution order
- a condition(s)

- a selection criteria

A rule can have one or more conditions but only one selection criteria. If a rule has multiple conditions, then each condition must be evaluated to **true** for the selection criteria to be executed.

## Example

For a given customer, the following conditions and selection criteria apply:

***Example of condition and selection criteria***

| Condition | Selection Criteria |
|---|---|
| Visitor belongs to the teenage segment | Access a predefined list of campaigns. |
| Visitor is registered | Select advertisements promoting products in the shopping context |

Oracle iMarketing supports four types of conditions:

- **visitor type**: **anonymous, registered with order history,** or **registered without order history.**
- **specialty store**: store(s) that the visitor is currently browsing (condition is **true** if the visitor is in any specialty store specified at the condition level)
- **customer segment**: segment(s) that the visitor belongs to (condition is **true** if the visitor belongs to any segment specified at the condition level)
- **empty shopping cart**: this condition can be used to check if the shopping cart is empty.

Two types of selection criteria are available:

- **Specific list of campaigns:** A list of campaigns defined by the merchant
- **Campaigns relevant to shopping context:** Returns a set of campaigns based on user’s shopping context (products being viewed, products in the shopping cart, and visitor’s profile)

### How Rule Sets are Evaluated by Oracle iMarketing

1. When a posting is called, the number of campaigns to be retrieved (N) is passed to the rule set for processing.

2. The rule having the highest execution order is executed first.
3. If all conditions are true for the first rule, then the selection criteria is evaluated and x campaigns are returned.
4. If x is less than N (x<N), then the next rule is evaluated. N minus x (N-x) campaigns remain to be retrieved.
5. When x=N, then stop rule processing.
6. If the condition is false for rule one, then the next rule is evaluated.

### Example: Targeting Advertisements on the Home page

The merchant wants to display an advertisement for a seminar targeted towards the Gold Account segment.

If the person browsing the site is anonymous, then a specific advertisement, selected from the Specific List of Campaigns is displayed. In this case, the campaigns displayed do not depend on the browsing context.

In this case, the **Specific List of Campaigns** selection criteria must be used.

The rule set is:

- **Rule 1:** If a visitor belongs to Gold Account segment, then display an iMarketing Gold advertisement.
- **Rule 2:** If a visitor is anonymous, then display an iMarketing Bronze advertisement.

### Example: Targeting Promotions Depending on the Browsing Context

The merchant wants to display an internet offer campaign related to the product being viewed and the applicable user segment.

If no user segment is available, the merchant wants to display an internet promotion campaign targeted to the customer segment. Offers returned first are campaigns with the highest priority. If no promotion is available, then display Default Campaign for Anonymous Customers.

The rule set is:

- **Rule 1**: If a visitor is registered, then display **promotions promoting products being viewed.**
- **Rule 2:** If a visitor is registered, then display campaigns returned by **any promotions** selection criteria

- **Rule 3**: If a visitor is anonymous, then display **Default Campaign for Anonymous Customers.**

The merchant will display a posting showing an internet promotion campaign appropriate to the product being viewed. This is so because the posting provides more information than only the value of the discount that is displayed in iStore.

The posting may display a marketing message urging the customer to buy the product immediately. For example, *a great promotion on a printer. Valid through the end of the month. Buy now! Get a free scanner if you purchase this printer.*

## Understanding Affiliates

An affiliate is a third party site that is used to direct internet traffic to the merchant's web store.

> **Note:** An affiliate's user is considered an anonymous user by the iMarketing execution engine.

### Posting Campaigns on Affiliate Sites

- registering and managing affiliates and postings
- selecting campaigns for display
- dynamically distributing up-to-date content

#### Steps To Be Done By The Merchant During Definition Time:

1. Register an affiliate in iMarketing.
2. Create and associate a posting to an affiliate.
3. If a posting is targeted to an affiliate, then HTML content is generated.
4. The HTML content that is generated is sent to the affiliate. The affiliate includes this content on his/her web pages.

#### Steps To Be Done By The iMarketing Engine During Execution Time:

1. The affiliate displays the page that includes the HTML content generated by iMarketing.

2. This HTML content dynamically calls the iMarketing execution engine which returns the appropriate campaign. This dynamic call uses javascript (if the user’s browser supports javascript), otherwise a default banner is displayed by the affiliate’s site.

# Style Sheets for Campaigns and Products

iMarketing uses style sheets to define what type of content (image, text etc.) will be displayed and how it will be displayed. The style sheet specification is used to describe a style sheet which processes the XML content delivered by iMarketing and generates HTML for rendering.

Style sheets are defined at the posting level.

## Style Sheets for Campaigns

There are three style sheets for Campaigns. These style sheets select the Attachment Subtypes for rendering. These style sheets are described below:

1. ibacpns1.xsl: This style sheet selects one of the following attachment subtypes:
   - IMG_FULL_BANNER
   - IMG_HLF_BANNER
   - TXT_DESC
2. ibacpns2.xsl: This style sheet selects one of the following attachment subtypes:
   - IMG_MICRO_BUTTON
   - TXT_DESC
   - TXT_LONG_DESC
3. ibacpnm3.xsl: This style sheet selects one of the following attachment subtypes. This stylesheet can be used for multiple Campaign selection.
   - IMG_MICRO_BUTTON
   - TXT_DESC.

The attachment sub types starting with IMG refers to images. If there is a Campaign with an attachment subtype not listed above (for example, **IMG_BUTTON1**), you can change the template rule that matches Campaign and looks for an Attachment subtypes of **IMG_BUTTON1**.

