Oracle® Scripting

Concepts and Procedures

Release 11*i*

October 2001

Part No. A95150-01

ORACLE®

Oracle Scripting Concepts and Procedures, Release 11*i*

Part No. A95150-01

Contents

Send Us Your Comments xi

Preface xiii

1 Understanding Oracle Scripting

1.1 Oracle Scripting Overview 1-1
1.1.1 What Is Oracle Scripting? 1-1
1.1.2 Why Use Oracle Scripting? 1-3
1.1.3 Who Are the End Users of Oracle Scripting? 1-4
1.1.3.1 Script Author Users 1-4
1.1.3.2 Scripting Engine Users 1-5
1.1.3.3 Survey Runtime Users 1-6
1.1.3.4 Survey Administrative Users 1-7
1.1.4 Why Is Oracle Scripting in the Interaction Center Suite? 1-7
1.2 Understanding the Script Author 1-8
1.2.1 How to Obtain the Script Author 1-8
1.2.2 What Are the Building Blocks of a Script? 1-9
1.2.2.1 Configurable Objects 1-10
1.2.2.1.1 Panels 1-10
1.2.2.1.2 Groups 1-10
1.2.2.1.3 Blocks 1-10
1.2.3 Non-Configurable Objects 1-11
1.2.3.1 Start Nodes 1-11
1.2.3.2 Termination Nodes 1-11

1.2.4 Branches ... 1-12
1.2.5 What Are the Minimum Requirements for Any Graph? ... 1-13
1.3 Understanding the Scripting Engine ... 1-14
1.3.1 The Panel Display Area ... 1-15
1.3.2 The Progress Area ... 1-16
1.3.3 The Script Information Area ... 1-17
1.3.4 The Shortcut Button Bar ... 1-17
1.3.5 The Disconnect Button ... 1-18
1.3.6 The Toolbar ... 1-18
1.4 Understanding the Survey Component ... 1-19
1.4.1 The Survey Campaign ... 1-19
1.4.2 The Survey Questionnaire ... 1-20
1.4.3 The Survey Admin Console ... 1-21

2 Planning Oracle Scripting Projects

2.1 Planning Scripting Engine Projects ... 2-1
2.1.1 Facets of Scripting-Specific Discovery Process ... 2-2
2.1.2 Bringing Together the Layers ... 2-5
2.1.3 Tools to Aid in Scoping and Discovery ... 2-6
2.1.4 Oracle Scripting Discovery Data Worksheet ... 2-7
2.1.5 Oracle Scripting Discovery Checklist Tool ... 2-9
2.2 Planning Oracle Scripting Survey Campaigns ... 2-11
2.2.1 Gathering All Survey Campaign Requirements ... 2-14
2.2.1.1 Planning Requirements for All Survey Campaigns ... 2-15
2.2.1.2 Additional Planning Requirements for List-Based Survey Campaigns ... 2-16
2.2.1.3 Methodology ... 2-17
2.2.2 Requirements for Creating and Deploying a Survey Questionnaire ... 2-18
2.2.2.1 Appropriate Network, Security, Hardware, and Software Resources ... 2-19
2.2.2.2 Trained Script Developers ... 2-21
2.2.2.3 Trained Java, PL/SQL, Oracle Forms, and API Programmers ... 2-22
2.2.2.4 Trained Oracle Applications and Database Administrators ... 2-23
2.2.2.5 Detailed Script-Specific Information ... 2-23
2.2.2.6 Environment-Specific Information ... 2-24
2.2.2.7 Detailed Survey Questionnaire Requirements ... 2-24
2.2.3 Requirements for Determining If Survey Campaign Is List-Based ... 2-25

2.2.3.1 List-Based Campaign = NO ... 2-26
2.2.3.2 List-Based Campaign = YES ... 2-26
2.2.4 Requirements for Generating Lists ... 2-27
2.2.4.1 Additional Implementation Requirements for List-Based Survey Campaigns ... 2-27
2.2.5 Requirements for Administering Survey Resources and Survey Campaign and Cycle Details 2-28
2.2.6 Requirements for Defining Deployments ... 2-30
2.2.7 Requirements for Deploying Survey Campaigns ... 2-31
2.2.8 Monitoring Survey Results ... 2-33
2.2.9 Collecting Survey Results in Oracle RDBMS ... 2-33
2.2.10 Requirements for Reporting Survey Campaign Deployment Results ... 2-34

3 Using Oracle Scripting

3.1 Using the Script Author ... 3-1
3.1.1 Getting Started with the Script Author ... 3-3
3.1.2 Obtaining Script Requirements Before Creating a Script ... 3-4
3.1.3 Defining Global Script Attributes ... 3-6
3.1.3.1 Designating Global Script Properties ... 3-7
3.1.3.1.1 Defining the Script Name ... 3-8
3.1.3.1.2 Defining Script Comments ... 3-9
3.1.3.1.3 Defining Script Language ... 3-10
3.1.3.1.4 Setting Footprinting ... 3-12
3.1.3.1.5 Setting Answer Collection for the Script ... 3-13
3.1.3.2 Defining Global Script Pre- and Post-Actions ... 3-14
3.1.3.3 Defining the Script Information Panel ... 3-15
3.1.3.4 Defining Shortcut Buttons ... 3-16
3.1.3.5 Programming the Script Disconnect Button ... 3-18
3.1.4 Working with Script Files ... 3-19
3.1.4.1 Starting a New Script ... 3-20
3.1.4.2 Opening an Existing Script from the File System ... 3-21
3.1.4.3 Opening an Existing Script from the Applications Database ... 3-22
3.1.4.4 Saving a Script to the File System ... 3-24
3.1.4.5 Reversing All Changes Since the Last Save Command ... 3-25
3.1.4.6 Importing a Script ... 3-26

3.1.4.7 Exporting a Script Group as a Separate Script File 3-27
3.1.4.8 Printing a Graph on the Script Author Canvas 3-28
3.1.4.9 Closing a Script 3-29
3.1.4.10 Toggling Sticky Mode 3-30
3.1.4.11 Exiting the Script Author 3-31
3.1.5 Working with the Tool Palette 3-32
3.1.5.1 Inserting an Object 3-33
3.1.5.1.1 Setting Properties Dialog Box to Pop Up at Object Creation 3-34
3.1.5.1.2 Inserting a Termination Node 3-34
3.1.5.2 Inserting a Branch 3-35
3.1.6 Viewing Objects on the Canvas 3-36
3.1.6.1 Fitting a Script Layout in the Canvas 3-37
3.1.6.2 Drilling Down Into or Up From a Group or Block 3-38
3.1.6.3 Aligning Objects 3-39
3.1.7 Working with Objects on the Canvas 3-40
3.1.7.1 Selecting Objects 3-40
3.1.7.2 Deselecting Objects 3-41
3.1.7.3 Moving Objects 3-42
3.1.7.4 Moving Branches 3-43
3.1.7.5 Deleting Panels, Groups, Blocks and Termination Nodes 3-44
3.1.7.6 Deleting Branches 3-45
3.1.8 Defining Panels 3-46
3.1.8.1 Inserting a Panel 3-47
3.1.8.2 Defining Panel Properties 3-48
3.1.8.2.1 Defining the Panel Name 3-49
3.1.8.2.2 Defining Panel Comments 3-50
3.1.8.2.3 Defining the Panel Label 3-50
3.1.8.2.4 Substituting a Java Bean for a Panel 3-51
3.1.8.2.5 Packaging Java Bean Code Into a JAR File 3-53
3.1.8.3 Defining Panel Text 3-53
3.1.8.3.1 Entering Panel Text 3-54
3.1.8.3.2 Inserting a Hyperlink 3-55
3.1.8.3.3 Inserting an Embedded Value 3-56
3.1.8.3.4 Inserting an Image 3-58
3.1.8.3.5 Exporting Panel Text to an HTML File 3-59

3.1.8.3.6 Importing an HTML File into the Panel Text Editor 3-60
3.1.9 Defining Groups 3-61
3.1.9.1 Inserting a Group 3-62
3.1.9.2 Importing a Saved Script as a Group 3-63
3.1.9.3 Defining the Group Name 3-64
3.1.10 Defining Blocks 3-65
3.1.10.1 Inserting a Block 3-67
3.1.10.2 Defining the Block Name 3-68
3.1.10.3 Defining a Database Query Block 3-69
3.1.10.4 Defining a Database Insert Block 3-70
3.1.10.5 Defining a Database Update Block 3-73
3.1.10.6 Defining Database Connections for a Block 3-76
3.1.10.7 Defining Database Tables for a Block 3-77
3.1.10.8 Defining Join Conditions for a Block 3-78
3.1.10.9 Defining Data Constraints for an Insert or Update Block 3-80
3.1.10.10 Defining Query Constraints for a Query Block 3-81
3.1.10.11 Defining Query Columns for a Query Block 3-83
3.1.10.12 Defining Panels Within a Block 3-84
3.1.11 Defining Branches 3-85
3.1.11.1 Setting Properties Dialog Box to Pop Up at Branch Creation 3-86
3.1.11.2 Defining a Default Branch 3-87
3.1.11.3 Defining a Distinct Branch 3-88
3.1.11.4 Defining a Conditional Branch 3-89
3.1.11.5 Defining an Indeterminate Branch 3-91
3.1.11.6 Defining the Branch Name 3-93
3.1.11.7 Reordering Branches 3-94
3.1.11.8 Working with Branches Visually on the Canvas 3-95
3.1.11.8.1 Inserting a Straight Branch 3-95
3.1.11.8.2 Inserting a Branch with Corners 3-96
3.1.11.8.3 Adding a Corner to an Existing Branch 3-97
3.1.11.8.4 Moving a Corner of a Branch 3-97
3.1.11.8.5 Deleting a Corner from a Branch 3-98
3.1.12 Controlling Script Flow 3-99
3.1.13 Defining Panel Answer Controls 3-100
3.1.13.1 Defining a Text Field Answer Control in a Panel 3-101

3.1.13.2 Defining a Text Area Answer Control in a Panel 3-103
3.1.13.3 Defining a Radio Button Group Answer Control in a Panel 3-105
3.1.13.4 Defining a Check Box Group Answer Control in a Panel 3-107
3.1.13.5 Defining a Button Answer Control in a Panel 3-108
3.1.13.6 Defining a Drop Down List Answer Control in a Panel 3-110
3.1.13.7 Defining a Password Answer Control in a Panel 3-112
3.1.14 Working With Answers 3-114
3.1.14.1 Defining Vertical or Horizontal Answer Control Layout 3-115
3.1.14.2 Adding Validation to an Answer Definition 3-116
3.1.14.3 Defining Data Constraints for an Answer Control 3-118
3.1.14.4 Reordering Answer Options 3-119
3.1.14.5 Reordering Answer Controls 3-120
3.1.14.6 Deleting Answer Options from an Answer Control 3-121
3.1.14.7 Deleting Answer Controls from a Script Panel 3-122
3.1.14.8 Substituting a Java Bean for an Answer 3-123
3.1.15 Defining Actions 3-124
3.1.15.1 Defining Script Pre- and Post-Actions 3-125
3.1.15.2 Defining Panel Pre- and Post-Actions 3-126
3.1.15.3 Defining Group Pre- and Post-Actions 3-127
3.1.15.4 Defining Block Pre- and Post- Actions 3-127
3.1.15.5 Defining Branch Actions 3-128
3.1.16 Defining Commands 3-129
3.1.16.1 Invoking a Public Method in a Java Class 3-130
3.1.16.2 Invoking a PL/SQL Stored Procedure or Function in a Database 3-132
3.1.16.3 Invoking a PL/SQL Procedure or Function in a Forms Server 3-134
3.1.16.4 Setting a Constant 3-135
3.1.16.5 Retrieving a Panel Answer From the Scripting Blackboard 3-136
3.1.17 Deploying the Script 3-137
3.1.17.1 Checking Script Syntax 3-138
3.1.17.2 Deploying a Script to the Database 3-139
3.1.17.3 Connecting to the Database 3-140
3.2 Troubleshooting the Script Author 3-141
3.2.1 Obtaining Help in the Script Author 3-142
3.2.2 Storing Strings in the Script Author 3-143
3.2.3 Known Issues Using COMMIT in Custom PL/SQL Procedures 3-143

3.3 Using the Scripting Engine 3-144
3.4 Taking a Survey 3-145
3.5 Using the Survey Admin Console 3-147
3.5.1 Campaigns and the Survey Component of Oracle Scripting 3-147
3.5.2 Survey Campaign, Cycle, Deployment Hierarchy 3-148
3.5.3 Survey Media Channels 3-150
3.5.4 Survey Respondent Entry Points 3-151
3.5.4.1 List-Based Entry Points 3-151
3.5.4.2 Non-List-Based Entry Points 3-152

4 Administering Oracle Scripting Survey Campaigns

4.1 Navigating the Survey Admin Console Graphic User Interface 4-2
4.1.1 Navigating Using Tabs and Sub-Tabs 4-3
4.1.2 Using the **Quick Fin**d Feature 4-5
4.1.3 Using the View By Dropdown Control 4-6
4.1.4 Setting General Preferences for Logged-In User in the Survey Admin Console .. 4-7
4.2 Administering Survey Resources 4-9
4.2.1 Creating Survey Resources 4-10
4.2.2 Defining Survey Resources 4-11
4.2.3 Uploading Survey Resources 4-14
4.2.4 Setting the Default Survey Resource in the Survey Admin Console 4-15
4.2.5 Non-List-Based URL Syntax and Example 4-17
4.3 Administering Survey Campaign Details 4-18
4.3.1 Creating Survey Campaigns 4-18
4.3.2 Defining Cycles 4-21
4.3.3 Defining Deployments 4-24
4.3.4 Deploying or Suspending a Survey Campaign Deployment 4-29
4.3.4.1 Deploying a Survey Campaign Deployment 4-30
4.3.4.2 Suspending a Survey Campaign Deployment 4-31
4.3.4.3 Understanding Status Conditions 4-33
4.3.4.4 Status Conditions for Survey Campaigns, Cycles, and Deployments 4-34
4.3.5 Modifying Survey Campaign Details 4-36
4.4 Generating Lists for List-Based Survey Campaigns 4-36
4.5 Setting Up Invitations and Reminders for List-Based Survey Campaigns 4-37
4.5.1 Fulfillment Activities Required for List-Based Survey Campaigns 4-38

4.5.2 Recommended Order of Steps 4-39
4.5.3 Invitations 4-40
4.5.4 Reminders 4-40
4.5.5 Sample Invitation Master Document 4-41
4.5.6 Sample Query 4-42
4.5.7 Verifying Status of a Fulfillment Request 4-43
4.6 Managing Survey Campaign Responses 4-44
4.6.1 Navigating the Response Tab 4-45
4.6.2 Viewing Individual Responses in the Survey Admin Console 4-45
4.6.3 Understanding the GUI Display of Individual Responses 4-51
4.6.3.1 Relationship to the Script Author 4-52
4.6.3.2 Survey Responses GUI Description 4-52
4.6.3.3 How Different UI Types Display 4-53
4.6.4 Manually Sending Reminders to Individual List Members 4-54
4.6.5 Administering Concurrent Processing to Export Respondent Data to Survey Summary Tables 4-56
4.7 Reporting and Analyzing Survey Respondent Data 4-58

Send Us Your Comments

Oracle Scripting Concepts and Procedures, Release 11*i*

Part No. A95150-01

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 in the following ways:

- Postal service:
  Oracle Corporation
  Oracle Scripting Documentation
  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

Audience for This Guide

Welcome to Release 11*i* of the Oracle® Scripting Concepts and Procedures Manual.

This guide assumes you have a working knowledge of the following:

- The principles and customary practices of your business area.
- Oracle Scripting

  If you have never used Oracle Scripting, Oracle suggests you attend one or more of the Oracle Scripting training classes available through Oracle University.
- The Oracle Applications graphical user interface.

  To learn more about the Oracle Applications graphical user interface, read the *Oracle Applications User's Guide*.

See Other Information Sources for more information about Oracle Applications product information.

How To Use This Guide

This guide contains the information you need to understand and use Oracle Scripting.

- **Understanding Oracle Scripting** includes an overview of Oracle Scripting, followed by sections to help you understand each of the three components of Oracle Scripting: Understanding the Script Author, Understanding the Scripting Engine, and Understanding the Survey Component.

- **Planning Oracle Scripting Projects** describes planning aspects required for the two different execution components of Oracle Scripting, including Planning Scripting Engine Projects and Planning Oracle Scripting Survey Campaigns.
- **Using Oracle Scripting** describes the use of the various components of Oracle Scripting from an agent, end-user, developer, administrator, and customer perspective. This section includes Using the Script Author, Troubleshooting the Script Author, Using the Scripting Engine, Taking a Survey, and Using the Survey Admin Console.
- **Administering Oracle Scripting Survey Campaigns** describes in detail the various tasks required to administer Oracle Scripting survey campaigns successfully.

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible, with good usability, to the disabled community. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Standards will continue to evolve over time, and Oracle Corporation is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For additional information, visit the Oracle Accessibility Program Web site at http://www.oracle.com/accessibility/.

Accessibility of Code Examples in Documentation

JAWS, a Windows screen reader, may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, JAWS may not always read a line of text that consists solely of a bracket or brace.

Accessibility of Links to External Web Sites in Documentation

This documentation may contain links to Web sites of other companies or organizations that Oracle Corporation does not own or control. Oracle Corporation neither evaluates nor makes any representations regarding the accessibility of these Web sites.

Other Information Sources

You can choose from many sources of information, including online documentation, training, and support services, to increase your knowledge and understanding of Oracle Scripting.

If this guide refers you to other Oracle Applications documentation, use only the Release 11*i* versions of those guides.

Online Documentation

All Oracle Applications documentation is available online (HTML or PDF). Some online help patches are available on Oracle*MetaLink*. Oracle Scripting online help is only available with installation of Oracle Scripting release 11.5.6 or later.

Related Documentation

Oracle Scripting shares business and setup information with other Oracle Applications products. Therefore, you may want to refer to other product documentation when you set up and use Oracle Scripting.

You can read the documents online by choosing Library from the expandable menu on your HTML help window, by reading from the Oracle Applications Document Library CD included in your media pack, or by using a Web browser with a URL that your system administrator provides.

If you require printed guides, you can purchase them from the Oracle Store at http://oraclestore.oracle.com.

Documents Related to All Products

Oracle Applications User's Guide

This guide explains how to enter data, query, run reports, and navigate using the graphical user interface (GUI) available with this release of Oracle Scripting (and any other Oracle Applications products). This guide also includes information on setting user profiles, as well as running and reviewing reports and concurrent processes.

You can access this user's guide online by choosing "Getting Started with Oracle Applications" from any Oracle Applications help file.

Documents Related to This Product

Oracle Scripting Implementation Guide 11*i*

This guide describes how to implement Oracle Scripting components and test the implementation appropriately.