You have to have a style sheet for-each clause for the new Attachment subtype. Close the <xsl:for-each ...> element with a corresponding </xsl:for-each> element. The following example shows a sample of the style sheet describing the above functionality.

```
<xsl:for-each select="AttachmentLst/Attachment[starts-
   with(AttachmentSubType, 'IMG_BUTTON1)]">
     <tr>
       <td>
         <xsl:call-template name="IMAGE"/>
       </td>
     </tr>
</xsl:for-each>
```

A template named **IMAGE** is supposed to be present in the style sheet file. The above procedure can be followed for all other attachment sub-types for image like **IMG_BUTTON2, IMG_SQUARE_BUTTON, IMG_VERTICAL_BANNER** etc.

## Style Sheets for Products

There are four Product related style sheets. They are as follows.

1. ibaprod1.xsl: This stylesheet shows ProductDetailLink and ShortDescription.
2. ibaprod2.xsl: This stylesheet shows ThumbNail image, ShortDescription, and ProductDetailLink.
3. ibaprod3.xsl: This stylesheet shows ThumbNail image, ShortDescription, Description, and ShoppingCartLink.
4. ibaprod4.xsl: This stylesheet shows Small image, ShortDescription, Description, and ShoppingCart Link.

## Product Properties in Style Sheets

The properties of a Product object accessible from the style sheets are:

- Short Description
- Description
- Product Detail Link
- Shopping Cart Link
- ThumbNail
- Small Image
- Large Image

Usually, a product has a Short Description and a Description. It also has a ProductDetailLink.

A product may also have a ShoppingCart Link if prices are defined. From the ShoppingCart Link you can go to the Shopping Cart page. To define prices for Products, See the latest version of *Oracle iStore Concepts and Procedures.*

For images, a product should have display contexts with titles like **THUMBNAIL**, **STORE_PRODUCT_SMALL_IMAGE,** and **STORE_PRODUCT_LARGE_ IMAGE**.

These display contexts should be set up in such a way that a proper image of appropriate type specified by the above titles, are selected at run time. If display context is not specified, a default image will be selected.

In the existing style sheets, where some child attribute are accessed from within a product rule by entering **child::**, you can replace it by another child attribute. e.g, **the child::ThumbNail** reference in ibaprod2.xsl and ibaprod3.xsl can be replaced by **child::SmallImage** or **child::LargeImage**.

You should not change **child::ShortDescription** by **child::ThumbNail** because this will display an image where text is being displayed.

You can change **child::ShortDescription** by **child::Description** because the changes are of similar type. The following example shows you how to display a SmallImage of a Product.

```
                        <xsl:element name="a">
    <xsl:element name="img">
      <xsl:attribute name="src">
        <xsl:value-of select="child::SmallImage"/>
      </xsl:attribute>
      <xsl:attribute name="alt">
        <xsl:text></xsl:text>
      </xsl:attribute>
      <xsl:attribute name="border">
        <xsl:text>0</xsl:text>
      </xsl:attribute>
    </xsl:element>
  </xsl:element>
```

# Using Oracle iMarketing

This topic group provides process-oriented, task-based procedures for using the application to perform essential business tasks.

This chapter covers overall Navigator and screen functionality, and explains how to use the fields and other elements within screens.

## Managing Segments

You can view segment details on Segment Details screen and create segments on Create Segment screen. This functionality is available both on Oracle Marketing Online and Oracle iMarketing user interfaces.

For Creating and Updating Segments, see the latest version of *Oracle Marketing Online Concepts and Procedures.*

## Managing Campaigns

For more information about Creating and Updating Campaigns, see the latest version of *Oracle Marketing Online Concepts and Procedures.*

## Creating a Posting

Use this procedure to create a posting from the Create Postings screen.

### Prerequisites

The user must have user permissions.

### Steps

1. In the Posting tab, choose **Posting.** The Postings screen appears listing the details defined by the merchant.
2. Click **Create**. The Create Posting screen appears.
3. Enter the posting **Name**.
4. Enter a **Description** for the posting.

5. In **Posting On** list:

   a. Select **Store** to post on the merchant store. **Or**

   b. Select **Affiliate Site** to post on an affiliate site.(See Selecting an Affiliate Site).

   > **Note:** When a posting is created for an affiliate, an additional Affiliate Content section also appears on the Posting Detail screen. This section includes the HTML content that has to be placed on the affiliate's site.

6. Assign the **Status** as Active or Inactive. The default status is Inactive.

   > **Note:** For campaigns to appear in postings during runtime, the campaign status should be "Active." For more information on campaigns, see the latest version of *Oracle Marketing Online Concepts and Procedures.*

7. In the **Content Detail** section, enter or search for the rule set to be used. To select a rule set, click **Go**. Rule Sets screen appears.

8. Select a Rule Set by clicking the desired rule set in the **Name** column. Create Posting screen appears with the selected rule set appearing on the Rule Set field.

9. Enter the XSL style sheet name in the **Rendering Style** field to assemble content for this posting.

   > **Note:** XSL is an eXtended StyLe sheet defined at the posting level.
   >
   > The rule set and style sheet are only mandatory if the posting status is set to active.

10. Click **Create**. Your new posting is saved.

# Viewing or Editing a Posting

### Prerequisites

The user must have user permissions.