Installation and System Administration

Oracle Applications Concepts

This guide provides an introduction to the concepts, features, technology stack, architecture, and terminology for Oracle Applications Release 11*i*. It provides a useful first book to read before an installation of Oracle Applications. This guide also introduces the concepts behind Applications-wide features such as Business Intelligence (BIS), languages and character sets, and Self-Service Web Applications.

Installing Oracle Applications

This guide provides instructions for managing the installation of Oracle Applications products. In Release 11*i*, much of the installation process is handled using Oracle Rapid Install, which minimizes the time to install Oracle Applications, the Oracle8 technology stack, and the Oracle8*i* Server technology stack by automating many of the required steps. This guide contains instructions for using Oracle Rapid Install and lists the tasks you need to perform to finish your installation. You should use this guide in conjunction with individual product user's guides and implementation guides.

Oracle Applications Supplemental CRM Installation Steps

This guide contains specific steps needed to complete installation of a few of the CRM products. The steps should be done immediately following that tasks given in the Installing Oracle Applications guide.

Upgrading Oracle Applications

Refer to this guide if you are upgrading your Oracle Applications Release 10.7 or Release 11.0 products to Release 11*i*. This guide describes the upgrade process and lists database and product-specific upgrade tasks. You must be either at Release 10.7 (NCA, SmartClient, or character mode) or Release 11.0, to upgrade to Release 11*i*. You cannot upgrade to Release 11*i* directly from releases prior to 10.7.

Maintaining Oracle Applications

Use this guide to help you run the various AD utilities, such as AutoUpgrade, AutoPatch, AD Administration, AD Controller, AD Relink, License Manager, and others. It contains how-to steps, screenshots, and other information that you need to run the AD utilities. This guide also provides information on maintaining the Oracle applications file system and database.

Oracle Applications System Administrator's Guide

This guide provides planning and reference information for the Oracle Applications System Administrator. It contains information on how to define security, customize menus and online help, and manage concurrent processing.

Oracle Alert User's Guide

This guide explains how to define periodic and event alerts to monitor the status of your Oracle Applications data.

Oracle Applications Developer's Guide

This guide contains the coding standards followed by the Oracle Applications development staff. It describes the Oracle Application Object Library components needed to implement the Oracle Applications user interface described in the *Oracle Applications User Interface Standards for Forms-Based Products*. It also provides information to help you build your custom Oracle Forms Developer 6*i* forms so that they integrate with Oracle Applications.

Oracle Applications User Interface Standards for Forms-Based Products

This guide contains the user interface (UI) standards followed by the Oracle Applications development staff. It describes the UI for the Oracle Applications products and how to apply this UI to the design of an application built by using Oracle Forms.

Other Implementation Documentation

Multiple Reporting Currencies in Oracle Applications

If you use the Multiple Reporting Currencies feature to record transactions in more than one currency, use this manual before implementing Oracle Scripting. This manual details additional steps and setup considerations for implementing Oracle Scripting with this feature.

Multiple Organizations in Oracle Applications

This guide describes how to set up and use Oracle Scripting with Oracle Applications' Multiple Organization support feature, so you can define and support different organization structures when running a single installation of Oracle Scripting.

Oracle Workflow Guide

This guide explains how to define new workflow business processes as well as customize existing Oracle Applications-embedded workflow processes.You also use this guide to complete the setup steps necessary for any Oracle Applications product that includes workflow-enabled processes.

Oracle Applications Flexfields Guide

This guide provides flexfields planning, setup and reference information for the Oracle Scripting implementation team, as well as for users responsible for the ongoing maintenance of Oracle Applications product data. This manual also provides information on creating custom reports on flexfields data.

Oracle eTechnical Reference Manuals

Each eTechnical Reference Manual (eTRM) contains database diagrams and a detailed description of database tables, forms, reports, and programs for a specific Oracle Applications product. This information helps you convert data from your existing applications, integrate Oracle Applications data with non-Oracle applications, and write custom reports for Oracle Applications products. Oracle eTRM is available on Oracle*MetaLink*

Oracle Manufacturing APIs and Open Interfaces Manual

This manual contains up-to-date information about integrating with other Oracle Manufacturing applications and with your other systems. This documentation includes API's and open interfaces found in Oracle Manufacturing.

Oracle Order Management Suite APIs and Open Interfaces Manual

This manual contains up-to-date information about integrating with other Oracle Manufacturing applications and with your other systems. This documentation includes API's and open interfaces found in Oracle Order Management Suite.

Oracle Applications Message Reference Manual

This manual describes Oracle Applications messages. This manual is available in HTML format on the documentation CD-ROM for Release 11i.

Oracle CRM Application Foundation Implementation Guide

Many CRM products use components from CRM Application Foundation. Use this guide to correctly implement CRM Application Foundation.

Training and Support

Training

Oracle offers training courses to help you and your staff master Oracle Scripting and reach full productivity quickly. You have a choice of educational environments. You can attend courses offered by Oracle University at any one of our many Education Centers, you can arrange for our trainers to teach at your facility, or you can use Oracle Learning Network (OLN), Oracle University's online education utility. In addition, Oracle training professionals can tailor standard courses or develop custom courses to meet your needs. For example, you may want to use your organization structure, terminology, and data as examples in a customized training session delivered at your own facility.

Support

From on-site support to central support, our team of experienced professionals provides the help and information you need to keep Oracle Scripting working for you. This team includes your Technical Representative, Account Manager, and Oracle's large staff of consultants and support specialists with expertise in your business area, managing an Oracle8*i* server, and your hardware and software environment.

Oracle*MetaLink*

Oracle*MetaLink* is your self-service support connection with web, telephone menu, and e-mail alternatives. Oracle supplies these technologies for your convenience, available 24 hours a day, 7 days a week. With Oracle*MetaLink*, you can obtain information and advice from technical libraries and forums, download patches, download the latest documentation, look at bug details, and create or update TARs. To use Oracle*MetaLink*, register at (http://metalink.oracle.com).

Alerts: You should check Oracle*MetaLink* alerts before you begin to install or upgrade any of your Oracle Appl i cati ons. Navigate to the Alerts page as follows: Technical Libraries/ERP Applications/Applications Installation and Upgrade/Alerts.

Self-Service Toolkit: You may also find information by navigating to the Self-Service Toolkit page as follows: Technical Libraries/ERP Applications/Applications Installation and Upgrade.

Do Not Use Database Tools to Modify Oracle Applications Data

Oracle STRONGLY RECOMMENDS that you never use SQL*Plus, Oracle Data Browser, database triggers, or any other tool to modify Oracle Applications data unless otherwise instructed.

Oracle provides powerful tools you can use to create, store, change, retrieve, and maintain information in an Oracle database. But if you use Oracle tools such as SQL*Plus to modify Oracle Applications data, you risk destroying the integrity of your data and you lose the ability to audit changes to your data.

Because Oracle Applications tables are interrelated, any change you make using Oracle Applications can update many tables at once. But when you modify Oracle Applications data using anything other than Oracle Applications, you may change a row in one table without making corresponding changes in related tables. If your tables get out of synchronization with each other, you risk retrieving erroneous information and you risk unpredictable results throughout Oracle Applications.

When you use Oracle Applications to modify your data, Oracle Applications automatically checks that your changes are valid. Oracle Applications also keeps track of who changes information. If you enter information into database tables using database tools, you may store invalid information. You also lose the ability to track who has changed your information because SQL*Plus and other database tools do not keep a record of changes.

About Oracle

Oracle Corporation develops and markets an integrated line of software products for database management, applications development, decision support, and office automation, as well as Oracle Applications, an integrated suite of more than 160 software modules for financial management, supply chain management, manufacturing, project systems, human resources and customer relationship management.

Oracle products are available for mainframes, minicomputers, personal computers, network computers and personal digital assistants, allowing organizations to integrate different computers, different operating systems, different networks, and even different database management systems, into a single, unified computing and information resource.

Oracle is the world's leading supplier of software for information management, and the world's second largest software company. Oracle offers its database, tools, and applications products, along with related consulting, education, and support services, in over 145 countries around the world.

1 Understanding Oracle Scripting

This topic group includes the following topics:

Oracle Scripting Overview

Understanding the Script Author

Understanding the Scripting Engine

Understanding the Survey Component

1.1 Oracle Scripting Overview

This topic group includes the following topics:

What Is Oracle Scripting?

Why Use Oracle Scripting?

Who Are the End Users of Oracle Scripting?

Why Is Oracle Scripting in the Interaction Center Suite?

See Also

Understanding the Script Author

Understanding the Scripting Engine

Understanding the Survey Component

1.1.1 What Is Oracle Scripting?

Oracle Scripting is a set of tools to create and display information sequentially to its end users. The information displayed is based (1) on a predetermined set of

requirements that have been programmed into a script, and (2) on responses provided by the end user. End users are guided through a specific flow, which can either be rigidly prescribed to enforce consistency, or which can be dynamically determined based on answers entered into the graphic user interface (GUI).

Oracle Scripting is composed of three components. Each of these are described briefly below. For more information, see the Understanding section for each component.

Script Author

Script Author is a powerful authoring environment of graphical layout tools to create, modify, and deploy scripts. The Script Author is a two-tiered script development studio built in Java. The script developer inserts objects into the script in a work area known as the canvas, using an intuitive GUI. The developer then associates properties to each object. These properties range from the simple (object names, labels, or text to display at run time) to the complex (such as Java, PL/SQL, or other commands). The objects in the Script Author are linked by various branch types to provide a graph that looks similar to a flowchart. In this manner, the objects embedded in the script control complicated processing, such as rules-based branching and data integration. Scripts built in the Script Author can be run in either or both of the other Oracle Scripting components.

Scripting Engine

The Scripting Engine is a Java-based GUI executed from an Oracle Applications session. Using the Scripting Engine, Interaction Center agents are guided through their interactions with customers. The business logic built into the script, including custom Java, PL/SQL, and other commands, guides the agent through the script. As the trained agent moves effortlessly through guided messages represented by script panels, the Oracle Scripting objects embedded in the script control complicated processing. The business logic embedded in the script (such as rules-based branching, data integration, and commands associated with specific objects and events) is evaluated dynamically, along with the responses provided by the agent, progressing or flowing through the script in a logical manner and accomplishing the exchange of information required by the transaction. Oracle Scripting scripts can currently be launched by interaction center agents from within three business applications in the CRM Suite: Oracle TeleService, Oracle TeleSales, and Oracle Collections.

Survey Component

Using the Survey component of Oracle Scripting, a survey questionnaire (in the form of a script developed using the Script Author) can be made available for people to take using an Oracle Applications 11*i*-compliant Web browser. Regardless of the entry point, the individual taking the survey (the respondent) will view the script in a Web browser running JSP pages on the Apache Web server of the polling enterprise. (For more information on entry points, see Survey Respondent Entry Points in the Using Oracle Scripting section of this document.)

In order for this script to be made available as a survey questionnaire, a survey campaign must be defined by an administrator using the Survey Admin console. The purpose for executing scripts as a survey campaign is to obtain a specific set of data from the target population (or a general population). This data is typically analyzed by the polling enterprise, and actions can be taken as a result of the survey campaign. These actions are directly dependent on the purpose for conducting the campaign and the population surveyed. For example, a large company polling its employees might be interested in knowing how satisfied the employees are with the current benefits or compensation packages currently available. Typical uses of a survey campaign are to measure political opinion; to measure satisfaction; to determine whether a set of products or services meets the needs of a target population; or to perform market research to determine the viability of new products service lines.

See Also

Survey Respondent Entry Points

Why Use Oracle Scripting?

Who Are the End Users of Oracle Scripting?

Why Is Oracle Scripting in the Interaction Center Suite?

1.1.2 Why Use Oracle Scripting?

The Script Author is used to develop scripts for execution in either the Scripting Engine or as a survey using the Survey component of Oracle Scripting.

The Scripting Engine can be used in inbound, outbound, or blended interaction centers in order to guide agents in their interactions with customers or prospects.

The Survey component of Oracle Scripting can be used for any purpose for which information must be collected and analyzed. Typical uses include customer

vendor surveys, etc. Survey is helpful not only in obtaining relevant information from customers, prospective customers, consumers using competing products or services, and the general public, but also from more clearly targeted populations such as the employees within a company, shareholders of stock, and so forth.

The applications of the product are unlimited in terms of benefits to enterprises that need to survey general or specific populations and process the return data.

- Oracle Scripting release 11.5.6 includes four out-of-the-box reports available to assist in analysis of data obtained by executing Oracle Scripting. All are applicable to survey campaigns, and the Footprinting report is useful for interaction center reporting (use of the Scripting Engine). More reports are expected to be available in future releases. The information obtained from a survey is stored in the Oracle Applications database, and can be accessed using customized SQL scripts or by using any tool that can access an Oracle database.

See Also

What Is Oracle Scripting?

Who Are the End Users of Oracle Scripting?

Why Is Oracle Scripting in the Interaction Center Suite?

1.1.3 Who Are the End Users of Oracle Scripting?

This topic group includes the following topics:

- Script Author Users
- Scripting Engine Users
- Survey Runtime Users
- Survey Administrative Users

1.1.3.1 Script Author Users

The Script Author is intended for the functional user with some technical knowledge. The Script Author development environment provides for reuse of defined commands and existing Scripting components. The script developer will define these commands. Hooks in the GUI reference technical components (for example, Oracle Forms, custom Java methods, and PL/SQL packages stored in the database) which are primarily developed external to the Script Author. These components provide sophisticated functions to accomplish the functionality most enterprises want. They must be developed by individuals certified and

knowledgeable in the relevant technologies. The script developer must work with these highly technical resources to ensure the custom components are appropriately integrated into the script. The script developer must also ensure the code is appropriately loaded on the server and referenced in the script. This will ensure the code is available to the base Java classes that provide Scripting runtime functionality for the Scripting Engine and the Survey component at runtime.

Users of the Script Author may be campaign administrators, Java developers or database programmers, business process engineers, or experienced interaction center agents with technical aptitude. The keys to successful script development are: (1) familiarity with how Oracle Scripting captures and processes data, (2) knowledge of associated technologies (including access to experts in these technologies), and (3) adequate training and familiarity with existing documentation.

No Oracle Applications responsibilities are required to use the Script Author, as this is an independent Java environment not requiring login to an Oracle Applications session.

See Also

- Scripting Engine Users
- Survey Runtime Users
- Survey Administrative Users

1.1.3.2 Scripting Engine Users

End users of the Scripting Engine are trained interaction center agents (“agents”). These are non-technical users who have received simple but thorough training in the Java-based Scripting Engine GUI. These individuals include call center agents taking or presenting information over the telephone, as well as interaction center agents taking advantage of other media such as Intranets, enterprise portals over the World-Wide Web, and so forth. These agents typically have access to (and training in) at least one Oracle eBusiness application integrated with Oracle Scripting. While the Scripting Engine can be launched *without* an integrated application (in “stand-alone mode”) — typically used for testing a script — the Scripting Engine is intended to be used in combination with integrated Oracle eBusiness applications in the CRM Suite of applications to take full advantage of the contact handling and service, sales, or collections features of those applications, respectively.

The table below lists the Oracle Applications and associated responsibilities that can currently launch the Scripting Engine.

Responsibilities That Can Launch the Scripting Engine

Table 1–1 Oracle Applications Responsibilities That Can Launch the Scripting Engine

Responsibility Name	Application	Purpose
`Customer Support`	Oracle TeleService	Enable a regular Customer Support (TeleService) agent to launch a script
`TeleSales Agent`	Oracle TeleSales	Enable a regular TeleSales agent to launch a script
`Collections Agent`	Oracle Collections	Enable a regular Collections agent to launch a script
`Senior Collections Agent`	Oracle Collections	Enable a Senior Collections agent to launch a script
`Scripting User`	Scripting Engine	Launch Scripting Engine in stand-alone mode
`Scripting Agent, Vision Enterprises`	Scripting Engine	Launch Scripting Engine in stand-alone mode (for environments without `Scripting User` responsibility)

References

- For information about establishing responsibilities, refer to the *Oracle Applications System Administrator's Guide Release 11i,* in the section Managing Oracle Applications Security > Defining a Responsibility.
- For information about application-specific responsibilities, please refer to the Implementation Guide for each specific business application.

See Also

- Script Author Users
- Survey Runtime Users
- Survey Administrative Users

1.1.3.3 Survey Runtime Users

End users of the Survey component of Oracle Scripting from a runtime perspective are customers or prospects viewing a script (typically a survey questionnaire) in any Oracle Applications-compliant Web browser. These may have been invited to participate in a survey through an e-mail message or through navigation to a

survey-enabled site, or they may have accessed the survey through a self-service Web application scenario such as Oracle iSupport.

No Oracle Applications responsibilities are required to execute a script as a survey in a Web browser.

See Also

- Script Author Users
- Scripting Engine Users
- Survey Administrative Users

1.1.3.4 Survey Administrative Users

Users of the JSP/HTML-based survey campaign administrative console are non-technical users with access to detailed project requirements. These individuals are typically interaction center campaign administrators or system administrators.

To access the Survey Campaign administrative GUI (from the eBusiness Suite login), these users must have the `Survey Administrator` Oracle Applications responsibility.

See Also

What Is Oracle Scripting?

Why Use Oracle Scripting?

Why Is Oracle Scripting in the Interaction Center Suite?

1.1.4 Why Is Oracle Scripting in the Interaction Center Suite?

The Scripting Engine provides enterprises a method of scripting interactions with customers or prospects and integrating desktop workflow between various applications. It is intended to be used in combination with integrated Oracle eBusiness applications in the CRM Suite of applications (Oracle TeleService, Oracle TeleSales and Oracle Collections) to take full advantage of the contact handling and sales or service features of those applications, respectively. Interaction centers leveraging business applications and the Scripting Engine therefore have the full benefit of interaction center communications. Both the Scripting Engine and the Survey component of Oracle Scripting allow an enterprise to capture customer needs, perform market research, and measure customer satisfaction. The Survey component is a powerful information gathering tool that allows rapid deployment of tailored questionnaires to gather data from various targeted populations.

Information gained as a result of executing a survey campaign can be used to drive new business initiatives, immediately spot customer satisfaction issues, test new products or ideas, and provide baseline measures for satisfaction improvement programs. This makes the Survey component a strong tool in the CRM toolbox, bringing another dimension of intelligence to the 360-degree view of the customer.

See Also

What Is Oracle Scripting?

Why Use Oracle Scripting?

Who Are the End Users of Oracle Scripting?

1.2 Understanding the Script Author

The Script Author component of Oracle Scripting is a powerful authoring environment of graphical layout tools to create, modify, and deploy scripts. The Script Author is a two-tiered script development tool supported on Windows clients (Windows 95, Windows 98, Windows NT, and Windows 2000 operating systems) for creating scripts. These scripts are deployed to the Oracle Applications database and can be executed in the Java-based Scripting Engine (in an interaction center) or as a survey (running as a JSP page in a Web browser).

This topic group includes the following topics:

How to Obtain the Script Author

What Are the Building Blocks of a Script?

What Are the Minimum Requirements for Any Graph?

See Also

Oracle Scripting Overview

Understanding the Scripting Engine

Understanding the Survey Component

1.2.1 How to Obtain the Script Author

In previous releases of Oracle Applications 11*i*, this software was included on a separate CD for interaction center software. For release 11.5.6, the Script Author software is laid down on the `APPL_TOP` of the Oracle Applications server during a Rapid Install.

Prerequisites

- Oracle Applications 11.5.6 or later must have been installed using a Rapid Install at your enterprise or site.
- The Script Author must be installed on a Windows (95,98, NT, 2000) machine.

Steps

1. To load the Script Author, download the installation package with a Web browser by navigating to the following url:

   ```
   http://<server>:<port>/OA_HTML/download/ies/iesauthr.zip
   ```

   where <server> and <port> are the values for the Apache Web server configured for JTF for your Oracle Applications installation.
2. Decompress the installation files.
3. Install the Script Author using the Oracle Universal Installer.
4. When running the Script Author for the first time, you may be prompted to either load strings from the database or install default. For more information, see Troubleshooting the Script Author > Storing Strings in the Script Author.

See Also

Storing Strings in the Script Author

What Are the Building Blocks of a Script?

What Are the Minimum Requirements for Any Graph?

1.2.2 What Are the Building Blocks of a Script?

This topic group includes the following topics:

- Configurable Objects
- Non-Configurable Objects
- Branches

A Script Author script is made up of configurable and non-configurable objects connected by branches. Objects are processing units that tell the script what to do. Branches are processing units that tell the script where to go.

See Also

How to Obtain the Script Author

What Are the Minimum Requirements for Any Graph?

1.2.2.1 Configurable Objects

There are three *configurable* objects in the Script Author. A configurable object is an object which has properties that you can modify. These objects are:

- Panels
- Groups
- Blocks

See Also

- Non-Configurable Objects
- Branches

1.2.2.1.1 Panels A **panel** object contains the information that is displayed in Oracle Scripting Engine at runtime. Answer controls defined in the panel (using the Script Author) are used to accept information from the user at runtime and advance the flow of the script.

1.2.2.1.2 Groups A **group** object is a container for other objects and branches. It may contain any type of object, including other groups. Groups can be nested as many levels as desired (the only limit is practicality). A group represents a child graph (a subgraph). It must certain requirements (see What Are the Minimum Requirements for Any Graph?). Groups are used to:

- Logically group a section of the script flow (typically, to contain a single function or process)
- Serve as a container for a Shortcut, so that in runtime a Shortcut button can appear in the GUI that will "jump" the user to that group
- Restrict the access of certain or all users to a section of the script at runtime

In addition, when a saved script is imported into a new script, it is imported as a group.

1.2.2.1.3 Blocks A **block** is used for one of two purposes. When used to query, update, or insert information in a database, it is always associated with one or more database tables and contains database connection information. When used as a container for an Application Program Interface (API), it contains a command that references the API.

A block can also be a container for panel objects. The answer controls in the panel objects can be used in block processing.

A block represents a child graph (a subgraph). It must meet certain requirements (see What Are the Minimum Requirements for Any Graph?).

1.2.3 Non-Configurable Objects

There are two *non-configurable* objects in Oracle Scripting Author. These objects are:

- Start node
- Termination node

See Also

- Configurable Objects
- Branches

1.2.3.1 Start Nodes

Every time a graph is created in the Script Author, it contains a Start node. Start nodes are non-configurable. They cannot be explicitly created, nor deleted. They contain no viewable properties. Start nodes simply visually represent the starting point on any graph. They can be moved about the canvas as desired, and must be attached using a branch to the first explicitly created object in the script.

1.2.3.2 Termination Nodes

Termination nodes are required on every graph. They represent the end of the flow of that level of the script in runtime. They are non-configurable, containing no viewable properties. Every graph must contain at least one Termination node, attached with branches to objects on the canvas. A graph may contain two or more Termination nodes, each depicting the end of a flow of a particular path.

Note: The single exception to the requirement for a Termination node on every graph is a graph that contains an Indeterminate branch. See Indeterminate Branch Exceptions in this document.

1.2.4 Branches

A branch is used to connect objects. The type of branch used determines the next object in the script flow. There are four types of branches:

- **Default:** The destination is the designated object.

  If there is more than one branch, this type of branch must be defined as the last-case default if no other branch is valid. Requires no properties to be associated with it in the Script Author.

- **Distinct:** The destination is based on the answer selected in a panel at runtime.

  This branch type requires lookup values to be defined in the panel, and requires values to be assigned to each distinct branch from the branch properties in the Script Author.

- **Conditional:** The destination is based on the evaluation of a Boolean expression.

  This branch type requires the definition of a Boolean expression to be associated with the branch in the Script Author and corresponding Java code to provide the expression to be evaluated at runtime.

- **Indeterminate:** The destination is based on information obtained at runtime.

  This branch type requires the definition of a command to be associated with the branch in the Script Author and corresponding Java code (to provide the expression and destination objects) to be evaluated at runtime. The script can then flow to any sibling object (panel, block or group on the same canvas) or a destination group (using the Shortcut property) on any level of the script.

Branches and Branch Order

Below the Start node, each object placed on the Script Author canvas is typically connected above and below by appropriate branches, ending with the Termination node. Outgoing branches (branches drawn *from* any object) are displayed in the Branch pane that results by clicking the Branch attribute under any object's Property dialog.

The branches are displayed in the order in which they are created. By default, this determines the order in which the branches will be evaluated. Following the procedures discussed in the topic Reordering Branches, this order can be changed after branches are created. Thus, for a panel with multiple branch types, it is crucial to ensure the Default branch is listed last so that other branch types will be evaluated.

See Also

- Configurable Objects
- Non-Configurable Objects

1.2.5 What Are the Minimum Requirements for Any Graph?

A Script Author script is represented as one or more graphs that use visual symbols to represent the three configurable object types—panels, groups, and blocks—connected with the appropriate branch type to control the flow of the script. Each valid graph within a script will also contain a Start node (created on each graph or sub-graph automatically) and at least one Termination node (inserted on the canvas by the script developer). These two non-configurable objects will also be connected with branching as appropriate.

The Shortest Syntactically Correct Script

Theoretically, to be syntactically correct, a script (or any graph on any level of a script) requires only a Start node (created automatically), a Termination node, and default branching between those objects. Of course, such a script does not contain any properties that would be processed or displayed at runtime. Adding any configurable objects includes additional requirements.

Additional Requirements for Panels

Every panel requires at least one "node" or answer to be defined in order for the script to by syntactically correct. The reason for this requirement is that every panel displayed at runtime requires end user interaction to progress the flow of the script. This interaction comes in the form of the end user's response to an answer definition in a panel, or a panel's Continue button.

The panel is the only Script Author object that visible at runtime, displaying panel text, any answers (nodes) that have been defined for the panel, and associated labels for those answers.

Additional Requirements for Groups or Blocks

When an object such as a group or block exists only as a container, it must still adhere to these rules. Thus, if the container object (whether a group or a block) requires no panels or other objects within it, it must still have a Termination node, connected with a Default branch from the Start node.

Furthermore, every object on a graph must contain branching, initiating from the Start node, in order to be processed by the script and pass a syntax check.

Indeterminate Branch Exceptions

Graphs that contain an Indeterminate branch introduce two exceptions:

5. An Indeterminate branch contains an action or expression that must be evaluated in order for the flow to be completed. A graph with an Indeterminate branch need not adhere to the requirement for a Termination node in order to pass a syntax check. While a Termination node is not *required*, it may be used without causing issues.
6. Indeterminate branches may contain Java code to "jump" the user to a particular panel (on the same graph) or group (anywhere in the script). Thus, when using Indeterminate branches you may use panels or groups that are not attached from above with branching. These must still be branched appropriately from that point onward on the graph.

See Also

How to Obtain the Script Author

What Are the Building Blocks of a Script?

1.3 Understanding the Scripting Engine

The Scripting Engine is a Java-based GUI executed from an Oracle Applications session. Using the Scripting Engine, Interaction Center agents are guided through their interactions with customers. The business logic built into the script, including custom Java, PL/SQL, and other commands, guides the agent through the script. As the trained agent moves effortlessly through guided messages represented by script panels, the Oracle Scripting objects embedded in the script control complicated processing. The business logic embedded in the script (such as rules-based branching, data integration, and commands associated with specific objects and events) is evaluated dynamically, along with the responses provided by the agent, progressing or flowing through the script in a logical manner and accomplishing the exchange of information required by the transaction. Oracle Scripting scripts can currently be launched by interaction center agents from within three business applications in the CRM Suite: Oracle TeleService, Oracle TeleSales, and Oracle Collections.

The Scripting Engine GUI is simple to use and promotes rapid agent training. When a script is requested by an agent, the Scripting Engine launches as a Java bean in a separate Oracle Forms window. The window is described in the following sections.

This topic group includes the following topics:

The Panel Display Area

The Progress Area

The Script Information Area

The Shortcut Button Bar

The Disconnect Button

The Toolbar

See Also

Oracle Scripting Overview

Understanding the Script Author

Understanding the Survey Component

1.3.1 The Panel Display Area

The largest visible region is the panel display area, where agents will see any panel text and answer controls for the one or more answer definitions programmed into the script.

The Scripting Engine application supports both a single-panel display and multiple-panel display. Using single-panel mode, only the *current* (active) panel will appear on the screen at any given time. Under multi-panel display mode, the active panel will have focus, and all text and answer controls will appear as designed on the active panel only. Panels previously visited will appear above, clearly unselected (the lettering in a panel without focus is smaller and gray, similar to a menu item that cannot be selected in a typical Windows-based application). At any given time, in multi-panel display mode, the agent can scroll up using a scroll bar located alongside the panel display area to see, and click on, previously displayed panels.

Agent interaction is required for each panel to progress to a subsequent panel. This comes in the form of interacting with the answer control or controls displaying in the current panel. Answer controls supported in the Scripting Engine include familiar answer types that render in HTML forms: buttons, checkboxes, radio buttons, text fields, text areas, and password fields. After interacting appropriately with the one or more answer controls on the panel, the agent must click the **Continue** button. The single exception is the case of a button control. A button answer control, *may* contain a value other than "Continue." Nonetheless, selecting that button will progress you to the next panel.

Note: If the script developer uses the "button" answer control UI type in a panel, that panel must contain only a single answer definition (or "node").

See Also

The Progress Area

The Script Information Area

The Shortcut Button Bar

The Disconnect Button

The Toolbar

1.3.2 The Progress Area

As the agent cycles through panels, each panel subsequently visited will appear in the Progress frame on the right. These are listed by the "label" property for each panel as programmed into the Script Author tool. Should an agent wish to revisit a panel, she can click on the panel label in the Progress area. That panel will again populate the panel display area.

The Scripting Engine application supports both a single-panel display and multiple-panel display. Using single-panel mode, only the *current* (active) panel will appear on the screen at any given time. Under multi-panel display mode, the active panel will have focus, and all text and answer controls will appear as designed on the active panel only. Panels previously visited will appear above, clearly unselected (the lettering in a panel without focus is smaller and gray, similar to a menu item that cannot be selected in a typical Windows-based application). At any given time, in multi-panel display mode, the agent can scroll up using a scroll bar located alongside the panel display area to see, and click on, previously displayed panels. In both display modes, an agent may click on the panel listed in the Progress area to return to a previously displayed panel.

See Also

The Panel Display Area

The Script Information Area

The Shortcut Button Bar

The Disconnect Button

The Toolbar

1.3.3 The Script Information Area

Optionally, information may be associated with a script in the area known as the **Script Information Area**. This is formerly known as the Static Information Panel. The Script Information Area can accommodate up to nine separate pieces of information (text or a timer), including a label for each. These can contain dynamic information using Scripting APIs. Any information in the Script Information Area is associated as a global script property.

See Also

The Panel Display Area

The Progress Area

The Shortcut Button Bar

The Disconnect Button

The Toolbar

1.3.4 The Shortcut Button Bar

Optionally, below the Script Information Area, buttons may appear in the **Shortcut button bar**. These buttons are also associated as a global script property, and can also contain dynamic properties using Scripting APIs. A script may contain Shortcut buttons without information in the Script Information Area, and vice versa, or may not contain information in either of these areas.

See Also

The Panel Display Area

The Progress Area

The Script Information Area

The Disconnect Button

The Toolbar

1.3.5 The Disconnect Button

Optionally, a **Disconnect** button which appears on the bottom right of the Scripting Engine window can be associated with a specific group (a "Wrapup group" to capture information from any individual at the close of the transaction) or it can be enabled to end the script without displaying information. This is programmed using the WrapUpShortcut property of a group in the Script Author tool as part of the script customization.

See Also

The Panel Display Area

The Progress Area

The Script Information Area

The Shortcut Button Bar

The Toolbar

1.3.6 The Toolbar

A toolbar appears at the top of the Scripting Engine window. From here you can navigate back to previously visited panels using the first tool on the toolbar, the **Previous panel** button. If you have backtracked in a script, you may move forward in the same path one panel at a time using the **Next panel** button, or skip to the last panel thus far visited using the **Last panel** button. You may also pop a Web browser using the **Toggle Browser** button (the same browser used to launch Oracle Applications will open in a separate window by default). The **Help** button is not yet implemented.

See Also

The Panel Display Area

The Progress Area

The Script Information Area

The Shortcut Button Bar

The Disconnect Button

1.4 Understanding the Survey Component

The Survey component of Oracle Scripting was previously known as iSurvey. Using this component, an enterprise can easily create and deploy a survey campaign. This campaign would involve polling a population to obtain specific data, typically for subsequent action.

Use of the Survey component of Oracle Scripting is based on a set of predetermined requirements that define a survey campaign. To implement the campaign, you must develop a script to serve as the survey questionnaire and administer a survey campaign.

This topic group includes the following topics:

The Survey Campaign

The Survey Questionnaire

The Survey Admin Console

See Also

Oracle Scripting Overview

Understanding the Script Author

Understanding the Scripting Engine

1.4.1 The Survey Campaign

The Survey component relies on the concept of a *campaign*— a focused effort to achieve a particular goal from a targeted population over a specific period of time for a particular business purpose. The goal of a *survey campaign* is typically for an enterprise to obtain specific data by polling a target market segment or population for subsequent analysis and subsequent action. Typical actions might include the offering of a new product or service, or a change in business processes to improve satisfaction.

Survey campaigns have requirements that must be defined based on the campaign goals (for example: target population, purpose for obtaining the information, which information will be gathered, for what purpose, how long the campaign will be conducted, and in how many separate deployments of the same requirements).

The campaign goals are achieved by two processes: creating a survey questionnaire built with the campaign goals in mind, and in administering the campaign from the enterprise.

See Also

The Survey Questionnaire

The Survey Admin Console

1.4.2 The Survey Questionnaire

The survey questionnaire is a Script Author script created to obtain specific information. Typically it consists of a set of questions to gather feedback, obtain market data, tabulate opinion, measure satisfaction, or otherwise collect willingly provided survey data. End users of the Survey component are customers or prospects viewing a survey questionnaire using any Oracle Applications 11*i*-compliant Web browser. As the individual participating in a survey ("survey respondent") moves effortlessly through each HTML page (corresponding to a single script panel in the Script Author), the objects embedded in the script control complicated processing. The business logic embedded in the survey questionnaire script (such as rules-based branching, data integration, and commands associated with specific objects and events) is evaluated dynamically, along with the respondent's answers.

Fixed or Dynamic Flow

The respondent is guided through the panels of the script either through a rigidly prescribed order, or through a dynamically determined path in the survey. This is determined by the construction of the script, based on the needs of the enterprise or specific survey campaign requirements. Some survey campaigns or enterprises require the same information to be presented to or solicited from every customer in every transaction. Other survey campaigns or enterprises incorporate flexibility into the script, taking advantage of rules-based branching provided by multiple branch types, and using Scripting Blackboard APIs to enable selection and dynamic presentation of appropriate questions based on answers previously provided by the respondent in the survey or by information pulled into the script from database tables. This greatly enhances questionnaire flow, and yields better data and higher response rates.

In the background, the script (and any custom code) runs as a Java Server Pages (JSP) file in an HTML-based execution GUI, typically on the Apache Web server associated with an enterprise applications server and used by Survey administrators.

The script serving as the survey questionnaire must be completed, tested, and deployed to the applications database prior to defining the survey campaign in the Survey Admin console.

Who Creates a Survey Questionnaire?

Based on existing requirements for a specific business purpose, individuals trained in the use of the Script Author component of Oracle Scripting should be used to develop the script that will be viewed by survey respondents. Since surveys are usually a series of simple questions and answers, surveys can in theory be created by relatively non-technical users who have been trained to use the Script Author. However, because Survey is based on the Oracle Scripting product, it also provides very sophisticated functions which can be extended to survey questionnaires in support of survey campaigns that require complex survey solutions. The same additional skill sets that may be required for complex scripts to be executed in the Scripting Engine are applicable to scripts intended for use by the Survey component of Oracle Scripting, including PL/SQL and custom Java, as well as the Scripting-specific use of Constant and Blackboard commands. In practice, therefore, script developers are typically moderately technical functional users with access to other development resources with substantial technical ability and training in the required supporting technologies.

See Also

The Survey Campaign

The Survey Admin Console

1.4.3 The Survey Admin Console

Survey administrators must define the survey campaign requirements in an HTML administrative console in order to get the survey questionnaire to its target segments. The Survey Admin console is a JSP/HTML-based survey campaign administrative GUI accessed from the Oracle eBusiness Center HTML login. In order to log into this console you must be a user with the `Survey Administrator` Oracle Applications responsibility. In this GUI, a survey campaign is created and its dependencies (JSP resources, Script Author script designated as the questionnaire, and so on) are defined.

Additionally, results from a survey (ongoing or completed) can be viewed in various ways from the Survey Admin console. Answers provided by respondents are stored in the Oracle Applications database schema. A survey administrator can view individual responses by accessing the **Response** tab in the GUI. Survey respondent data is also summarized for optimum performance and made available to administrators through a set of preexisting Survey reports under the **Analysis** tab. Additionally, custom reports can be generated from tables in the Oracle Applications schema as desired.

All survey campaign information must be available to the survey administrator, including the global name of the script serving as the survey questionnaire and any predefined JSP resources, prior to the administrator defining a survey campaign and creating its child objects (the cycle and the deployment).