### Steps

1. In the Posting tab, choose **Posting.** The Postings screen appears listing the details defined by the merchant.
2. To review details for a posting or to edit information, click the hyperlink for the posting **Name**. The posting details screen appears displaying information for the selected posting.

# Selecting an Affiliate Site

Use this procedure to select an affiliate site on the Create Posting screen.

### Prerequisites

Complete the initial steps for creating a posting.

### Steps

1. In the **Posting On** field, select **Affiliate Site** to post on an affiliate site. An additional field appears.
2. Click **Go**. Affiliate screen appears with a list of available affiliates.
3. Click an affiliate name that you want to select. Create Posting screen appears with selected affiliate name added in the additional field.

# Managing Rule Sets and Rules

A rule set is a package of rules that define how to select content to be displayed in a campaign.

# Creating Rule Sets

Use this procedure to create rule sets from the Create Rule Set screen.

### Prerequisites

None

### Steps

1. In the Posting tab, choose **Rule Set.**

   Rule Sets screen appears listing existing rule sets.

2. Click **Create.**

   The Create Rule Set screen appears.

3. Enter a **Rule Set Name** and **Description** for the rule set.

   In the description field, the merchant can enter the rule set meaning and guidelines on how to use it.
   Examples of what the merchant's web site manager might add, include:

   * the meaning of the rule set, what the rule set is expected to return
   * where and when this rule set should be used
   * what does the merchant imply by shopping context; is it products being viewed or products in the shopping cart

4. In the Content section, select either campaign or product button. ( (See Selecting Content).

5. Enter a number in the Maximum Number Displayed field. This number determines the maximum number of objects (campaigns or products) that will be displayed.

**Note:** the number of objects that you enter in the Maximum Number displayed field on Create Rule Set screen will take precedence over the number entered in the JSPs.

6. Click **Create.** Rule Sets screen appears listing the new rule set.

# Creating Rules

Use this procedure to create a rule on the Create Rule Set screen.

For creating a rule, the merchant defines the conditions and selection criteria (or "action" upon which the rule is executed, if conditions are true).

## Prerequisites

A Rule Set has been created.

## Steps

1. In the Posting tab, choose **Rule Set.** Rule Sets screen appears listing the existing rule sets.
2. To create another rule for an existing rule set, click the **Name** hyperlink. Rule Set Detail screen appears. Click **Add New Rule.** To create a rule for a new rule set, click **Create**. The Create Rule Set screen appears. Enter a **Rule Set Name** and **Description.**
3. Click **Add New Rule**. Create Rule screen appears.
4. Enter a **Rule Name**, **Execution Order**, and **Description** for the rule. Click **Add Condition**. Rule Detail screen appears.

**Note:** The Execution Order for a rule has to be 1 or greater than 1.

5. In the Available Conditions section, select a condition type from the following list. (See Selecting a Condition).
   - condition regarding customer segment
   - condition for specialty store
   - condition regarding visitor type
   - condition for empty shopping cart
6. Depending on the Condition Type selected, make adequate selections to complete the process of selecting a condition. Click **Done**.

**Note:** You can add more than one condition for a rule.

7. Rule Detail screen appears. (Only if you selected campaign as the content type). Click **Done**. Create Rule screen appears.
8. Click **Continue**. Rule Detail screen appears if you selected Campaign as the content type. Product Selection screen appears if you selected product as the campaign type. (See Selecting Content).
9. Make adequate selections depending on the content type selected.
10. Rule Detail screen appears showing the selection criteria.
11. Click **Done**. Rule Set Detail screen appears with the Rule details.

## Selecting A Condition

A condition(s) for a rule can be selected from the Available Condition section of the Rule Details screen. Select a condition from the following list:

- condition regarding customer segment
- condition for specialty store
- condition regarding visitor type
- condition for empty shopping cart

**Note:** A Rule can have multiple conditions. Conditions are combined with an AND operator.

## Selecting a Condition Regarding Customer Segment

Use this procedure to select a condition regarding customer segment from the Rule Detail screen.

### Prerequisites

Complete the initial steps of creating a rule on the Create Rule screen.

### Steps

1. In the Available Condition Section on the Create Rule screen, select a **condition regarding customer segment.**
2. Customer Segment screen appears. Click **Add**.

3. Add Customer Segment screen appears. Select a segment to be included from the list available by checking the adjoining checkbox in the Add column. Click **Add**.
4. Customer Segment screen appears. Click **Done**. Rule Details screen appears listing your selected condition(s) in the Available Conditions section.

## Selecting a Condition for a Specialty Store

Use this procedure to select a condition for a specialty store from the Rule Detail screen.

### Prerequisites

Complete the initial steps of creating a rule on the Create Rule screen.

### Steps

1. In the Available Condition Section on the Create Rule screen, select a **condition for a specialty store**.
2. Specialty Store screen appears. Click **Add**.Specialty Store screen appears listing the stores available.
3. Check the box associated with the specialty store to be added in the **Add** column. Click **Add**.
4. Specialty Store screen appears with the selected store. Click **Done**. Rule Details screen appears listing your selected condition(s) in the Available Conditions section.

## Selecting a Condition Regarding the Visitor Type

Use this procedure to select a condition regarding the visitor type from the Rule Detail screen.

### Prerequisites

Complete the initial steps of creating a rule on the Create Rule screen.

### Steps

1. In the Available Condition section on the Create Rule screen, select the **condition regarding the visitor type**.

2. Visitor Type screen appears. Select either **Registered with order history, Registered no order history,** or **Anonymous** radio button.
3. Click **Done**. Rule Details screen appears listing your selected condition(s) in the Available Conditions section.