See Also

The Survey Campaign

The Survey Questionnaire

2

Planning Oracle Scripting Projects

Oracle Scripting includes three components: the Script Author, the Scripting Engine, and the Survey component. Any Oracle Scripting project requires a script to be developed using the Script Author. This script can then be executed in either the Scripting Engine (a Java-based GUI used by agents in the interaction center), or as a survey (in an Oracle Applications 11*i*-compliant Web browser as a JSP page). The same script can also be executed in both modes.

Appropriate planning for Scripting projects depends on many factors. Chief among them is the manner of execution of the script—in other words, will the script be executed in a call center or interaction center (as a call guide for an inbound, outbound, or blended environment)? Or will the script be used as the survey questionnaire in a survey campaign to gather information from a targeted population? Different considerations apply, and obviously all considerations are applicable in the case of a script intended to execute in both modes.

This topic group includes the following topics:

Planning Scripting Engine Projects

Planning Oracle Scripting Survey Campaigns

2.1 Planning Scripting Engine Projects

Planning Scripting projects to be executed in the interaction center is typically the task of a consulting group within an enterprise. This involves gathering detailed requirements for building the Script Author script, understanding the business rules, integration requirements, short- and long-term goals of interaction center managers. Thus, the primary aspects of planning for Scripting project managers involve project scoping and discovery of existing, emerging and future requirements.

Note: Many of these aspects are also required of scripts to be executed as surveys. As such, this information may also prove relevant to scripts to be run in a Web browser.

This topic group includes the following topics:

Facets of Scripting-Specific Discovery Process

Bringing Together the Layers

Tools to Aid in Scoping and Discovery

Oracle Scripting Discovery Data Worksheet

Oracle Scripting Discovery Checklist Tool

2.1.1 Facets of Scripting-Specific Discovery Process

There are three main aspects to the Discovery process as it relates directly to Oracle Scripting:

1. Text Layer
2. Logic Layer
3. Technology Layer

Text Layer

The Text Layer refers to the text that is to be read aloud by an agent following a script. In order to appropriately plan a Scripting Engine project, obtain a detailed script or the existing call guide (if one is already in production that will be replaced with Scripting). Ensure you understand fully any changes to an existing script or call guide that the customer wants. It is important to freeze requirements early to ensure deadlines are met.

Logic Layer

The Logic Layer represents a clear understanding of the logic of the desired script, including expected flow, criteria for branching into specific functional areas of a script under specified criteria, ensuring there are no dead ends in flow or inescapable logic loops. In order to create a clear logic layer, it is important that the customer and the consultants understand and map out all business rules covering every eventuality. Oracle Scripting is an excellent tool to assist agents in presenting information logically and consistently, but the tool alone is not a guarantee of

success. A clear understanding of the business requirements is key. In order to satisfy both the customer and the consulting team, detailed flowcharting of the entire script and all its possible branches should be performed. This may require specifically trained business analysts and should be a prerequisite of accepting a scripting engagement, or at the very least, must be completed (and agreed upon by both parties) prior to initiating development.

Note: This point bears repeating: if you do not know all requirements of a script prior to accepting the engagement, there is no clear acceptance criteria to determine when the project is complete. It is to the benefit of enterprises implementing Scripting as much as to the consultants customizing the scripts to have a complete, clear flow and to design and agree upon acceptance criteria based on the Logic Layer.

Constant business requirements analysis is typically required during script development as the Oracle Scripting technology automates the process of an agent interacting with customers, and replaces any previous technology. Scripting has limitations which are easily overcome when the objective is clear, which is why it is important for the enterprise using this tool to be educated in Scripting's approach, its question-and-answer paradigm, and any specific obstacles that are encountered and overcome during development of the initial script. It is in the enterprise's interest to follow the progress of the development of the initial script to be delivered, so interaction center managers and technical staff can begin to appreciate techniques, approaches, strengths, and so on. In this way, subsequent script development (or modifications to a script that is delivered and accepted by the enterprise) is greatly simplified—whether performed by the enterprise staff alone, or with additional consulting support.

Technology Layer

Considerations for the Technology Layer include, but are not limited to, an understanding of what technology choices will suit the needs of the script at hand (and specific functionality with in the script). Discovery for the Technology Layer includes forming choices regarding Script Author objects such as Panels, Groups, Blocks, and appropriate Branches. For example, what technology choice makes for the best, most effective method of text to be delivered by the agent? A single panel? Multiple panels? A group containing logically related questions?

However, the GUI provided by the Script Author has powerful capabilities behind it that greatly extend functionality for the script. Many of these require custom

programming. For example, what is the best way for an agent to obtain a set of information required by the business rules? Should the choice be simply to present all customers with a long string of questions in consecutive panels? Or is it better to perform a PL/SQL query to the customer database to determine what information about the customer is known, and determine which remaining information must be collected by creating complex Java methods that analyze the data placed on the Scripting Blackboard by the query, and route the agent only to the appropriate questions?

The Technology Layer includes consideration of, but is not limited to, the following:

- Creating and implementing Blocks making database calls or using APIs
- Writing PL/SQL procedures
- Writing and storing on the database PL/SQL packages
- Creating custom Java components
- Using APIs to take advantage of Scripting functionality or integrate with other applications
- Creating Shortcuts programmed into Groups on the canvas
- Using Indeterminate branches combined with custom Java methods to perform "jumps" to different functional Groups within the script (uses the Shortcut property of a group, or the panel or block name if the destination object is on a sibling object)
- Populating the global Blackboard with values to be used during execution of the script
- Populating the Blackboard with key value pairs (by answering questions in a script session) and retrieving them using custom Java methods to control the script flow
- Creating custom data tracking in custom databases
- Reusing functionality in existing scripts as templates (importing existing scripts or portions thereof)
- Integrating and modifying Best Practices scripts or surveys

See Also

Bringing Together the Layers

Tools to Aid in Scoping and Discovery

Oracle Scripting Discovery Data Worksheet

Oracle Scripting Discovery Checklist Tool

2.1.2 Bringing Together the Layers

The Text Layer is typically the first to be considered, and while it may be most important to the enterprise implementing the script, it is the least important implementation consideration—providing that specific text is supplied to those building the script, or that there is some flexibility in modifying the required text, for example when the Scripting approach is best served by breaking up a question into two parts.

The Text Layer can provide certain challenges. For example, in some cases a proposed call guide or script may have undergone legal review prior to coding with the Script Author. In the process of building the flowchart representing the final script or building the script itself, changes may be necessitated due to gaps in logic or other reasons. This may then require further approval for any changes, by the legal authority or department of an enterprise. This can impact a delivery schedule if not planned for in advance.

The Technology Layer aspect of the Discovery process is by far the most time-consuming to implement. Nonetheless, success in determining requirements for the Technology Layer are strongly dependent on flowcharting being provided as discussed in the Logic Layer section above. Without detailed business analysis, Discovery for the Technology Layer will be ineffective and implementing this layer more difficult, time- and resource-consuming.

The Technology Layer includes many hidden tasks and covers integration points with Scripting and other applications. Full analysis of the Logic and Technology layers may indicate that a project is more manageable and more likely to serve the needs of the enterprise when broken into a phased approach. In this way, enterprises can benefit enormously from benchmarking efforts required in an initial phase, and determine realistic assessments for subsequent phases to add future Scripting functionality. In the meantime, in early phases the enterprise can have the interaction center up and running using Oracle Scripting, taking the opportunity to train agents and staff while experiencing the advantages of the efficiencies and return on investment of automated scripting.

See Also

Facets of Scripting-Specific Discovery Process

Tools to Aid in Scoping and Discovery

Oracle Scripting Discovery Data Worksheet

Oracle Scripting Discovery Checklist Tool

2.1.3 Tools to Aid in Scoping and Discovery

Appropriate planning is crucial to the success of a Scripting implementation. Tools to assist in this planning include the Oracle Application Implementation Methodology (AIM), a methodology that follows the full life cycle of a custom software implementation. Even if Oracle Scripting is only part of a broader implementation of Oracle Applications, it is recommended to plan for it separately, using a separate Scope, Objectives, and Approach (SOA) document (AIM CR.010) to help plan the resources required for this portion of the implementation.

Part of a consultant's skill set is the ability to perform a thorough Discovery process to fully understand an enterprise's requirements, existing infrastructure and environment, and long-term needs. Included below are some tools to aid in scoping a potential Scripting project. These include the Oracle Scripting Discovery Data Worksheet, which helps to gather information specific to Scripting that should be obtained for the potential Scripting project in general. The other tool provided here is the Oracle Scripting Discovery Checklist Tool, which should be filled out for each distinct functionality expected to be created in a script (for example, each group on the canvas).

Using these as aids to the Discovery process can help gather information that will be crucial to appropriate planning, allocation of resources, and scoping of the effort in general.

See Also

Facets of Scripting-Specific Discovery Process

Bringing Together the Layers

Oracle Scripting Discovery Data Worksheet

Oracle Scripting Discovery Checklist Tool

Oracle Scripting Implementation Guide, including section "AIM's Role in Scripting Implementation Planning "

2.1.4 Oracle Scripting Discovery Data Worksheet

The following worksheet covers an entire Scripting engagement for which you are attempting to gauge the scope in order to properly provide a time estimate and to assess the skill and resource requirements.

1. For which of the following purposes does the customer/prospect intend to use Oracle Scripting?

   ☐ Pure scripting purposes ☐ Desktop integration tool ☐ Both ☐ Undetermined

   If the customer wishes to use Oracle Scripting for desktop integration, describe the requirements in general terms:

2. Characterize the Scripting required:

   ☐ Inbound Call Guide ☐ Outbound Call Guide ☐ Both

   ☐ Guided Selling ☐ Undetermined

3. Does the customer require call guides that support languages other than English?

   ☐ Yes ☐ No If yes, identify the language(s):

4. How many call guides or individual scripts does the customer want Oracle (or Oracle partners) to develop? If undetermined, please indicate.

5. Does the customer currently have existing scripts that they wish to convert to Oracle Scripting?

   ☐ Yes ☐ No If yes, how many?

If appropriate, attach the existing scripts to this document and indicate form of existing scripts:

Attached as: ☐ Hardcopy ☐ Electronic File ☐ Not Available

6. What call guide or script currently exists from which Oracle Scripting scripts will be developed?

   ☐ Narrative ☐ Flowcharting ☐ System Requirements Document (SRD) ☐ Existing scripts

7. Is Computer-Telephony Integration (CTI) intended for use in the facility or facilities?

   ☐ Yes ☐ No

   Does the customer have a need to display different call guides using CTI?

   ☐ Yes ☐ No

8. If so, what will be the criteria?

   ☐ Dialed Number Information Service (DNIS) ☐ Automatic Number Identification (ANI)

   ☐ Unique ID (UUID) ☐ Account type ☐ IVR information ☐ Other

   If Interactive Voice Response (IVR) unit information required, is IVR in place? ☐ Yes ☐ No

9. Does CTI currently exist? ☐ Yes ☐ No Describe:

10. Does the customer plan to create and maintain its own call guides or scripts? ☐ Yes ☐ No

11. **Gap Analysis.** Describe in detail any modifications that Oracle (or partner) must make to the generic version of Oracle Scripting in order to satisfy the needs of the customer:

See Also

Facets of Scripting-Specific Discovery Process

Bringing Together the Layers

Tools to Aid in Scoping and Discovery

Oracle Scripting Discovery Checklist Tool

2.1.5 Oracle Scripting Discovery Checklist Tool

First, qualify each call guide or script required using the following checklist. Then, again using the checklist template below, break down the requirements for *each distinct functionality within each script* (as represented by Groups, or functions separated by agent roles) as the Discovery and Scoping process continues.

Be careful to label each main script checklist, and its dependent checklists, appropriately.

Discovery Checklist

Checklist ID _____ ☐ Main script checklist ☐ Script functionality breakdown checklist

Number of Panels:		***Development Level Assessment***
☐ Very Small:	1 to 10 panels	☐ Very Simple
☐ Small:	11 to 25 panels	☐ Simple
☐ Medium:	25 to 40 panels	☐ Lower Intermediate
☐ Medium to Large:	40 to 100 panels	☐ Upper Intermediate
☐ Large:	100 to 200 panels	☐ Complex
☐ Very Large:	Over 200 panels	☐ Very Complexs

Scripting Elements Required:

☐ Mostly Panels ☐ Mostly Panels and Groups ☐ Panels, Groups, and Blocks

Branching Required: (Check all that apply)

☐ Default (order or branching of panels does not change based on panel answers, e.g., a survey)
☐ Conditional (boolean logic)
☐ Distinct (branching based on answers, with all paths known)
☐ Indeterminate (no 1:1 relationship between answers and path before this interaction)
☐ All branch types

Branching info Comments:

Database Integration: (Check all that apply)

☐ No database will be used (skip to next)
☐ Database integration required
☐ Custom tables required
☐ Schema(s) available (if so, attach)

PL/SQL: (Check all that apply)

☐ No PL/SQL will be used (skip to next)

Amount of PL/SQL commands required:

☐ 1 or 2 ☐ Several ☐ Many

Are the PL/SQL statements for IES tables or custom tables? ☐ IES ☐ Custom ☐ Both

Technology Required:

☐ HTML	Currently exists?	☐ Y	☐ N	Amount:	☐ 1 or 2	☐ Several	☐ Many (List)
☐ Hyperlinks	Currently exists?	☐ Y	☐ N	Amount:	☐ 1 or 2	☐ Several	☐ Many (List)
☐ Graphics	Currently exists?	☐ Y	☐ N	Amount:	☐ 1 or 2	☐ Several	☐ Many (List)

Script Author Commands:

Custom PL/SQL	Currently exists?	☐ Y	☐ N	Amount:	☐ 1 or 2	☐ Several	☐ Many (List)
Custom Java	Currently exists?	☐ Y	☐ N	Amount:	☐ 1 or 2	☐ Several	☐ Many (List)
Blackboard	Currently exists?	☐ Y	☐ N	Amount:	☐ 1 or 2	☐ Several	☐ Many (List)
Other (list)	Currently exists?	☐ Y	☐ N	Amount:	☐ 1 or 2	☐ Several	☐ Many (List)

Integration Required:

☐ Yes ☐ No With: ☐ Forms ☐ Reports ☐ HR ☐ Financials ☐ CRM ☐ Other (list:)

Requires jumping (using indeterminate branches and Java) to other groups? ☐ Yes ☐ No

CONSULTING TIME ESTIMATE **Build:** **Test:**

See Also

Facets of Scripting-Specific Discovery Process

Bringing Together the Layers

Tools to Aid in Scoping and Discovery

Oracle Scripting Discovery Data Worksheet

2.2 Planning Oracle Scripting Survey Campaigns

The main purpose for using the Survey component of Oracle Scripting is to obtain data (via a clearly defined survey questionnaire) from a population over a specific period of time for a particular business purpose. The Survey component of Oracle Scripting enables an enterprise to rapidly solicit and receive such data at low cost, using a Script Author script as the survey questionnaire. This data is then typically collected and analyzed to serve the enterprise by improving products or processes or otherwise allowing the enterprise to be responsive. The Survey component of Oracle Scripting release 11.5.6 includes the ability to create ad-hoc reports from the Survey Admin console to aid in data analysis.

This data may be solicited from a targeted population (using predefined lists) or from a general population. Targeted populations may include customers or prospective customers, but also partners or affiliates, a company's own employees, and so forth. Business purposes may in some cases be served by responses from a random population. At other times business purposes may be best served by receiving feedback from an identified population. Both are supported by the Survey component of Oracle Scripting.

Survey Business Process Flow

The ten steps depicted in the flowchart are crucial to understanding the administration and use of the Survey component of Oracle Scripting. The process flow steps, as enumerated in Figure 2–1 and described briefly below, are required to plan and execute a survey campaign. The *planning aspects* of each process flow step are addressed in detail in each section described below.

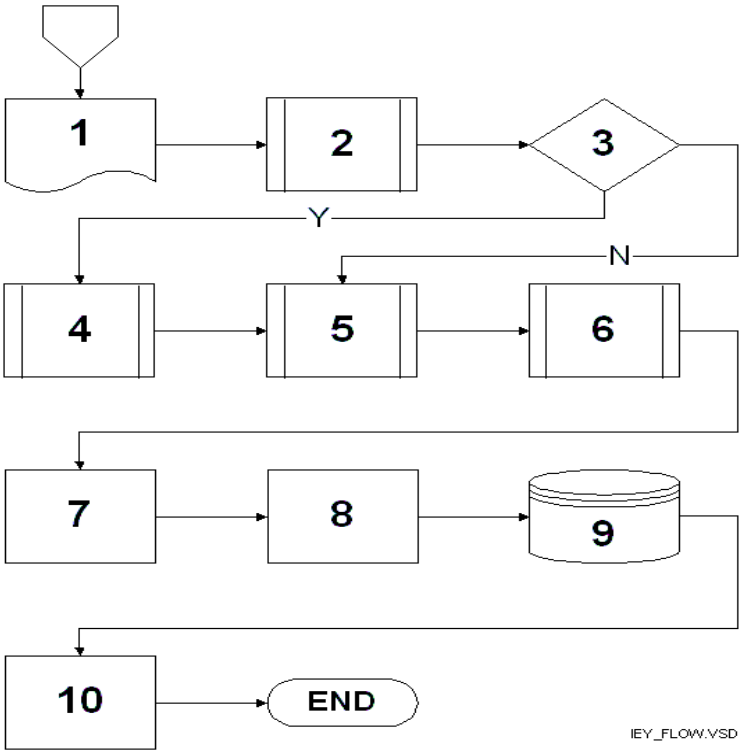

Figure 2–1 Survey Business Process Flow.

1. Define survey campaign requirements.

Obtain requirements for all aspects of the survey campaign (survey campaign, cycle, and deployment-specific requirements). These should be documented using appropriate methodology; for example, if using the Application Implementation Methodology (AIM), obtain Scope, Objectives and Approach document and Business Requirements documents such as BR.100.

2. Create and deploy script.

Create script for survey questionnaire based on documented requirements (captured in step 1) and deploy to appropriate applications database.

3. Decision: List-based campaign or non-list-based campaign?

Is survey campaign List-based? If "NO," skip to step 5. If yes, continue to step 4.

4. Generate lists (for list-based survey campaigns).

Generate, create or import lists in the Survey Admin console (one list per deployment, if multiple deployments are required). Note that the same list can be used for multiple deployments (and also for multiple survey campaigns). Generating lists requires the implementation of list management portions of Oracle Marketing Online. Note that successful delivery of invitations (and reminders) to participate in survey campaigns requires fully implemented and configured Oracle One-to-One Fulfillment and a Fulfillment Engine.

5. Administer survey resources and survey campaign details.

Create and upload (or modify existing) JSP survey resources to be displayed when using survey campaigns. Then define these survey resources in the Survey Admin console. You may establish default resources per user (survey administrator) if desired. The preceding comprise all survey resources administration steps. Subsequently, administer the survey campaign details in the Survey Admin console. This includes creating a survey campaign, and defining its subordinate cycle or cycles.

6. Define deployments.

In the Survey Admin console, define deployment information for one or more deployments subordinate to each cycle in a survey campaign, based on documented requirements. For list-based campaigns, this will include lists, invitations, and (if included in requirements) reminders.

7. Deploy survey campaign.

In the Survey Admin console, set deployments to Active status. This makes the survey campaign available to respondents immediately (or, for list-based deployments, as soon as the invitations are delivered by the Fulfillment Engine).

8. Monitor ongoing results.

Monitor survey campaign. From the **Responses** tab of the Survey Admin console, individual responses can be monitored from the moment a deployment becomes active. Each question and the response selected is displayed per respondent.

9. Collect information in Oracle RDBMS.

As respondents complete survey questionnaires, information is collected in Oracle RDBMS. No action is required to collect this data.

10. Report survey results.

Generate ad-hoc reports from the **Analysis** tab of the Survey Admin console, or use SQL-based tools to generate custom reports from the RDBMS.

Planning

This topic contains the following topic groups:

Gathering All Survey Campaign Requirements

Requirements for Creating and Deploying a Survey Questionnaire

Requirements for Determining If Survey Campaign Is List-Based

Requirements for Generating Lists

Requirements for Administering Survey Resources and Survey Campaign and Cycle Details

Requirements for Defining Deployments

Requirements for Deploying Survey Campaigns

Monitoring Survey Results

Collecting Survey Results in Oracle RDBMS

Requirements for Reporting Survey Campaign Deployment Results

2.2.1 Gathering All Survey Campaign Requirements

Prior to creating a script to use as the survey questionnaire or to administering survey campaign data in the Survey Admin console, a survey campaign must be planned, from top-level strategic aspects down to a detailed level. This includes gathering the requirements for all aspects of the survey campaign, from top-level goals to the names of cycles and deployments and start and end dates of each deployment. Step 1 of the business process flow includes the determination of information that is required for all other steps in the business process flow.

See Also

Requirements for Creating and Deploying a Survey Questionnaire

Requirements for Determining If Survey Campaign Is List-Based

Requirements for Generating Lists

Requirements for Administering Survey Resources and Survey Campaign and Cycle Details

Requirements for Defining Deployments

Requirements for Deploying Survey Campaigns

Monitoring Survey Results

Collecting Survey Results in Oracle RDBMS

Requirements for Reporting Survey Campaign Deployment Results

2.2.1.1 Planning Requirements for All Survey Campaigns

Substantial information must be gathered in the planning stage prior to proceeding to step 2 of the business process flow or beyond. If you are using AIM in your implementation, this data would be identified in various AIM documents.

The following information is required for *all survey campaigns:*

- The goals of a survey campaign must be clearly documented.
- The target population (if any) must be identified.
- The method of obtaining survey data (the respondent entry point) must be determined. For more information, see the section Survey Respondent Entry Points.
- The requirement for list-based or non-list-based deployments must be indicated. Non-list-based deployments have substantially fewer requirements.
- Survey questionnaire requirements must be obtained to a detailed level. The manner in which data will be collected must be defined, including data type, collection method, sequencing and flow.
  - What type of data will be collected from respondents?
  - How will this data be used or evaluated?
  - Will branching logic will be employed? Or will the same questions be asked of all respondents, regardless of previous answers supplied by respondent?
  - What types of technologies (Java, PL/SQL, Oracle Forms, and so on) will be included in the survey questionnaire script?

    - What level of skill is required to incorporate the appropriate technology into the script?
- In order to ensure that the business goals of the survey campaign will be met, a detailed flowchart and Script test plan must be planned for.
- The survey questionnaire Script name (the name designated in the Script Author file properties) must be identified.
- The layout, configuration and composition of survey resources (Header, Error Page, and Final Page displayed to survey respondents) must be identified.
- Number of cycles required must be identified, including business rules for each cycle.
- Number of deployments per cycle must be identified.
- Deployment name for each deployment must be identified.
- Business requirement for anonymous or non-anonymous cycles must be determined. See the section Reminders and the Anonymous Flag.
- Minimum response percentage for survey cycles must be identified.
- Enterprise Web server and port must be identified.
- Deployment date and time must be identified per deployment.
- Deployment response end date and time must be identified per deployment.

2.2.1.2 Additional Planning Requirements for List-Based Survey Campaigns

Additionally, for *list-based survey campaigns,* the following is required:

- Oracle Marketing Online implementation prerequisites must be met. List management portions of OMO must be implemented to make list management features of the Survey Admin console available during step 4 and subsequent steps of the business process flow.
- Survey campaign information must be identified down to a deployment level, including list members.
- List member requirements must be identified for each deployment.
- Oracle One-to-One Fulfillment implementation prerequisites must be met. Features that must be available include the ability to create and upload master documents, create queries, and create templates. This requires Fulfillment implementation, Fulfillment server configuration, and a running Fulfillment Engine.

- Invitation master document layout and content must be determined. These will be created and uploaded when defining deployments (step 6).
- E-mail invitation message subject must be identified.
- Business rules regarding maximum number of responses per respondent must be identified. This is typically set to one (the default value), which ensures a list member can take a survey only one time.
- Requirement for reminders must have been determined.
- Reminder master document layout and content must be determined if reminders are to be employed.
- The number of reminders and interval between (in days) must be identified if reminders are to be employed.

2.2.1.3 Methodology

Survey campaign requirements and detailed information should be collected and documented using a consistent methodology.

Application Implementation Method

Oracle customers, partners and consultants have access to the Application Implementation Method Advantage (AIM). AIM is a set of pre-packaged approaches for implementing Oracle Applications, developed by Oracle's Applications Global Service Line group. There are two subsets (AIM Advantage and AIM FastForward). Each is a proven, scalable toolkit for implementing Oracle Applications.

AIM is broken into six phases to cover the full software life cycle: Definition, Operations Analysis, Solution Design, Build, Transition, and Production.

For the various phases, AIM provides macro-driven document templates to address numerous processes, which typically span more than one phase. Each of these processes is described by a two-letter acronym:

Table 2–1 AIM Processes and Two-Letter Process Designations

Acronym	AIM Process
CR	Customer Requirements (Project Management)
BP	Business Process Architecture
BR	Business Requirements Definition

Table 2–1 AIM Processes and Two-Letter Process Designations

Acronym	AIM Process
RD	Business Requirements Mapping
TA	Application and Technical Architecture
MD	Module Design and Build
CV	Data Conversion
DO	Documentation
TE	Business System Testing
PT	Performance Testing
AP	Adoption and Learning
PM	Production Migration

Documents generated for each AIM process are identified by the process acronym and a number. For example, the top-level document describing the scope, objectives, and requirements of a project (the main customer requirements) is referred to as a CR.010. You may see specific documents within the AIM methodology referenced in various Oracle documentation.

Note that AIM is *not* a requirement for customer implementations. It is one example of a proven methodology. While planning is required in order to execute an implementation and the subsequent use and administration of a system successfully, *any* effective methodology will suffice.

2.2.2 Requirements for Creating and Deploying a Survey Questionnaire

A survey campaign has a one-to-one correspondence with a single survey questionnaire. In other words, each survey campaign employs only one script. These survey questionnaire scripts can only be created, modified, and deployed from the Script Author.

Part of documenting the requirements for a survey campaign is the detailed definition of the requirements for the survey questionnaire script. This includes information such as specific questions, data format of responses, validation requirements, target audience, branching logic requirements, supporting technologies, questionnaire sequencing and flow. In a best-case scenario, you will have a detailed flow chart depicting the flow of the script, branching, and so forth. You should also have a detailed script test plan to ensure the script, as designed, meets the requirements of the survey campaign. (For more information, see the

section Planning Requirements For All Survey Campaigns.) Before creating the script you will need a full understanding of the campaign and enterprise business rules, dependencies, and any integration requirements.

Following a gathering of the requirements, you will need the following to create, modify, and deploy the survey questionnaire script:

- Appropriate Network, Security, Hardware, and Software Resources
- Trained Script Developers
- Trained Java, PL/SQL, Oracle Forms, and API Programmers
- Trained Oracle Applications and Database Administrators
- Detailed Script-Specific Information
- Environment-Specific Information
- Detailed Survey Questionnaire Requirements

See Also

Gathering All Survey Campaign Requirements

Requirements for Determining If Survey Campaign Is List-Based

Requirements for Generating Lists

Requirements for Administering Survey Resources and Survey Campaign and Cycle Details

Requirements for Defining Deployments

Requirements for Deploying Survey Campaigns

Monitoring Survey Results

Collecting Survey Results in Oracle RDBMS

Requirements for Reporting Survey Campaign Deployment Results

2.2.2.1 Appropriate Network, Security, Hardware, and Software Resources

The Script Author component of Oracle Scripting is required to build scripts to serve as the survey questionnaire. This requires a networked script development workstation. Network, security, hardware and software resources are indicated below. For more information regarding what is required and recommended for use as a script development workstation, refer to the section entitled “Using Oracle Scripting.”

Network

Script Author workstation must be on a network and able to connect with the applications database server in order to deploy scripts.

Security

- The Script Author uses the thin JDBC driver to communicate with the database. As of release 11.5.6, the Apache mid-tier architecture enables the Scripting Engine to be executed through a firewall using the HyperText Transfer Protocol (HTTP).
- If the enterprise uses HTTPS (HyperText Transfer Protocol—Secure), then Oracle Scripting must be configured for HTTPS also.
- If the enterprise uses proxy servers, then Oracle Scripting must be configured for the proxy servers also.
- The Network access setting in the **Basic** tab of the JInititator Control Panel should be set to `Applet Host`.

> **Note:** If using the Caching Architecture of Oracle Scripting only (supported *only* for pre-11.5.6 implementations), *and* if the database and applications servers are on two different physical hosts, the Network access setting in the **Basic** tab of the JInititator Control Panel should be set to `Unrestricted`.

Appropriately configured security settings are the responsibility of Web server administrators at the enterprise.

Hardware

Hardware required to create and deploy a survey questionnaire script (using the Script Author) follows the Oracle CRM hardware requirement standards. Any workstation in your enterprise that meets the requirements for running Forms-based applications is suitable.

It is from this workstation that the script will be deployed to the applications database. The script can be created on this workstation or created on another workstation with the Script Author, and copied to the workstation within the enterprise firewall to deploy.

Software

Software for Developing Scripts:

- **Operating system:** Microsoft Windows NT, 95, 98, or 2000.
- Script Author component of Oracle Scripting
- Oracle CRM 11*i*-compliant Web browser (see Oracle*MetaLink* for description of the latest current certifications)

Software for Additional Development tasks:

Additional software required for Java development:

- Java IDE such as Oracle JDeveloper
- Java Development Kit (JDK)

Additional software required for database and PL/SQL development:

- Appropriately configured TNSNAMES.ORA and SQLNET.ORA files
- SQL tools such as SQL Plus or SQL Worksheet
- Oracle Client (recommended)

See Also

- Trained Script Developers
- Trained Java, PL/SQL, Oracle Forms, and API Programmers
- Trained Oracle Applications and Database Administrators
- Detailed Script-Specific Information
- Environment-Specific Information
- Detailed Survey Questionnaire Requirements

2.2.2.2 Trained Script Developers

Since survey questionnaires are often a series of simple questions and answers, the questionnaire script can be created by relatively non-technical users trained in the use of the Script Author. This is particularly true with survey questionnaires employing primarily sequential logic. However, because the Survey component is based on the Oracle Scripting product, it also provides the ability to include very sophisticated functions in the survey script, if required. Therefore, in order to create adequate survey scripts, you must have received training in using the Script Author

tool. Based on the complexity of the script in question, you must also be knowledgeable in the various supporting technologies, or have access to resources with such knowledge. The selection of supporting technologies required depends on specific requirements for each survey campaign, as well as the degree to which Scripting-supported technologies will be utilized. These technologies can include custom Java and application of APIs, PL/SQL, Oracle Forms, or Scripting-specific commands such as Constant and Blackboard commands. For more information, see Trained Java, PL/SQL, Oracle Forms, and API Programmers below.

See Also

- Appropriate Network, Security, Hardware, and Software Resources
- Trained Java, PL/SQL, Oracle Forms, and API Programmers
- Trained Oracle Applications and Database Administrators
- Detailed Script-Specific Information
- Environment-Specific Information
- Detailed Survey Questionnaire Requirements

2.2.2.3 Trained Java, PL/SQL, Oracle Forms, and API Programmers

Enterprises that intend to leverage the full capabilities of Oracle Scripting in survey campaigns may require the services of highly trained Java developers with experience customizing application program interfaces (APIs), PL/SQL programmers, developers experienced in accessing and working with Oracle Forms, logic and flow experts, and so forth.

See Also

- Appropriate Network, Security, Hardware, and Software Resources
- Trained Script Developers
- Trained Oracle Applications and Database Administrators
- Detailed Script-Specific Information
- Environment-Specific Information
- Detailed Survey Questionnaire Requirements

2.2.2.4 Trained Oracle Applications and Database Administrators

Use of custom Java will also require deployment of custom code to the enterprise server and modification of the `jserv.properties` file to reference the class path of any custom Java appropriately. This requirement replaces the need to configure the `appsweb.cfg` configuration file in the Caching Architecture for Oracle Scripting. This requires experienced Oracle Applications administrators and will be required any time new JAR files are created in support of a script requiring custom Java classes.

Any custom PL/SQL packages for use with the survey campaign must be appropriately built, deployed to the enterprise database, and tested. Deploying PL/SQL packages to the enterprise's applications database may require database administrator (DBA) privileges.

See Also

- Appropriate Network, Security, Hardware, and Software Resources
- Trained Script Developers
- Trained Java, PL/SQL, Oracle Forms, and API Programmers
- Detailed Script-Specific Information
- Environment-Specific Information
- Detailed Survey Questionnaire Requirements

2.2.2.5 Detailed Script-Specific Information

To develop a script that meets the survey campaign requirements, individuals creating scripts or supporting code will need access to existing flow charts or other documented script-specific requirements. They must also be apprised of any dependencies, business rules, and predefined integration requirements. Upon completing development, script developers will need access to any existing test plans ensuring survey questionnaire requirements have been met.

From an AIM perspective, in order to achieve acceptance of the script it must meet the requirements of the system design—as represented by MD.070 or MD.080—and any and all test documents—as represented typically by the TE.040.

See Also

- Appropriate Network, Security, Hardware, and Software Resources
- Trained Script Developers

- Trained Java, PL/SQL, Oracle Forms, and API Programmers
- Trained Oracle Applications and Database Administrators
- Environment-Specific Information
- Detailed Survey Questionnaire Requirements

2.2.2.6 Environment-Specific Information

Scripts must be deployed to the applications database from a workstation within the enterprise firewall. The parameters named below are unique to each environment, as established and maintained by the systems administrator (sysadmin) and the database administrator (DBA). To deploy a script you need:

- Oracle Applications database server host name
- TNS Port number
- Database SID
- `apps` username and password

For more information, see the section Using the Script Author > Deploying the Script in the *Script Author Online Help*.

See Also

- Appropriate Network, Security, Hardware, and Software Resources
- Trained Script Developers
- Trained Java, PL/SQL, Oracle Forms, and API Programmers
- Trained Oracle Applications and Database Administrators
- Detailed Script-Specific Information
- Detailed Survey Questionnaire Requirements

2.2.2.7 Detailed Survey Questionnaire Requirements

Trained individuals use the Script Author tool to visually lay out the flow of a script based on the script requirements. All of the requirements must be made known to the developer of the script to create an effective survey questionnaire. The information in the table below includes the type of information required to successfully create a script that meets the survey campaign requirements. For any script, other information may be required that is not listed below.

Table 2–2

Panel text:	Text that appears at the top of each HTML page. This includes any information assigned to the "Label for reporting" field when defining a panel answer.
Lookup values:	Predefined answer choices.
Constant commands:	Used to provide *or change* an answer default. In the HTML GUI, the first selection in the range of lookup values becomes the default choice, and checkboxes default to checked or "true." (This is different in the Java GUI.) Use a constant command to provide a different default value for a dropdown, radio button, text field or text area, or to deselect a checkbox as the default.
Validation ranges or requirements:	Validation can be enforced in an answer control. This requires associating a Java method with an answer in the data dictionary and must be explicitly programmed. The requirement for validation on a particular answer—and the range of valid answers, if part of the validation routine—must be provided to the developer of the script in advance.
Branching logic:	If a survey questionnaire is appropriately planned, branching logic can be clearly indicated in a flowchart.
PL/SQL packages:	PL/SQL commands that are loaded in the applications database can be referenced from a script. All such commands must be identified prior to development of the script.
Database table field names and locations:	If tables are referenced from a script, the precise table names and locations must be made available to the individual building that portion of the script.

2.2.3 Requirements for Determining If Survey Campaign Is List-Based

The third step in the Survey business process flow is a decision block, indicating the requirement to determine if the survey campaign is list-based or non-list-based. This determination affects the remaining process flow in terms of order of steps and complexity of campaign setup and administration.

As mentioned in the section Survey Respondent Entry Points, a survey campaign may have an existing list comprised of members of a targeted population. For a list-based survey campaign, this list is created using Oracle Marketing Online capabilities. List-based survey campaigns obtain survey feedback by inviting list members to participate in the survey. This *invitation* is an Oracle One-to-One

Fulfillment document that is sent to list members by electronic mail using Fulfillment capabilities. The same list may be used to send *reminders*, an e-mail message typically reinforcing the deadline for list member participation. List members that respond by taking a survey are the survey respondents.

2.2.3.1 List-Based Campaign = NO

If the ability to execute list-based survey campaigns in an enterprise is not required, step 4 of the process flow (generate lists) is skipped and step 6 (define deployments) is substantially simplified. Prerequisites such as implementation of Fulfillment and OMO are also precluded. The abilities you sacrifice by choosing a *non*-list-based survey campaign include the ability to invite members of a list to participate in a survey, the ability to track survey respondents by name or ID number, the ability to send reminders, or the ability to execute subsequent deployments that reaches the same precise audience.

2.2.3.2 List-Based Campaign = YES

When planning to implement or use the Survey component of Oracle Scripting with list-based survey campaigns, the business process flow requires list generation as the next step. For more information refer to Oracle Marketing Online product documentation.

See Also

Gathering All Survey Campaign Requirements

Requirements for Creating and Deploying a Survey Questionnaire

Requirements for Generating Lists

Requirements for Administering Survey Resources and Survey Campaign and Cycle Details

Requirements for Defining Deployments

Requirements for Deploying Survey Campaigns

Monitoring Survey Results

Collecting Survey Results in Oracle RDBMS

Requirements for Reporting Survey Campaign Deployment Results

2.2.4 Requirements for Generating Lists

Step 4 of the Survey process flow is the generation of lists using Oracle Marketing Online functionality. These lists are used to send invitations and reminders via e-mail to list members, encouraging them to click the included URL and participate in a survey.