## Selecting Content

Content for a rule can be selected from the Content section of the Rule Detail screen. Select content type of either product or campaign.

For a campaign content type, select from the following two types of campaigns.

- specific list of campaigns
- campaigns relevant to the shopping context

## Selecting Specific List of Campaigns

Use this procedure to choose a selection criteria for selecting a specific list of campaigns.

### Prerequisites

Complete the initial steps of creating a rule set.

### Steps

1. In the Content section of the Create Rule Set screen, select the Campaign button.
2. Enter a positive integer in the Maximum number displayed field. Click **Add New Rule.**
3. Complete the steps for creating a rule.
4. In the **Content Selection** section of the Rule Details screen, select **Specific List of Campaigns.** Campaigns screen appears listing all the campaigns.
5. Select a campaign and click **Add**.
6. Campaigns screen listing the selected campaign appears. You can either add another campaign or click **Done**.

**Note:** The merchant chooses

- one specific campaign to display, or
- a list of campaigns (to enable a kind of campaign rotation).

The merchant manages the list and controls its consistency and coherency.

The algorithm used to get campaigns from a "specific list of campaigns" consists of a list of N campaigns:

The rule engine requests X campaigns from the specific list of campaigns selection criteria.

- If x< N, selection criteria returns X campaigns picked randomly from the list of N campaigns.
- If x ≥N, selection criteria returns X campaigns.

A control is enforced.The returned campaigns are active and the current date is between the campaign start and end date.

## Selecting Campaigns Relevant to Shopping Context

Use this procedure to choose a selection criteria for selecting campaigns relevant to the shopping context.

### Prerequisites

Complete the initial steps of creating a rule set.

### Steps

1. In the Content section of the Create Rule Set screen, select the Campaign button.
2. Enter a positive integer in the Maximum number displayed field. Click **Add New Rule.**
3. In the **Content Selection** section of the Rule Details screen, select **Campaigns Relevant to Shopping Context.** (See Guidelines on Selecting Campaigns Relevant to the Shopping Context).
4. Select a campaign type-Promotions, Advertisements, or both.
5. Select the product context from the **Product Context** drop-down list.
6. Select a relationship from the **Relationship** drop-down list and click **Done**.
7. Rule Details screen appears with the details selected.

**Note:** Cross-sell and up-sell are available from every page but the home page in iMarketing.

## Selecting Product Content Type

Use this procedure to select the Product content type from the Create Rule Set screen.

### Prerequisites

Complete the initial steps for creating a rule set.

### Steps

1. In the Content section, select the **Product** button.
2. Enter a positive integer in the Maximum number displayed field.
3. Click **Add New Rule** button. Complete the steps for creating a rule. Product Selection screen appears.
4. Select a Product Context from the list.
5. Select a Relationship from the list. Click **Done**.
6. Rule Detail screen appears with the details selected. Click **Done**.

# Managing Affiliates

Affiliates are third parties (subsidiaries, suppliers, ad networks, etc.) that participate in a merchant campaign. Use this procedure to update information about your affiliates and to add new affiliates.

### Delivering Postings to Affiliates

For postings to be delivered to affiliates users, some HTML content is generated and displayed in the posting screen. For this to occur, HTML content must be placed on the affiliate site to call a posting. A merchant can email this HTML to the affiliate or display HTML content in a web self service application.

When executed on a browser by affiliate end-users, this HTML code returns the corresponding HTML content and/or images to be displayed by the browser. The page is assembled by the user’s browser. It mixes content from the affiliate site and content from the merchant site through a call to Oracle iMarketing.

This delivery mechanism is useful in the following ways:

- requires no development effort at the affiliate site
- can be used even within static web pages
- easy to use with HTML authoring tools

# Creating an Affiliate

Use this procedure to set up an affiliate site on which you will display your campaign.

### Prerequisites

The user must have user permissions.

### Steps

1. Click the Affiliate tab. Affiliate screen appears.
2. Click **Create**. Create Affiliate screen appears.
3. Enter an Affiliate name in the Affiliate field and address information in the company information section.
4. Click **Create**.

# Viewing or Editing an Affiliate

Use this procedure to view or edit an affiliate on the Affiliates screen. Affiliates cannot be deleted.

### Prerequisites

The user must have user permissions.

### Steps

1. In the Affiliate tab, choose **Affiliate.** Affiliate screen appears displaying a list of current affiliates.

2. To review details for an affiliate or to edit information, click the hyperlink **Affiliate Name**. The Affiliate Details screen appears and shows address information for the selected affiliate.

3. To edit information, make the requisite changes in the required field(s) and click **Update**. Affiliate screen appears.

## Creating Content (Attachments)

You can associate content (text, image) to a campaign in the Create Content screen. Use this procedure to create content from the Create Content screen.

### Prerequisites

- Account in Oracle Marketing Online.
- User permissions in iMarketing.

### Steps

1. Click the Campaign tab. Campaigns screen appears.

2. Click the name of the campaign for which you want to view the details. Campaign Details screen appears with the details.

3. Click **Attachments** quick links on the left. Campaign Details-Attachment screen appears.

4. Click **Create**. Create Content screen appears.

5. Select a type of campaign from the Type list. Depending on this selection, the screen is refreshed with other fields.

   a. If you select Image Type:

   - Select a Sub Type from the drop-down list.
   - Enter a File Name.
   - Enter Location.
   - Enter Image Width, Image Height, and/or Alternate Text.
   - Select a Language.
   - Enter a Link to URL. (See Examples of URL Links).

   b. If you select Text Type

- Select a Sub Type from the drop-down list.
- Enter Text.
- Select a Language
- Enter a Link to URL.(See Examples of URL Links)

**Note:** For more information about navigating to the iStore screens, see *Oracle iStore and iMarketing Implementation Guide*.