See Also

Gathering All Survey Campaign Requirements

Requirements for Creating and Deploying a Survey Questionnaire

Requirements for Determining If Survey Campaign Is List-Based

Requirements for Administering Survey Resources and Survey Campaign and Cycle Details

Requirements for Defining Deployments

Requirements for Deploying Survey Campaigns

Monitoring Survey Results

Collecting Survey Results in Oracle RDBMS

Requirements for Reporting Survey Campaign Deployment Results

2.2.4.1 Additional Implementation Requirements for List-Based Survey Campaigns

Additional implementation requirements for enterprises using list-based survey campaigns include OMO and Fulfillment steps.

OMO

Implementation of list management aspects of OMO are required. Once these prerequisites are met:

- OMO can be used to create and administer lists for use with surveys.
- Lists can be generated (OMO functionality) from within the Survey Admin console.

See Also

See *Oracle Marketing Online Implementation Guide* for more information on implementing OMO. See *Oracle Marketing Online Concepts and Procedures Manual* for more information on working with and generating OMO lists.

Additional Fulfillment implementation and configuration requirements are prerequisite for executing list-based survey campaigns. This includes:

- Implementation and configuration of Fulfillment Server, including:
  - An identified outgoing e-mail server
  - An associated survey group
  - Survey administrator users added to Fulfillment server group
- A functioning Fulfillment Engine

Once these prerequisites are met, Fulfillment can take OMO lists and merge data from those lists into survey campaign-specific invitations and reminders and send these out through the identified e-mail server. These steps must be accomplished by a Fulfillment Administrator.

2.2.5 Requirements for Administering Survey Resources and Survey Campaign and Cycle Details

After obtaining requirements (step 1), creating and deploying a script (step 2), determining whether lists are required (step 3) and generating lists if appropriate (step 4), you are ready to administer survey resources and administer survey campaign information. Survey resources are JSP-format files that display when a respondent takes a survey. Chief among these is a header which appears on each HTML page (each panel in the script appears as a single HTML page). The other resources are an error page that results in the even of a Java stack error when processing the survey, and a final page that appears after the last panel in the survey. For planning purposes, you must determine the requirements for these resources (what they must look like, if they already exist, whether the JSP-format page includes any live JSP elements or is simple HTML, and whether default resources should be designated for each survey administrator or for the site). This information is entered by a Survey administrator in the Survey Admin console, accessed from the Oracle eBusiness Suite login.

This step in the flow also includes the creation of a survey campaign and its child object, the cycle. Both can be created from the **Survey Campaign** sub-tab. A survey campaign may contain more than one cycle. Cycles can also be created from the **Cycle** sub-tab of the **Survey Campaign** tab.

The order of steps described here is not arbitrary. Even though survey resources are administered from the fourth (**Resources**) sub-tab of the **Survey Campaign** tab, they must be in place *prior* to creating a survey campaign in the first (**Survey Campaign**) sub-tab of the **Survey Campaign** tab, since the identification of

resources is survey campaign-specific and must occur before you can save a new survey campaign.

> **Tip:** The order of steps described here is not arbitrary. Even though survey resources are administered from the fourth (**Resources**) sub-tab of the **Survey Campaign** tab, they must be in place *prior* to creating a survey campaign in the first (**Survey Campaign**) sub-tab of the **Survey Campaign** tab, since the identification of resources is survey campaign-specific and must occur before you can save a new survey campaign.

For planning purposes, the following types of information are required:

Survey Campaign-Level Information:

- Script name
- Survey campaign name
- Survey resources:
  - Defaults from profile (must be established in advance) or campaign-specific?
  - Names for Header, Error, and Final Page survey resources (must be established in advance)
  - Cycle-Level Information
- Number of cycles required
- Cycle names
- Requirement for anonymous flag
- Minimum response percentage

See Also

Gathering All Survey Campaign Requirements

Requirements for Creating and Deploying a Survey Questionnaire

Requirements for Determining If Survey Campaign Is List-Based

Requirements for Generating Lists

Requirements for Defining Deployments

Requirements for Deploying Survey Campaigns

Monitoring Survey Results

Collecting Survey Results in Oracle RDBMS

Requirements for Reporting Survey Campaign Deployment Results

2.2.6 Requirements for Defining Deployments

Deployments are associated with a specific cycle, which in turn is associated with a single survey campaign. On the same HTML page, information specific to list-based survey campaigns is also required.

> **Note:** If a survey campaign is non-list-based, all list-related information on the **Create Deployment** page should be left blank.

The information you will need in order to define a deployment includes:

Deployment-Specific Information

- Survey under which this deployment is associated.
- Cycle under which this deployment is associated.
- Deployment name for each deployment.
- Deploy date and time
- Response end date and time
- Enterprise Web server URL and port

List-Specific Information

- List name
- Invitation template name
- Reminder template name
- Maximum responses per person
- Minimum responses for close
- E-mail message subject heading
- Number of reminders

See Also

Gathering All Survey Campaign Requirements

Requirements for Creating and Deploying a Survey Questionnaire

Requirements for Determining If Survey Campaign Is List-Based

Requirements for Generating Lists

Requirements for Administering Survey Resources and Survey Campaign and Cycle Details

Requirements for Deploying Survey Campaigns

Monitoring Survey Results

Collecting Survey Results in Oracle RDBMS

Requirements for Reporting Survey Campaign Deployment Results

2.2.7 Requirements for Deploying Survey Campaigns

Once one or more deployments have been defined under a cycle for a survey campaign, it can be deployed immediately.

Non-List-Based Scenario

For non-list-based campaigns, the act of deploying (setting a deployment to Active status) enables respondents to take surveys as soon as you press the Deploy button.

List-Based Scenario

For list-based campaigns, setting a deployment to Active status allows e-mail invitations to be sent out based on the date and time parameters established for each specific deployment. This requires:

- An invitation Master Document, with an appropriate query matching all merge fields and associated with the Master Document.
- A Fulfillment Template, associating a specific Master Document and its components.
- Fully configured lists, typically generated in OMO.
- Appropriately configured concurrent processes.

When a deployment is set to Active, a concurrent job request is posted to the Concurrent Manager. The Concurrent Manager is functionality built on Oracle

Application Object Library (AOL) classes, which are included in Oracle Applications 11*i* with an appropriate Rapid Install.

The Deploy Date and Deploy Time are passed to the Concurrent Manager as the appropriate execution time to run the process. If the date and time specified are past the SYSDATE, the job executes immediately. Otherwise, the request remains on the concurrent request queue until the execution time arrives, and a Fulfillment request ID. Upon execution time, the Fulfillment server is notified to follow the instructions provided to it by the Fulfillment template. The template is associated with a particular Master Document (invitation or reminder) and query. The query pulls the contact information of the list member from OMO and unique id from `IES_SVY_LIST_ENTRIES` to merge them in the Master Document, personalizing the e-mail invitation and providing the appropriate list-based unique URL.

This personalized message, containing the data from all merge fields (at minimum, typically, the customer name and survey URL), is then sent to the outgoing e-mail server identified by the Fulfillment group and its member users (Survey administrators), and delivered through the Fulfillment server.

If reminders are associated with a deployment, then upon setting the deployment to active, concurrent jobs are submitted for reminders with calculated reminder dates and times, and instructions regarding the date to execute the job. These are also processed by the specified template, merging information as necessary and sending out the e-mail messages to the e-mail server identified with the Fulfillment group and its member users.

See Also

Gathering All Survey Campaign Requirements

Requirements for Creating and Deploying a Survey Questionnaire

Requirements for Determining If Survey Campaign Is List-Based

Requirements for Generating Lists

Requirements for Administering Survey Resources and Survey Campaign and Cycle Details

Requirements for Defining Deployments

Monitoring Survey Results

Collecting Survey Results in Oracle RDBMS

Requirements for Reporting Survey Campaign Deployment Results

2.2.8 Monitoring Survey Results

There are no hard and fast requirements for monitoring survey results. As soon as respondents complete a survey over the Web, the information is passed to the Scripting schema in the Oracle applications database. From the Survey Admin console, each individual response can be reviewed on an ongoing basis from the **Response** tab in the Survey Admin console. You must drill down to the specific deployment to see responses for that deployment.

See Also

Gathering All Survey Campaign Requirements

Requirements for Creating and Deploying a Survey Questionnaire

Requirements for Determining If Survey Campaign Is List-Based

Requirements for Generating Lists

Requirements for Administering Survey Resources and Survey Campaign and Cycle Details

Requirements for Defining Deployments

Requirements for Deploying Survey Campaigns

Collecting Survey Results in Oracle RDBMS

Requirements for Reporting Survey Campaign Deployment Results

2.2.9 Collecting Survey Results in Oracle RDBMS

When scripts are executed, information is automatically collected into the Scripting schema in the Oracle applications database. When surveys are taken over the Web, additional information is made available in survey transaction tables. No action or planning is required for this collection to take place.

See Also

Gathering All Survey Campaign Requirements

Requirements for Creating and Deploying a Survey Questionnaire

Requirements for Determining If Survey Campaign Is List-Based

Requirements for Generating Lists

Requirements for Administering Survey Resources and Survey Campaign and Cycle Details

Requirements for Defining Deployments

Requirements for Deploying Survey Campaigns

Monitoring Survey Results

Requirements for Reporting Survey Campaign Deployment Results

2.2.10 Requirements for Reporting Survey Campaign Deployment Results

You can view four types of reports (Response Summary, List Summary, Question Frequency, or Panel Footprint) by selecting the **Analysis** tab from the Survey Admin console. Each has its own required parameters to query from the **Analysis** tab. You must have the appropriate information for each report in order to query the information to display. If no surveys in a deployment have received responses, no information will be available to display in these reports.

From Where Does Reporting Information Derive?

When a script is executed as a survey through the Web, a concurrent process moves survey transaction information from the Scripting schema into survey transaction summary tables from which Survey campaign reports are generated. Summary tables can also be updated at scheduled times, as described in the administration section of this document.

For more information, see Reporting and Analyzing Survey Respondent Data > Parameters Required for Reports Accessible in Survey Admin Console.

See Also

Gathering All Survey Campaign Requirements

Requirements for Creating and Deploying a Survey Questionnaire

Requirements for Determining If Survey Campaign Is List-Based

Requirements for Generating Lists

Requirements for Administering Survey Resources and Survey Campaign and Cycle Details

Requirements for Defining Deployments

Requirements for Deploying Survey Campaigns

Monitoring Survey Results

Collecting Survey Results in Oracle RDBMS

3

Using Oracle Scripting

Oracle Scripting is a set of tools, each with its own users. End users (users of a script) include survey respondents (for the Survey component) and interaction center agents (for the Scripting Engine component). Users of the Script Author include script developers (trained functional users with some technical knowledge), and Java, PL/SQL, and Oracle Forms developers. Individuals with strong interaction center and campaign planning backgrounds participate in survey campaign requirements definition and survey campaign administration using the Survey Admin console.

Using Oracle Scripting involves all these different facets. Some of this information is dispersed throughout different sections of this document. Reference will be made herein to other sections as appropriate.

This topic includes the following topic groups:

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

See Also

Understanding Oracle Scripting > Who Are the End Users of Oracle Scripting?

3.1 Using the Script Author

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

categories. Thus a single subtopic may be referenced under multiple locations in this document.

This topic group includes the following topics:

Getting Started with the Script Author

Obtaining Script Requirements Before Creating a Script

Defining Global Script Attributes

Working with Script Files

Working with the Tool Palette

Viewing Objects on the Canvas

Working With Objects on the Canvas

Defining Panels

Defining Groups

Defining Blocks

Defining Branches

Controlling Script Flow

Defining Panel Answer Controls

Working With Answers

Defining Actions

Defining Commands

Deploying the Script

See Also

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.1 Getting Started with the Script Author

There is no single correct manner in which to create scripts in the Script Author. However, certain information is essential, and a recommended flow of events is described in this section. Other task-oriented processes are described in the remainder of this section, including step-by-step procedures to use the Script Author effectively.

To develop a new script you must obtain script campaign requirements, insert the scripting objects needed to implement the business rules into the Script Author, associate properties for each script object, ensure branches will provide the appropriate flow, define global script attributes based on your requirements, save the script, and deploy it to the applications database. The specific major tasks associated with this process are indicated below.

Steps

1. Obtain script campaign requirements.
2. Start a new script project.
3. Insert appropriate script objects.
4. Define properties for each configurable script object (panels, groups, and blocks).
5. Insert appropriate branching to meet flow objectives.
6. Define global script attributes, such as:
    a. Script properties (Script Name, Comments, Script Language, Footprinting and Answer Collection
    b. Global script pre- and post-actions
    c. Script information panel
    d. Shortcut buttons
    e. Script Disconnect button
7. Check the syntax of the script.
8. Save the script.
9. Deploy the script to the Oracle Applications database.

See Also

Obtaining Script Requirements Before Creating a Script

Defining Global Script Attributes
Working with Script Files
Working with the Tool Palette
Viewing Objects on the Canvas
Working With Objects on the Canvas
Defining Panels
Defining Groups
Defining Blocks
Defining Branches
Controlling Script Flow
Defining Panel Answer Controls
Working With Answers
Defining Actions
Defining Commands
Deploying the Script
Using the Script Author
Troubleshooting the Script Author
Using the Scripting Engine
Taking a Survey
Using the Survey Admin Console

3.1.2 Obtaining Script Requirements Before Creating a Script

From a planning perspective, a script is typically part of a campaign that must be executed. A **campaign** can be defined as a focused effort to achieve a particular goal from a targeted population over a specific period of time for a particular business purpose. This concept is typical of interaction centers and is supported from a software perspective in a variety of software products in the CRM portion of the eBusiness Suite.

Prior to inserting a single panel on the canvas, a script developer should have in her possession explicit script guidance in the form of detailed requirements. These requirements must be closely tailored to attain the campaign objectives.

Needed by Script Developers:

In order to develop scripts appropriately, the following, at minimum, must be identified in advance:

- The full set of business rules that must be enforced by the script
- Any text that must be communicated verbatim to the script audience (such as legal disclaimers, etc.)
- Decision points that cause branching in the flow of the script
- Data elements which must be captured during a script runtime session and used as a variable or otherwise processed later
- Data (table fields) which must be queried from or written to database tables

Script campaign administrators must provide detailed requirements to script developers. In addition, they should develop a detailed flowchart depicting the flow of the intended script. With these items in hand, a script developer can begin to implement the business requirements of the proposed campaign using Script Author objects and custom code.

For more information, see the *Oracle Scripting Implementation Guide* appendix, Scripting Project Scoping and Discovery.

See Also

Getting Started with the Script Author

Defining Global Script Attributes

Working with Script Files

Working with the Tool Palette

Viewing Objects on the Canvas

Working With Objects on the Canvas

Defining Panels

Defining Groups

Defining Blocks

Defining Branches

Controlling Script Flow

Defining Panel Answer Controls

Working With Answers

Defining Actions

Defining Commands

Deploying the Script

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.3 Defining Global Script Attributes

Global script attributes are attributes that affect the entire script. These include the script's global properties (including the script name recognized by the database), as well as aspects that appear during runtime in the Scripting Engine: the presence of a script information panel, shortcut buttons, and enabling the Disconnect button. You can perform the following tasks:

Designating Global Script Properties

Defining Global Script Pre- and Post-Actions

Defining the Script Information Panel

Defining Shortcut Buttons

Programming the Script Disconnect Button

See Also

Getting Started with the Script Author

Obtaining Script Requirements Before Creating a Script

Working with Script Files

Working with the Tool Palette

Viewing Objects on the Canvas

Working With Objects on the Canvas

Defining Panels

Defining Groups

Defining Blocks

Defining Branches

Controlling Script Flow

Defining Panel Answer Controls

Working With Answers

Defining Actions

Defining Commands

Deploying the Script

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.3.1 Designating Global Script Properties

Global script properties control the way scripts behave from a database perspective and in runtime, and affect data that is collected when scripts execute in the Scripting Engine. These script properties are all accessible in the Script Author by selecting **File > Script Properties**. The first set of global script attributes are the script properties listed below:

- Defining the Script Name
- Defining Script Comments
- Defining Script Language
- Setting Footprinting
- Setting Answer Collection for the Script

See Also

Defining Global Script Pre- and Post-Actions

Defining the Script Information Panel

Defining Shortcut Buttons

Programming the Script Disconnect Button

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.3.1.1 Defining the Script Name Use this procedure to define the global name of the script. This is the name under which the script is saved and referenced by the database. Do not include the percent character, single quotes, or double quotes in the global script name, as these are special database characters. The name you provide in this step is displayed (along with the designated script language) in the Script Chooser window at runtime.

The global script name is distinct from the script name from a file system perspective. To avoid confusion you may wish to use the same name in the script properties that you identify in the file system. If so, you must also avoid using spaces, slashes, or backslashes in the name, which are not permitted in the file system of some operating systems. Using a `.script` or `.scr` file extension in the global script name *is not necessary*, although it will cause no harm.

> **Caution:** When a script is created, it is automatically designated a global script name of `untitled1`. Unless you assign a new global script name, deploying it to the database will overwrite any other unnamed scripts that have been deployed to the database with the same default name.

Prerequisites

Create or open a script. You must have a script open to designate the script name.

Steps

1. Choose **File > Script Properties** or double-click on the canvas away from any objects.

   The Properties dialog box appears.

2. In the Script tree, select **Properties**.

3. In the Name field, type the name of the script.

   Do not include the percent character, single quotes, or double quotes in this name.

4. Click **Apply** to apply your work or click **OK** to save your work and exit the Properties dialog box.

See Also

Defining Script Comments

Defining Script Language

Setting Footprinting

Setting Answer Collection for the Script

Starting a New Script

Opening an Existing Script from the File System

Opening an Existing Script from the Applications Database

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.3.1.2 Defining Script Comments Use this procedure to include a comment associated with the global script. If script developers routinely open scripts from the database, or if multiple developers work on the same script, providing script comments may be helpful in identifying particular script versions or features. There is no maximum character limit to a comment and no character restrictions.

Prerequisites

Create or open a script. You must have a script open to define script comments.

Steps

1. Choose **File > Script Properties** or double-click on the canvas away from any objects.

   The Properties dialog box appears.

2. In the Script tree, select **Properties**.
3. In the Comments field, type the desired comment.
4. Click **Apply** to apply your work or click **OK** to save your work and exit the Properties dialog box.

See Also

Defining the Script Name

Defining Script Language

Setting Footprinting

Setting Answer Collection for the Script

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.3.1.3 Defining Script Language Use this procedure to designated a script language associated with a script. If using American English, this step is not necessary, as the value `AMERICAN` is the default. The Script Author script language setting can use any language supported by Oracle Applications as designated in the `FND_LANGUAGES` table.