6. Click **Create**. Campaign Details-Attachment screen appears with the new content type appearing in the table.

## Examples of URL Links

You can enter four different values in **Attachment linkToURL** field. The following list gives examples of URL links that can be entered in the **Attachment linkToURL** field while defining an Attachment in a Campaign.

- Information about Item Detail Page
- Information about Item Shopping Cart Page
- Information about Home Page
- Any other URL

For information about Item Detail Page, enter **STORE_CTLG_ITEM_ROUTE?item=<itemid>&section=<sectionid>.**

- **STORE_CTLG_BUY_ROUTE** is case-sensitive
- Item Id is an integer like 2035 that denotes the id of the item in the database
- SectionId is not mandatory. It is an integer like 345

For information about Item Shopping Cart Page, enter **STORE_CTLG_BUY_ROUTE?item=<itemid>.**

Section id is not required to go to a shopping Cart page.

For information about Home page, enter **STORE_HOME**.

For any other URL, include all other URLs, except the ones listed above. For example, http://www.oracle.com

The following table includes details about the attachment URL

***Details about the Attachment URL***

| Link To | What You Should Enter |
|---|---|
| Item Detail Page | STORE_CTLG_ITEM_ROUTE?<br>item=<itemid>&section=<sectionid><br>* section is optional |
| Shopping Cart Page | STORE_CTLG_BUY_ROUTE?<br>item=<itemid> |
| Home Page | STORE_HOME |
| Any other URL | Any other URL |

# Viewing or Editing an Affiliate's Posting Preferences

## Default Posting

In the default posting section, specify the posting used by default by the iStore template.

## Affiliate Server URL

This URL is used in the HTML code of the affiliate posting page to display dynamic customized posting on the affiliate page.

## Prerequisites

The user must have user permissions.

## Steps

1. In the Administration tab, choose **Posting**. The current server URL appears in the **Server URL** field. It has the server name, a port number, and a path.
2. To change the URL, type the new URL in the **Server URL** field and click **Update**. Setup complete message appears when the new URL is saved successfully.

**Note:** If you see an error message, repeat step 2.

# Style Sheets for Campaigns and Products

iMarketing uses style sheets to define what type of content (image, text etc.) will be displayed and how it will be displayed. The style sheet specification is used to describe a style sheet which processes the XML content delivered by iMarketing and generates HTML for rendering.

The following style sheets are out-of-the-box style sheets provided with Oracle iMarketing. You can also customize these style sheets.

## Style Sheets for Campaigns

### ibacpns1.xsl

```
<?xml version="1.0"?>
<!--$Header: ibasinglecpnpstng1.xsl 115.3 2000/06/21 23:13:39 asaha ship $-->
<!--This stylesheet is for rendering Camapign by its Full Banner image
-->
<!--If a Full Banner-->
<!--image is not available, Half Banner image is tried for. Otherwise, a default
image is -->
<!--shown alongwith short description. The attachment types expected by this
stylesheet are-->
<!--IMG_FULL_BANNER, IMG_HALF_BANNER and TXT_SHORT_DESC-->
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
<xsl:template match="/">
  <table border="0" cellpadding="1" cellspacing="2">
    <xsl:apply-templates select="//Campaign"/>
  </table>
</xsl:template>
<xsl:template match="Campaign">
  <tr>
    <td>
      <table border="0" cellpadding="2" cellspacing="0" align="center">
        <tr></tr>
        <xsl:choose>
          <xsl:when test="AttachmentLst/Attachment/AttachmentSubType='IMG_FULL_
BANNER'">
            <!--got a full banner image for this Campaign-->
            <xsl:for-each
select="AttachmentLst/Attachment[starts-with(AttachmentSubType, 'IMG_FULL_
BANNER')]">
              <tr>
                <td>
```

```
                  <xsl:call-template name="IMAGE"/>
                </td>
              </tr>
            </xsl:for-each>
          </xsl:when>
          <xsl:when test="AttachmentLst/Attachment/AttachmentSubType='IMG_HLF_
BANNER'">
            <!--got a half banner image for this Campaign-->
            <xsl:for-each
select="AttachmentLst/Attachment[starts-with(AttachmentSubType, 'IMG_HLF_
BANNER')]">
              <tr>
                <td>
                  <xsl:call-template name="IMAGE"/>
                </td>
              </tr>
            </xsl:for-each>
          </xsl:when>
          <xsl:otherwise>
            <!--no full or half banner image available for this Campaign-->
            <xsl:for-each
select="AttachmentLst/Attachment[starts-with(AttachmentSubType, 'TXT_DESC')]">
              <tr>
                <td>
                  <xsl:call-template name="TEXT"/>
                </td>
              </tr>
            </xsl:for-each>
          </xsl:otherwise>
        </xsl:choose>
      </table>
    </td>
  </tr>
</xsl:template>
<xsl:template name="IMAGE">
  <xsl:variable name="height">
    <xsl:value-of select="floor(child::DisplayHeight)"/>
  </xsl:variable>
  <xsl:variable name="width">
    <xsl:value-of select="floor(child::DisplayWidth)"/>
  </xsl:variable>
  <xsl:element name="a">
    <xsl:attribute name="href">
      <xsl:value-of select="child::linkURL"/>
    </xsl:attribute>
    <xsl:element name="img">
      <xsl:attribute name="src">
        <xsl:value-of select="child::FileLocation"/>
      </xsl:attribute>
      <xsl:attribute name="alt">
        <xsl:value-of select="child::AlternateText"/>
      </xsl:attribute>
      <!--add height attribute if non-zero and number-->
      <xsl:if test="not($height='NaN')">
        <xsl:if test="not($height='0')">
          <xsl:attribute name="height">
           <xsl:value-of select="$height"/>
          </xsl:attribute>
        </xsl:if>
      </xsl:if>
      <!--add width attribute if non-zero and number-->
      <xsl:if test="not($width='NaN')">
        <xsl:if test="not($width='0')">
          <xsl:attribute name="width">
           <xsl:value-of select="$width"/>
          </xsl:attribute>
        </xsl:if>
      </xsl:if>
      <xsl:attribute name="border">
        <xsl:text>0</xsl:text>
      </xsl:attribute>
    </xsl:element>
  </xsl:element>
</xsl:template>
<xsl:template match="TEXT">
      <xsl:element name="a">
        <xsl:element name="img">
          <xsl:attribute name="src">
            <!--default image source file name-->
            <xsl:text></xsl:text>
          </xsl:attribute>
          <xsl:attribute name="alt">
            <!--default image alternate text-->
            <xsl:text></xsl:text>
          </xsl:attribute>
        </xsl:element>
      </xsl:element>
      <br>
      </br>
      <!--place the Campaign's short description here-->
      <xsl:element name="a">
        <xsl:attribute name="href">
          <xsl:value-of select="child::linkURL"/>
        </xsl:attribute>
        <xsl:value-of select="child::DisplayText"/>
      </xsl:element>
</xsl:template>
<xsl:template match="*"></xsl:template>
</xsl:stylesheet>
```

## ibacpns2.xsl

```
<?xml version="1.0"?>
<!--$Header: ibasinglecpnpstng2.xsl 115.3 2000/06/21 23:13:39 asaha ship $-->
<!--This stylesheet is for rendering Camapign by its short description,
description-->
<!--and Micro Button image.The attachment types expected by this stylesheet
are-->
<!--TXT_SHORT_DESC, TXT_DESC and IMG_MICRO_BUTTON-->
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
<xsl:template match="/">
  <table border="0" cellpadding="1" cellspacing="2">
    <xsl:apply-templates select="//Campaign"/>
  </table>
</xsl:template>
<xsl:template match="Campaign">
  <tr>
    <td>
      <table border="0" cellpadding="2" cellspacing="0" align="center">
        <tr></tr>
        <xsl:for-each
select="AttachmentLst/Attachment[starts-with(AttachmentSubType, 'TXT_DESC')]">
          <tr>
            <td>
              <xsl:call-template name="SHORTDESC"/>
            </td>
          </tr>
        </xsl:for-each>
        <xsl:for-each
select="AttachmentLst/Attachment[starts-with(AttachmentSubType, 'TXT_LONG_
DESC')]">
          <tr>
            <td>
              <xsl:call-template name="DESC"/>
            </td>
          </tr>
        </xsl:for-each>
        <xsl:for-each
select="AttachmentLst/Attachment[starts-with(AttachmentSubType, 'IMG_MICRO_
BUTTON')]">
          <tr>
            <td>
              <xsl:call-template name="IMAGE"/>
            </td>
          </tr>
        </xsl:for-each>
      </table>
    </td>
  </tr>
</xsl:template>
<xsl:template name="IMAGE">
  <xsl:variable name="height">
    <xsl:value-of select="floor(child::DisplayHeight)"/>
  </xsl:variable>
  <xsl:variable name="width">
    <xsl:value-of select="floor(child::DisplayWidth)"/>
  </xsl:variable>
  <xsl:element name="a">
    <xsl:attribute name="href">
      <xsl:value-of select="child::linkURL"/>
    </xsl:attribute>
    <xsl:element name="img">
      <xsl:attribute name="src">
        <xsl:value-of select="child::FileLocation"/>
      </xsl:attribute>
      <xsl:attribute name="alt">
        <xsl:value-of select="child::AlternateText"/>
      </xsl:attribute>
      <!--add height attribute if non-zero and number-->
      <xsl:if test="not($height='NaN')">
        <xsl:if test="not($height='0')">
          <xsl:attribute name="height">
           <xsl:value-of select="$height"/>
          </xsl:attribute>
        </xsl:if>
      </xsl:if>
      <!--add width attribute if non-zero and number-->
      <xsl:if test="not($width='NaN')">
        <xsl:if test="not($width='0')">
          <xsl:attribute name="width">
```

```
           <xsl:value-of select="$width"/>
          </xsl:attribute>
        </xsl:if>
      </xsl:if>
      <xsl:attribute name="border">
        <xsl:text>0</xsl:text>
      </xsl:attribute>
    </xsl:element>
  </xsl:element>
</xsl:template>
<xsl:template name="SHORTDESC">
  <xsl:element name="a">
    <xsl:attribute name="href">
      <xsl:value-of select="child::linkURL"/>
    </xsl:attribute>
    <xsl:value-of select="child::DisplayText"/>
  </xsl:element>
</xsl:template>
<xsl:template name="DESC">
  <xsl:element name="a">
    <xsl:value-of select="child::DisplayText"/>
  </xsl:element>
</xsl:template>
<xsl:template match="*"></xsl:template>
</xsl:stylesheet>
```