Note: Changing the script language setting *does not translate GUI elements or script elements.* The script language setting is intended solely to communicate to agents attempting to launch a script in what language that script has been created. If an enterprise has a business requirement to translate an English script to Spanish, for example, a script developer can open the version labeled `AMERICAN`, change the setting to `SPANISH`, and then must manually translate all panel text, answer lookup values, panel names or labels, or any customized aspect of a script. The name of the script can be left the same as the English version. When the Spanish version is deployed to the database, both will list in the Script Chooser.

Prerequisites

Create or open a script. You must have a script open to designate the script language.

Steps

1. Choose **File > Script Properties** or double-click on the canvas away from any objects.

   The Properties dialog box appears.

2. In the Script tree, select **Properties**.

3. In the Script language field, enter the appropriate language to indicate in what language the script is written.

   Any language used in `FND_LANGUAGES` is supported.

See Also

Defining the Script Name

Defining Script Comments

Setting Footprinting

Setting Answer Collection for the Script

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.3.1.4 Setting Footprinting Use this procedure to set footprinting for the appropriate Oracle Scripting runtime component. This is accomplished on a per-script basis from the Script Author. Footprinting records the sequence of panels enabled during a script runtime interaction (regardless of whether the script is viewed in the Scripting Engine or the Survey mode), as well as the start time and end time (in milliseconds) for each panel in an interaction. Additionally, footprinting records deleted status—indicated that a panel was removed from the final flow based on a changed answer that takes the user down a different flow path. Footprinting data provides interaction center managers the ability to review existing scripts to determine potential problems and to tune the script accordingly. For example, you can analyze footprinting data to identify areas in the script where substantial amount of time is spent or where a specific panel is often deleted during an interaction. These scripts can then be modified with the goal of decreasing average talk time (for an interaction center) or increasing valid response rate by improving the flow of a script.

Prerequisites

Create or open a script. You must have a script open to set footprinting.

Steps

1. Choose **File > Script Properties** or double-click on the canvas away from any objects.

   The Properties dialog box appears.

2. In the Script tree, select **Properties**.
3. To record the start time and end time for every script panel that is enabled during an agent interaction, select **Footprinting**.
4. Click **Apply** to apply your work or click **OK** to save your work and exit the Properties dialog box.

See Also

Defining the Script Name

Defining Script Comments

Defining Script Language

Setting Answer Collection for the Script

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.3.1.5 Setting Answer Collection for the Script

Use this procedure to have Oracle Scripting record the answers to the script panels enabled during an agent interaction. These are stored in the Scripting schema of the Oracle Applications database.

Prerequisites

Create or open a script. You must have a script open to set Answer Collection.

Steps

1. Choose **File > Script Properties** or double-click on the canvas away from any objects.

   The Properties dialog box appears.

2. In the Script tree, select **Properties**.
3. To record the answers to the script panels enabled during an agent interaction, select **Answer Collection**.
4. Click **Apply** to apply your work or click **OK** to save your work and exit the Properties dialog box.

See Also

Defining the Script Name

Defining Script Comments

Defining Script Language

Setting Footprinting

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.3.2 Defining Global Script Pre- and Post-Actions

Use this procedure to define an action to take place before the first script panel appears (a pre-action) or after the last panel is executed in a script (a post-action).

Prerequisites

Create or open a script. You must have a script open to designate global pre- and post-actions.

Steps

1. In the tool palette, click the **Toggle Select Mode** tool.
1. Choose **File > Script Properties** or double-click on the canvas away from any objects.

   The Properties dialog box appears. These are the properties for the global script.
2. In the Script tree, expand **Actions** and then select **Pre Actions** or **Post Actions**.
3. In the Actions pane, click **Add**.

   The Command dialog box appears.
4. Define a command for the action. (See Defining Commands.)
5. Click **Apply** to apply your work or click **OK** to save your work and exit the Properties dialog box.

See Also

Designating Global Script Properties

Defining the Script Information Panel

Defining Shortcut Buttons

Programming the Script Disconnect Button

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.3.3 Defining the Script Information Panel

Use this procedure to setup the script information panel for the Scripting Engine. The information panel can contain up to nine data elements (text or a timer). The information panel will display in three rows, each able to display a data element and label at left, middle and right position. The data elements can be any combination of alphanumeric elements or timers. Sometimes referred to as the Static Information Panel, the data elements here actually may contain static or dynamic information. This panel is often used to identify static information such as the particular campaign name or business purpose of the script. It may also be used dynamically, displaying dynamic timers or customer profile information. This can be populated by executing a query (using a block) for customer information, and, using commands associated from the information panel, populating the data elements in the panel. Using Scripting APIs, timers can be started and stopped, and data in the script information panel can be updated by explicit command.

Prerequisites

Create or open a script. You must have a script open to define script information panel elements.

Steps

1. Choose **File > Script Properties**.

   The Properties dialog box appears.

2. In the Script tree, select **Static Info Panel**.
3. In the Static Info Panel pane, click an area in the representation of the script header.
4. If you want to insert text, then click **Text**.
5. If you want to insert a timer, then click **Timer**.
6. In the ID field, enter an identifying value.

   The Oracle Scripting API requires this ID.

7. In the Label field, enter the label that appears in the script information panel at runtime.

8. In the Command field, click **Edit**.

   The Command dialog box appears.

9. Define a command for the information panel text or timer. (See Defining Commands.)

10. Click **Apply** to apply your work or click **OK** to save your work and exit the Properties dialog box.

See Also

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.3.4 Defining Shortcut Buttons

Use this procedure to define a button in the toolbar of the Scripting Engine interface. Clicking this button in runtime will jump the interaction center agent to the group in the script with the appropriate shortcut property.

Prerequisites

Insert a group.

Steps

1. In the tool palette, click the **Toggle Select Mode** tool.

2. On the canvas, double-click a group.

   The Properties dialog box appears.

3. In the Group tree, select **Shortcut**.

4. In the Shortcut pane, type the name of the shortcut.

5. Click **Apply** to apply your work or click **OK** to save your work and exit the Properties dialog box for the group.

6. Choose **File > Script Properties**.

   The Properties dialog box appears.

7. In the Script tree, select **Shortcut Panel**.
8. In the Shortcut Panel pane, select **Add**.

   Shortcut Info Entry Dialog dialog box appears.
9. In the ID field, type the shortcut name for the group.
10. In the Label field, type the label that appears on the shortcut button at runtime.
11. In the Tooltip field, type an on screen description of the shortcut button that appears when the user's pointer pauses over the button.
12. Click **Edit** in the Command field.

    The Command dialog box appears.
13. Define a command for the shortcut button. (See Defining Commands.)
14. If you want to make the shortcut enabled in the script interface for the compiled script, then select **Enabled**.
15. Click **OK** to exit the Shortcut Info Entry Dialog dialog box.
16. Click **Apply** to apply your work or click **OK** to save your work and exit the Properties dialog box for the script.

See Also

Designating Global Script Properties

Defining Global Script Pre- and Post-Actions

Defining the Script Information Panel

Programming the Script Disconnect Button

Programming the Script Disconnect Button

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.3.5 Programming the Script Disconnect Button

Use this procedure to program the Disconnect button in the Scripting Engine interface. Selecting the Disconnect button at runtime will route the agent directly to this group. The group need not contain any panels, as long as it meets the syntax rules for any group (see What Are the Minimum Requirements for Any Graph?). If panels are included in this group, they will be displayed each time the Disconnect button is clicked. For a quick exit disconnect feature, simply insert a Termination node into this group and attach the Start and Termination nodes with a default branch.

Prerequisites

Insert a group.

Steps

1. In the tool palette, click the **Toggle Select Mode** tool.
2. On the canvas, double-click a group.

   The Properties dialog box appears.
3. In the Group tree, select **Shortcut**.

   The Shortcut pane will display.
4. In the Shortcut field in the Shortcut pane, enter the value **WrapUpShortcut**.

   This must be entered exactly as indicated (this property is case sensitive).
5. Click **Apply** to apply your work or click **OK** to save your work and exit the Properties dialog box for the group.

   When the Disconnect button is selected at runtime, the first panel of this group is displayed.

See Also

Designating Global Script Properties

Defining Global Script Pre- and Post-Actions

Defining the Script Information Panel

Defining Shortcut Buttons

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.4 Working with Script Files

You can perform the following tasks:

Starting a New Script

Opening an Existing Script from the File System

Opening an Existing Script from the Applications Database

Saving a Script to the File System

Reversing All Changes Since the Last Save Command

Importing a Script

Exporting a Script Group as a Separate Script File

Printing a Graph on the Script Author Canvas

Closing a Script

Toggling Sticky Mode

Exiting the Script Author

See Also

Getting Started with the Script Author

Obtaining Script Requirements Before Creating a Script

Defining Global Script Attributes

Working with the Tool Palette

Viewing Objects on the Canvas

Working With Objects on the Canvas

Defining Panels

Defining Groups

Defining Blocks

Defining Branches

Controlling Script Flow

Defining Panel Answer Controls

Working With Answers

Defining Actions

Defining Commands

Deploying the Script

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.4.1 Starting a New Script

Use this procedure to create a new script.

Prerequisites

None

Steps

- Choose **File** > **New**, or click the **New Script** icon in the toolbar.

  The **New Script** icon is the first icon from the left of the toolbar, located immediately below the menus in the Script Author interface.

  The Start node appears on the canvas.

See Also

Opening an Existing Script from the File System

Opening an Existing Script from the Applications Database

Saving a Script to the File System

Reversing All Changes Since the Last Save Command

Importing a Script

Exporting a Script Group as a Separate Script File

Printing a Graph on the Script Author Canvas

Closing a Script

Toggling Sticky Mode

Exiting the Script Author

3.1.4.2 Opening an Existing Script from the File System

Use this procedure to open a saved script from the file system. Scripts can be opened from the file system on any local volume, or from the applications database for your environment.

Prerequisites

None

Steps

1. Choose **File** > **Open** or click the **Open a Script** icon in the toolbar.

   The **Open a Script** icon is the second icon from the left of the toolbar, located immediately below the menus in the Script Author interface.

   The **Script Chooser** window for the Script Author will open.

2. Click the **File System** tab from the **Script Chooser** window, if not already selected.

3. From the **Location** pull-down menu, select the appropriate local volume.

4. From the **Files** list, locate the script (`*.script` file) that you want to open and then click **Open**.

   The script objects appear on the canvas.

> **Note:** If you wish to preserve the original script, ensure you change the name of the just-opened script in both the global script properties (**File > Script Properties**) and in the file system (**File > Save As**).

See Also

Starting a New Script

Opening an Existing Script from the Applications Database

Saving a Script to the File System

Reversing All Changes Since the Last Save Command

Importing a Script

Exporting a Script Group as a Separate Script File

Printing a Graph on the Script Author Canvas

Closing a Script

Toggling Sticky Mode

Exiting the Script Author

3.1.4.3 Opening an Existing Script from the Applications Database

Use this procedure to open a saved script from the applications database. Scripts can be opened from the file system on any local volume, or from the applications database for your environment.

Prerequisites

You must have appropriate database connection information, including a username with access to the `IES_DEPLOYED_SCRIPTS` table in the applications database.

Steps

1. Choose **File** > **Open**.

   The **Script Chooser** window for the Script Author will open.

2. Click the **Database** tab from the **Script Chooser** window, if not already selected.

3. Click **Edit** (located next to the **Connection** field).

   The **Edit Profile**

c. In the **Port Number** field, enter the TNS port of the database.

d. In the **SID** field, enter the database global name or SID.

e. In the **User ID** field, enter an appropriate database User ID.

   This user must have privileges to access the `IES_DEPLOYED_SCRIPTS` table of the applications database (for example, the `apps` user).

f. Once you have entered all appropriate connection information, click **OK**.

   The **Edit Profile** window will close. The **Connection** field will be populated with the connection name and User ID entered in the previous step.

5. In the **Password** field, enter the password for the configured User ID.
6. Optionally, you can select from the **Published** or **Deployed** radio buttons to view different scripts.
7. Locate the script (`*.script` file) that you want to open in the **Scripts** list and then click **Open**.

   The script objects appear on the canvas.

Note: If you wish to preserve the original script in the database, ensure you change the name of the just-opened script in the global script properties (**File > Script Properties**). If you will be saving the script to your local file system, change the file name of the script as well (**File > Save As**).

See Also

Starting a New Script

Opening an Existing Script from the File System

Saving a Script to the File System

Reversing All Changes Since the Last Save Command

Importing a Script

Exporting a Script Group as a Separate Script File

Printing a Graph on the Script Author Canvas

Closing a Script

Toggling Sticky Mode

Exiting the Script Author

3.1.4.4 Saving a Script to the File System

Use this procedure to save a script to the file system of a local volume or mounted storage medium. You can save a script using its current name and location, or save a copy of the script using a different name and location. This name must not have any special characters, such as spaces, slashes, or backslashes, which are not permitted in the file system of some operating systems.

> **Note:** In order for the Script Author to be able to recognize a script file, it must end with a `.script` or a `.scr` file extension.

The name which you provide when you save the script in the *file system* is distinct from the name by which the script is identified in the *script properties*. It is the name in the script properties which identifies the script in the database and in the Script Chooser window at runtime. Therefore, you may wish to use the same name in the script properties that you identify in this step for the file name.

> **Note:** In the title bar of the Script Author, the asterisk to the right of the script name indicates that changes have been made to the script that have not yet been saved.

Prerequisites

None

Steps

1. Do one of the following:
   - To save the script using its current name and location, choose **File** > **Save** or click the **Save Script** icon in the toolbar.

     The **Save Script** icon is the third icon from the left of the toolbar, located immediately below the menus in the Script Author interface.
   - To save a copy of the script using a different name and location, choose **File** > **Save As**.
   - Ensure the script name ends with a `.script` or `.scr` file extension.

See Also

Defining the Script Name

Starting a New Script

Opening an Existing Script from the File System

Opening an Existing Script from the Applications Database

Reversing All Changes Since the Last Save Command

Importing a Script

Exporting a Script Group as a Separate Script File

Printing a Graph on the Script Author Canvas

Closing a Script

Toggling Sticky Mode

Exiting the Script Author

3.1.4.5 Reversing All Changes Since the Last Save Command

Use this procedure to undo all changes made since you last saved the script. This action can be performed any time script changes are made and saved.

Prerequisites

None

Steps

Choose **File** > **Revert to > Last Save**.

See Also

Starting a New Script

Opening an Existing Script from the File System

Opening an Existing Script from the Applications Database

Saving a Script to the File System

Importing a Script

Exporting a Script Group as a Separate Script File

Printing a Graph on the Script Author Canvas

Closing a Script

Toggling Sticky Mode

Exiting the Script Author

3.1.4.6 Importing a Script

Use this procedure to import a previously existing script into the current script.

The import command imports all objects in an existing script. If you wish to import only a portion of a script, you can (1) export that portion from the existing script and then import it as described below, or (2) import the entire script as described below and discard the unwanted portions.

> **Caution:** Every Script Author script is essentially a customized product. It is the script developer's responsibility to ensure **imported** scripts will function in the target environment. If a script you **import** contains commands referencing custom code (e.g., Java, PL/SQL, Forms, and so on), the corresponding code referenced by the script must be available in the environment in which the script is deployed.

Prerequisites

None

Steps

1. Choose **File > Import**.

   The Open dialog box appears.

2. Locate the script file (`*.script`) that you want to import.

3. Click the file name and then click **Open**.

   The imported script appears on the canvas as a group.

See Also

Starting a New Script

Opening an Existing Script from the File System

Opening an Existing Script from the Applications Database

Saving a Script to the File System

Reversing All Changes Since the Last Save Command

Exporting a Script Group as a Separate Script File

Printing a Graph on the Script Author Canvas

Closing a Script

Toggling Sticky Mode

Exiting the Script Author

3.1.4.7 Exporting a Script Group as a Separate Script File

Use this procedure to export a group as a separate script file. Exported groups can then be imported and used in other scripts, providing for modularity and code reuse.

Prerequisites

None

The import command imports all objects in an existing script. If you wish to import only a portion of a script, you can (1) export that portion from the existing script and then import it as described below, or (2) import the entire script as described below and discard the unwanted portions.

> **Caution:** Every Script Author script is essentially a customized product. It is the script developer's responsibility to ensure exported groups will function in the target environment. If a script component you export contains commands referencing custom code (e.g., Java, PL/SQL, Forms, and so on), the corresponding code referenced by the script must be available in the environment in which the script is deployed.

Steps

1. On the canvas, select a group.
1. Choose **File > Export**.

   The Save dialog box appears.
2. Select the destination.

3. Type the file name and then click **OK**.

See Also

Starting a New Script

Opening an Existing Script from the File System

Opening an Existing Script from the Applications Database

Saving a Script to the File System

Reversing All Changes Since the Last Save Command

Importing a Script

Printing a Graph on the Script Author Canvas

Closing a Script

Toggling Sticky Mode

Exiting the Script Author

3.1.4.8 Printing a Graph on the Script Author Canvas

Use this procedure to print a graph displayed on the Script Author canvas.

Prerequisites

None

Steps

1. Select the graph you wish to print.
2. Choose **File > Print**.
3. Select your print options and then click **OK**.

> **Note:** To print a subgraph contained within a group or block, you must first click to select the appropriate container object (the parent group or block) and then click the **Go down into child graph** button.

See Also

Fitting a Script Layout in the Canvas

Drilling Up or Down to a Related Script

Aligning Objects

Starting a New Script

Opening an Existing Script from the File System

Opening an Existing Script from the Applications Database

Saving a Script to the File System

Reversing All Changes Since the Last Save Command

Importing a Script

Exporting a Script Group as a Separate Script File

Closing a Script

Toggling Sticky Mode

Exiting the Script Author

3.1.4.9 Closing a Script

Use this procedure to close a script. You can close a script at any time. If you have unsaved changes, you will be asked whether you want to save the changes.

Prerequisites

None

Steps

1. From the **File** menu, choose **Close**.

   A prompt will appear in the **Select an Option** window asking if you wish to save the changes in the graph.

2. Select the appropriate choice from the prompt.

See Also

Saving a Script to the File System

Starting a New Script

Opening an Existing Script from the File System

Opening an Existing Script from the Applications Database

Saving a Script to the File System

Reversing All Changes Since the Last Save Command

Importing a Script

Exporting a Script Group as a Separate Script File

Printing a Graph on the Script Author Canvas

Toggling Sticky Mode

Exiting the Script Author

3.1.4.10 Toggling Sticky Mode

In every Script Author session, the sticky mode feature is enabled by default. In sticky mode, the script object selected in the tool palette remains "stuck" (selected) until explicitly turned off by selecting the Toggle Selection Mode tool in the tool palette. This feature suits developers who prefer to insert objects on the canvas first (based on their requirements) and subsequently assign attributes the script objects. If you prefer to assign attributes to each object as you create it, leaving sticky mode on can result in the unwanted effect of inserting multiple objects when your intended result is to designate properties for the current object. This is an example of when you would toggle sticky mode off.