## ibacpnm3.xsl

```
<?xml version="1.0"?>
<!--$Header: ibamultiplecpnpstng3.xsl 115.3 2000/06/21 23:13:39 asaha ship $-->
<!--This stylesheet is for rendering Camapign list. Every Campaign in the
list-->
<!--has its short desription and micro button image.The attachment types
expected by-->
<!--this stylesheet are TXT_SHORT_DESC and IMG_MICRO_BUTTON-->

<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
<xsl:template match="/">
  <table border="0" cellpadding="1" cellspacing="2">
    <xsl:apply-templates select="//Campaign"/>
  </table>
</xsl:template>
<xsl:template match="Campaign">
  <tr>
    <td>
```

```
      <table border="0" cellpadding="2" cellspacing="0" align="center">
        <tr>
          <xsl:for-each
select="AttachmentLst/Attachment[starts-with(AttachmentSubType, ’TXT_DESC’)]">
            <td>
              <xsl:call-template name="TEXT"/>
            </td>
          </xsl:for-each>
          <xsl:for-each
select="AttachmentLst/Attachment[starts-with(AttachmentSubType, ’IMG_MICRO_
BUTTON’)]">
            <td>
              <xsl:call-template name="IMAGE"/>
            </td>
          </xsl:for-each>
        </tr>
      </table>
    </td>
  </tr>
</xsl:template>
<xsl:template name="IMAGE">
  <xsl:variable name="height">
    <xsl:value-of select="floor(child::DisplayHeight)"/>
  </xsl:variable>
  <xsl:variable name="width">
    <xsl:value-of select="floor(child::DisplayWidth)"/>
  </xsl:variable>
  <xsl:element name="a">
    <xsl:attribute name="href">
      <xsl:value-of select="child::linkURL"/>
    </xsl:attribute>
    <xsl:element name="img">
      <xsl:attribute name="src">
        <xsl:value-of select="child::FileLocation"/>
      </xsl:attribute>
      <xsl:attribute name="alt">
        <xsl:value-of select="child::AlternateText"/>
      </xsl:attribute>
      <!--add height attribute if non-zero and number-->
      <xsl:if test="not($height=’NaN’)">
        <xsl:if test="not($height=’0’)">
          <xsl:attribute name="height">
           <xsl:value-of select="$height"/>
          </xsl:attribute>
        </xsl:if>
      </xsl:if>
      <!--add width attribute if non-zero and number-->
      <xsl:if test="not($width='NaN')">
        <xsl:if test="not($width='0')">
          <xsl:attribute name="width">
           <xsl:value-of select="$width"/>
          </xsl:attribute>
        </xsl:if>
      </xsl:if>
      <xsl:attribute name="border">
        <xsl:text>0</xsl:text>
      </xsl:attribute>
    </xsl:element>
  </xsl:element>
</xsl:template>
<xsl:template name="TEXT">
  <xsl:element name="a">
    <xsl:attribute name="href">
      <xsl:value-of select="child::linkURL"/>
    </xsl:attribute>
    <xsl:value-of select="child::DisplayText"/>
  </xsl:element>
</xsl:template>
<xsl:template match="*"></xsl:template>
</xsl:stylesheet>
```

## Style Sheets for Products

### ibaprod1.xsl

```
<?xml version="1.0"?>
<!--$Header: ibaproductpstng1.xsl 115.3 2000/06/21 23:13:39 asaha ship $-->
<!--This stylesheet is for rendering Product list. The properties of Product-->
<!--object accessed here are ProductDetailLink and ShortDescription.-->
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
<xsl:template match="/">
  <table border="0" cellpadding="1" cellspacing="2">
    <xsl:apply-templates select="//Product"/>
  </table>
</xsl:template>
<xsl:template match="Product">
  <tr>
    <td>
      <table border="0" cellpadding="2" cellspacing="0" align="center">
```

```
        <tr>
          <td>
            <xsl:element name="a">
              <xsl:attribute name="href">
                <xsl:value-of select="child::ProductDetailLink"/>
              </xsl:attribute>
              <xsl:value-of select="child::ShortDescription"/>
            </xsl:element>
          </td>
        </tr>
      </table>
    </td>
  </tr>
</xsl:template>
<xsl:template match="*"></xsl:template>
</xsl:stylesheet>
```

## ibaprod2.xsl

```
<?xml version="1.0"?>
<!--$Header: ibaproductpstng2.xsl 115.3 2000/06/21 23:13:39 asaha ship $-->
<!--This stylesheet is for rendering Product list. The properties of Product
object -->
<!--accessed here are ProductDetailLink, ShortDescription and ThumbNail
image.-->
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
<xsl:template match="/">
  <table border="0" cellpadding="1" cellspacing="2">
    <xsl:apply-templates select="//Product"/>
  </table>
</xsl:template>
<xsl:template match="Product">
  <tr>
    <td>
      <table border="0" cellpadding="2" cellspacing="0" align="center">
        <tr>
          <td>
            <xsl:element name="a">
              <xsl:attribute name="href">
                <xsl:value-of select="child::ProductDetailLink"/>
              </xsl:attribute>
              <xsl:value-of select="child::ShortDescription"/>
            </xsl:element>
          </td>
          <td>
```

```
            <xsl:element name="a">
              <xsl:element name="img">
                <xsl:attribute name="src">
                  <xsl:value-of select="child::ThumbNail"/>
                </xsl:attribute>
                <xsl:attribute name="alt">
                  <xsl:text></xsl:text>
                </xsl:attribute>
                <xsl:attribute name="border">
                  <xsl:text>0</xsl:text>
                </xsl:attribute>
              </xsl:element>
            </xsl:element>
          </td>
        </tr>
      </table>
    </td>
  </tr>
</xsl:template>
<xsl:template match="*"></xsl:template>
</xsl:stylesheet>
```

### ibaprod3.xsl

```
<?xml version="1.0"?>
<!--$Header: ibaproductpstng3.xsl 115.3 2000/06/21 23:13:39 asaha ship $-->
<!--This stylesheet is for rendering Product list. The properties of Product
object -->
<!--accessed here are ProductDetailLink, ShortDescription, ThumbNail image,
Description-->
<!--and Shopping Cart Link.-->
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
<xsl:template match="/">
  <table border="0" cellpadding="1" cellspacing="2">
    <xsl:apply-templates select="//Product"/>
  </table>
</xsl:template>
<xsl:template match="Product">
  <!--row for ThumbNail image and short description-->
  <tr>
    <td>
      <table border="0" cellpadding="2" cellspacing="0" align="left">
        <tr>
          <td>
            <xsl:element name="a">
            <xsl:element name="img">
              <xsl:attribute name="src">
                <xsl:value-of select="child::ThumbNail"/>
              </xsl:attribute>
              <xsl:attribute name="alt">
                <xsl:text></xsl:text>
              </xsl:attribute>
              <xsl:attribute name="border">
                <xsl:text>0</xsl:text>
              </xsl:attribute>
            </xsl:element>
          </xsl:element>
        </td>
        <td>
          <xsl:element name="a">
            <xsl:attribute name="href">
              <xsl:value-of select="child::ProductDetailLink"/>
            </xsl:attribute>
            <xsl:value-of select="child::ShortDescription"/>
          </xsl:element>
        </td>
      </tr>
    </table>
  </td>
</tr>
<!--row for product detail page and shopping cart link-->
<tr>
  <td>
    <table border="0" cellpadding="2" cellspacing="0" align="left">
      <!--row for description-->
      <tr>
      <td>
        <xsl:element name="a">
          <xsl:value-of select="child::Description"/>
        </xsl:element>
      </td>
      </tr>
      <tr>
      <td>
        <xsl:element name="a">
          <xsl:attribute name="href">
            <xsl:value-of select="child::ShoppingCartLink"/>
          </xsl:attribute>
          <xsl:text>Add To Cart</xsl:text>
        </xsl:element>
        </td>
        </tr>
      </table>
    </td>
  </tr>
  <tr>
    <td>
    </td>
  </tr>
</xsl:template>
<xsl:template match="*"></xsl:template>
</xsl:stylesheet>
```

### ibaprod4.xsl

```
<?xml version="1.0"?>
<!--$Header: ibaproductpstng3.xsl 115.3 2000/06/21 23:13:39 asaha ship $-->
<!--This stylesheet is for rendering Product list. The properties of Product
object -->
<!--accessed here are ProductDetailLink, ShortDescription, Small image,
Description-->
<!--and Shopping Cart Link.-->
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
<xsl:template match="/">
  <table border="0" cellpadding="1" cellspacing="2">
    <xsl:apply-templates select="//Product"/>
  </table>
</xsl:template>
<xsl:template match="Product">
  <!--row for ThumbNail image and short description-->
  <tr>
    <td>
      <table border="0" cellpadding="2" cellspacing="0" align="left">
        <tr>
          <td>
            <xsl:element name="a">
              <xsl:element name="img">
                <xsl:attribute name="src">
                  <xsl:value-of select="child::SmallImage"/>
                </xsl:attribute>
                <xsl:attribute name="alt">
                  <xsl:text></xsl:text>
                </xsl:attribute>
                <xsl:attribute name="border">
                  <xsl:text>0</xsl:text>
              </xsl:attribute>
            </xsl:element>
          </xsl:element>
        </td>
        <td>
          <xsl:element name="a">
            <xsl:attribute name="href">
              <xsl:value-of select="child::ProductDetailLink"/>
            </xsl:attribute>
            <xsl:value-of select="child::ShortDescription"/>
          </xsl:element>
        </td>
      </tr>
    </table>
  </td>
</tr>
<!--row for product detail page and shopping cart link-->
<tr>
  <td>
    <table border="0" cellpadding="2" cellspacing="0" align="left">
      <!--row for description-->
      <tr>
      <td>
        <xsl:element name="a">
          <xsl:value-of select="child::Description"/>
        </xsl:element>
      </td>
      </tr>
      <tr>
      <td>
        <xsl:element name="a">
          <xsl:attribute name="href">
            <xsl:value-of select="child::ShoppingCartLink"/>
          </xsl:attribute>
          <xsl:text>Add To Cart</xsl:text>
        </xsl:element>
      </td>
      </tr>
    </table>
  </td>
</tr>
<tr>
  <td>
  </td>
</tr>
```

```
</xsl:template>
<xsl:template match="*"></xsl:template>
</xsl:stylesheet>
```

# Implementing Oracle iMarketing

For information on Implementing Oracle iMarketing, see the latest version of *Oracle iStore and iMarketing Implementation Guide*.

## Setting up Roles

For information on Implementing Oracle iMarketing, see the latest version of *Oracle iStore and iMarketing Implementation Guide*.