Use this procedure to toggle sticky mode on or off.

Prerequisites

None

Steps

1. From the **File** menu, view the **Sticky Mode** setting.

   If sticky mode is toggled on, the checkbox will be enabled. If sticky mode is toggled off, no check will appear in the checkbox.

2. If sticky mode is toggled *on*, the checkbox is checked. To toggle sticky mode off, click the checkbox.

   The **File** menu will close, and sticky mode is toggled off. To verify, select the **File** menu and view the checkbox, which will now be unchecked.

3. If sticky mode is toggled *off*, no check will appear in the checkbox. To toggle sticky mode on, click the checkbox.

The **File** menu will close, and sticky mode is toggled on. To verify, select the **File** menu and view the checkbox, which will now be checked.

See Also

Starting a New Script

Opening an Existing Script from the File System

Opening an Existing Script from the Applications Database

Saving a Script to the File System

Reversing All Changes Since the Last Save Command

Importing a Script

Exporting a Script Group as a Separate Script File

Printing a Graph on the Script Author Canvas

Closing a Script

Exiting the Script Author

3.1.4.11 Exiting the Script Author

Use this procedure to exit the Script Author application. If you have unsaved changes, you will be asked whether you want to save the changes.

Prerequisites

None

Steps

1. From the **File** menu, choose **Exit**.

   A prompt will appear in the **Select an Option** window asking if you wish to save the changes in the graph.

2. Select the appropriate choice from the prompt.

See Also

Saving a Script to the File System

Starting a New Script

Opening an Existing Script from the File System

Opening an Existing Script from the Applications Database

Saving a Script to the File System

Reversing All Changes Since the Last Save Command

Importing a Script

Exporting a Script Group as a Separate Script File

Printing a Graph on the Script Author Canvas

Closing a Script

Toggling Sticky Mode

3.1.5 Working with the Tool Palette

You can perform the following tasks:

Inserting an Object

Inserting a Branch

See Also

Selecting Objects

Deselecting Objects

Moving Objects

Moving Branches

Deleting Panels, Groups, Blocks and Termination Nodes

Deleting Branches

Getting Started with the Script Author

Obtaining Script Requirements Before Creating a Script

Defining Global Script Attributes

Working with Script Files

Viewing Objects on the Canvas

Working With Objects on the Canvas

Defining Panels

Defining Groups

Defining Blocks

Defining Branches

Controlling Script Flow

Defining Panel Answer Controls

Working With Answers

Defining Actions

Defining Commands

Deploying the Script

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.5.1 Inserting an Object

Script Author has three configurable objects (panels, blocks and groups) and two non-configurable objects (the Start and Termination nodes). These objects must be attached with branches to direct the flow of the script.

- If you want to display information (such as text, an image, a hyperlink, a variable, or a Java bean) to the script viewer at runtime or accept information entered by the script viewer, then insert a panel.
- If you want to query, update, or insert information in database tables, or insert a command or API with a clear visual indicator, then insert a block.
- If you want to logically group a section of the script's functionality, access a section of the script in runtime using a shortcut button, or group functionality that is the target of a Java method in runtime associated with an indeterminate branch, then insert a group.
- If you want to direct the script flow to the end of processing on that graph, then insert a Termination node.

You can perform the following tasks:

- Setting Properties Dialog Box to Pop Up at Object Creation

- Inserting a Panel
- Inserting a Group
- Inserting a Block
- Inserting a Termination Node

3.1.5.1.1 Setting Properties Dialog Box to Pop Up at Object Creation Use this procedure to set up Script Author to automatically display the Properties dialog box when you insert a panel, group, or block onto the canvas. This feature can be particularly helpful when initially inserting panels, as panels require at least one answer be defined. Groups and Blocks also require appropriate termination and branching within their subgraphs. For more information, see What Are the Minimum Requirements for Any Graph?

Prerequisites

None

Steps

- Choose **View** > **Popup on Blob Creation**.

See Also

Inserting a Panel

Inserting a Group

Inserting a Block

Inserting a Termination Node

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.5.1.2 Inserting a Termination Node Use this procedure to insert a Termination node onto the canvas. There are no properties to associate for this object. For more information, see What Are the Building Blocks of a Script? > Termination Nodes.

Note: Every script and all subscripts for groups and blocks must end with a Termination node. Even if the block does not have any panels, it still must have a Termination node.

Prerequisites

None

Steps

1. In the tool palette, click the **Termination Point Insertion Mode** tool.

   When the pointer is over the canvas, the pointer a pointing finger

2. On the canvas, click where you want to insert the Termination node.

See Also

Setting Properties Dialog Box to Pop Up at Object Creation

Inserting a Panel

Inserting a Group

Inserting a Block

Termination Nodes

Deleting Panels, Groups, Blocks and Termination Nodes

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.5.2 Inserting a Branch

Use this procedure to insert a branch onto the canvas to connect objects. Branches direct the flow of the script based on the branch type and properties. For more information on branches, see Defining Branches.

Prerequisites

- Insert an object from which to begin branching (or use Start node).
- Insert a destination object.

Steps

1. In the tool palette, click the appropriate branch type tool.

   When the pointer is over the canvas, the pointer is a cross hair (+).

2. On the canvas, drag the pointer from the source object to the destination object. In the case of an Indeterminate branch, an empty space on the canvas is the destination.

   When you release the pointer over the destination object, the branch arrow appears. The branch is red. This indicates that the branch is currently selected.

See Also

Defining Branches

Inserting a Panel

Inserting a Group

Inserting a Block

Inserting a Termination Node

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.6 Viewing Objects on the Canvas

You can perform the following tasks:

Fitting a Script Layout in the Canvas

Drilling Up or Down to a Related Script

Aligning Objects

See Also

Printing a Graph on the Script Author Canvas

Getting Started with the Script Author

Obtaining Script Requirements Before Creating a Script

Defining Global Script Attributes

Working with Script Files

Working with the Tool Palette

Working With Objects on the Canvas

Defining Panels

Defining Groups

Defining Blocks

Defining Branches

Controlling Script Flow

Defining Panel Answer Controls

Working With Answers

Defining Actions

Defining Commands

Deploying the Script

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.6.1 Fitting a Script Layout in the Canvas

You can zoom in to focus on the details of the script layout or zoom out to see more of the script layout.

Prerequisites

None

Steps

Do one of the following:

- To increase the magnification, click the **Zoom In** button on the canvas toolbar.
- To decrease the magnification, click the **Zoom Out** button on the canvas toolbar.
- To set the magnification to 100%, click the **Zoom to 100%** button on the canvas toolbar.
-

2. To drill down, select a group or block on the canvas and then click **Go down into child graph** on the toolbar.
3. To drill up one level, click **Go up to parent graph** on the toolbar.
4. To go to the top level graph, click **Go to root graph** on the toolbar.

See Also

Fitting a Script Layout in the Canvas

Aligning Objects

Printing a Graph on the Script Author Canvas

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.6.3 Aligning Objects

Use this procedure to align objects vertically or horizontally.

Prerequisites

None

Steps

1. In the tool palette, click the **Toggle Select Mode** tool.
2. On the canvas, select the objects that you want to align.
3. On the canvas toolbar, click **Align Selected Objects Vertically** or **Align Selected Object Horizontally**.

See Also

Fitting a Script Layout in the Canvas

Drilling Up or Down to a Related Script

Printing a Graph on the Script Author Canvas

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.7 Working with Objects on the Canvas

You can perform the following tasks:

Selecting Objects

Deselecting Objects

Moving Objects

Moving Branches

Deleting Panels, Groups, Blocks and Termination Nodes

Deleting Branches

See Also

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.7.1 Selecting Objects

Use this procedure to select one or more objects. You can select several objects at the same time, or you can add objects to an existing selection.

Prerequisites

All of the objects to be selected must be on the same canvas.

Steps

1. In the tool palette, click the **Toggle Select Mode** tool.
2. Do one of the following:

- To select an object, click the object.
- To select several object in the same area, point outside the objects and drag diagonally to draw a selection border around them.

  All objects in or on the selection border are selected. This includes branches.
- To add an object to a selection, Shift-click the object. You can also Control-click the object.

See Also

Deselecting Objects

Moving Objects

Moving Branches

Deleting Panels, Groups, Blocks and Termination Nodes

Deleting Branches

Fitting a Script Layout in the Canvas

Drilling Up or Down to a Related Script

Aligning Objects

Printing a Graph on the Script Author Canvas

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.7.2 Deselecting Objects

Use this procedure to deselect one or more selected objects.

Prerequisites

None

Steps

Do one of the following:

- To deselect an object, click outside the object.
- To deselect one of several selected objects, Shift-click the object. You can also Control-click the object.
- To deselect all selected objects, click on the canvas away from any objects.

See Also

Selecting Objects

Moving Objects

Moving Branches

Deleting Panels, Groups, Blocks and Termination Nodes

Deleting Branches

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.7.3 Moving Objects

Use this procedure to move panels, groups, and blocks on the canvas.

Prerequisites

None

Steps

1. In the tool palette, click the **Toggle Select Mode** tool.
2. On the canvas, select one or more objects.
3. Drag the object to the desired location.

   If a branch is connected to the object, then the branch moves with the object.

See Also

Selecting Objects

Deselecting Objects

Moving Branches

Deleting Panels, Groups, Blocks and Termination Nodes

Deleting Branches

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.7.4 Moving Branches

Use this procedure to change the destination object for a branch.

Prerequisites

None

Steps

1. In the tool palette, click the **Toggle Select Mode** tool.
2. On the canvas, select the destination object for a branch.

   Note: This procedure requires you to delete the destination object. To retain the object, use the Edit command to copy the object and paste it on the canvas.

3. Choose **Edit** > **Delete**.

   The object that was previously the destination of the branch is deleted.
4. Select the new destination object.
5. Drag the destination object to the branch.
6. When the branch turns red, release the destination object.

   The branch is redrawn to the new destination object.

See Also

Selecting Objects

Deselecting Objects

Moving Objects

Deleting Panels, Groups, Blocks and Termination Nodes

Deleting Branches

Using the Script Author

Deselecting Objects

Moving Objects

Moving Branches

Deleting Branches

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.7.6 Deleting Branches

Use this procedure to delete branches from the canvas.

> **Note:** Deleting a branch deletes all properties associated with the branch.

Prerequisites

None

Steps

1. In the tool palette, click the **Toggle Select Mode** tool.
2. On the canvas, select a branch.

   When the branch is selected, the branch is red.
3. Choose **Edit** > **Delete**.

See Also

Selecting Objects

Deselecting Objects

Moving Objects

Moving Branches

Deleting Panels, Groups, Blocks and Termination Nodes

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.8 Defining Panels

The panel is the only Script Author object that is visible at runtime, displaying panel text, any answers (nodes) that have been defined for the panel, and associated labels for those answers.

You can perform the following tasks:

Inserting a Panel

Defining Panel Properties

Defining Panel Answer Controls

Defining Panel Text

See Also

Defining Panel Pre- and Post-Actions

Getting Started with the Script Author

Obtaining Script Requirements Before Creating a Script

Defining Global Script Attributes

Working with Script Files

Working with the Tool Palette

Viewing Objects on the Canvas

Working With Objects on the Canvas

Defining Groups

Defining Blocks

Defining Branches

Controlling Script Flow

Defining Panel Answer Controls

Working With Answers

Defining Actions

Defining Commands

Deploying the Script

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.8.1 Inserting a Panel

Use this procedure to insert a panel onto the canvas. The panel is the only Script Author object that visible at runtime, displaying panel text, any answers (nodes) that have been defined for the panel, and associated labels for those answers. For more information on panels, see Defining Panels.

Prerequisites

None

Steps

1. In the tool palette, click the **Panel Insertion Mode** tool.

   When the pointer is over the canvas, the pointer is a pointing finger

2. On the canvas, click where you want to insert the panel.

   If the Popup on Blob Creation option is selected, then the Properties dialog box for the object appears.

> **Caution:** Ensure you define an answer for each panel created, or the script will not pass a syntax check. In runtime, every panel requires end user interaction (response to an answer definition).

See Also

Setting Properties Dialog Box to Pop Up at Object Creation

Inserting a Group

Inserting a Block

Inserting a Termination Node

Defining Panel Properties

Defining the Panel Name

Defining the Panel Comments

Defining the Panel Label

Defining Vertical or Horizontal Answer Control Layout

Substituting a Java Bean for a Panel

Defining Panel Pre- and Post-Actions

Defining Panel Answer Controls

Defining Panel Text

What Are the Minimum Requirements for Any Graph?

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.8.2 Defining Panel Properties

Use this procedure to define the properties of a panel. Panel properties affect the way in which a panel is stored in the database and displayed in runtime.

You can perform the following tasks:

- Defining the Panel Name
- Defining Panel Comments
- Defining the Panel Label
- Substituting a Java Bean for a Panel

Prerequisites

Insert a panel onto the canvas.

3.1.8.2.1 Defining the Panel Name Use this procedure to define the name of a panel. The panel name is how the panel is identified in Script Author. Note that the panel name is distinct from the panel label, which is displayed in the Progress Area of the Scripting Engine at runtime.

Prerequisites

Insert a panel onto the canvas.

Steps

1. In the tool palette, click the **Toggle Select Mode** tool.
2. On the canvas, double-click a panel.

   The Properties dialog box appears.
3. In the Panel tree, select **Properties**.
4. In the Name field, type the name of the panel.
5. Click **Apply** to apply your work or click **OK** to save your work and exit the Properties dialog box.

See Also

Inserting a Panel

Defining Panel Comments

Defining the Panel Label

Defining Vertical or Horizontal Answer Control Layout

Substituting a Java Bean for a Panel

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.8.2.2 Defining Panel Comments Use this procedure to define a comment for a panel. There is no maximum character limit to a comment and no character restrictions.

Prerequisites

Insert a panel onto the canvas.

Steps

1. In the tool palette, click the **Toggle Select Mode** tool.
2. On the canvas, double-click a panel.

   The Properties dialog box appears.
3. In the Panel tree, select **Properties**.
4. In the Comments field, type the desired comment.
5. Click **Apply** to apply your work or click **OK** to save your work and exit the Properties dialog box.

See Also

Defining the Panel Name

Defining the Panel Label

Defining Vertical or Horizontal Answer Control Layout

Substituting a Java Bean for a Panel

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.8.2.3 Defining the Panel Label Use this procedure to define the label for a panel. The panel label is the value used to identify specific panels in the Progress Area of the Scripting Engine at runtime. Only panels that have been visited by the end user in the flow will be listed (by panel label) in the Progress Area. Note that the value the Script Author uses to reference panels is the panel name, not the panel label.

Prerequisites

Insert a panel onto the canvas.

Steps

1. In the tool palette, click the **Toggle Select Mode** tool.
2. On the canvas, double-click a panel.

   The Properties dialog box appears.
3. In the Panel tree, select **Properties**.
4. In the Label field, type the label of the panel.
5. Click **Apply** to apply your work or click **OK** to save your work and exit the Properties dialog box.

See Also

Inserting a Panel

Defining the Panel Name

Defining Panel Comments

Defining Vertical or Horizontal Answer Control Layout

Substituting a Java Bean for a Panel

Packaging Java Bean Code Into a JAR File

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.8.2.4 Substituting a Java Bean for a Panel Use this procedure to substitute a Java bean for a panel.

Caution: Java beans are customized components. As such, no support is provided for scripts that have difficulties substituting panels with Java beans. The functionality is provided with the tool to allow unsupported customization.

Prerequisites

- Insert a panel onto the canvas.
- Write the code for the Java bean.

Steps

1. In the tool palette, click the **Toggle Select Mode** tool.
2. On the canvas, double-click a panel.

   The Properties dialog box appears.
3. In the Panel tree, select **Properties**.
4. In the Properties pane, select **Replace with a Java Bean**.
5. In the Bean Name field, type the full path and name of the Java bean (for example, mybeans.foobean).
6. In the Jar File Name field, type the full path and name of the jar file for the Java bean.
7. Click **Apply** to apply your work or click **OK** to save your work and exit the Properties dialog box.

See Also

Inserting a Panel

Defining the Panel Name

Defining Panel Comments

Defining the Panel Label

Defining Vertical or Horizontal Answer Control Layout

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.8.2.5 Packaging Java Bean Code Into a JAR File Oracle Scripting 11*i* allows the substitution of a custom Java Bean user interface in place of a standard Scripting panel. In order for a script with a custom Java Bean to supply that code to the Scripting Engine GUI at runtime, the code for the Java Bean must first be packaged into a `.jar` file. Due to a limitation in JDK 1.1.8, there are only two supported methods of packaging the JAR file appropriately. Either of these methods described below can be used to package a `.jar` file appropriately.

Caution: While an entire panel can be substituted, substituting a Java bean for *an answer* (or panel node) *within a panel* is no longer supported in Oracle Scripting as of release 11.5.6 and subsequent releases, due to the new WYSIWYG editing feature of the Script Author.

Prerequisites

- As with all custom code, the Java bean must be created by a certified Java developer using any appropriate Java development tool.
- Oracle recommends compiling custom Java code in support of Oracle Scripting against the Java Development Kit (JDK) 1.1.8.

Steps

- Use the "jar" utility from JDK 1.1.8 and specify no compression (e.g., `jar -cf0 TestBean.jar ...`).
- Create the custom JAR file in the standard `.zip` format (with any PC-standard compression utility) with or without compression, and then simply rename the file extension from `.zip` to `.jar`.

3.1.8.3 Defining Panel Text

You can perform the following tasks:

- Entering Panel Text
- Inserting a Hyperlink

- Inserting an Embedded Value
- Inserting an Image
- Exporting Panel Text to an HTML File
- Importing an HTML File into the Panel Text Editor

See Also

Inserting a Panel

Defining the Panel Name

Defining Panel Comments

Defining the Panel Label

Defining Panel Pre- and Post-Actions

Substituting a Java Bean for a Panel

3.1.8.3.1 Entering Panel Text Use this procedure to define the text that displays in the script panel at runtime.

Prerequisites

Insert a panel onto the canvas.

Steps

1. In the tool palette, click the **Toggle Select Mode** tool.
2. On the canvas, select a panel.
3. From the menu, choose **Tools > Panel Text Editor**.

   The panel text editor appears.
4. Type the panel text.
5. Optionally, select text and then select a format from the **Font** or **Lists** menu.
6. If you want to save the text to the panel, then choose **File > Save** from the menu.
7. If you want to save the text to an HTML file, then choose **File > Export** from the menu.

Note: Exporting the text to an HTML file does not save the text to the panel.

See Also

Inserting a Hyperlink

Inserting an Embedded Value

Inserting an Image

Exporting Panel Text to an HTML File

Importing an HTML File into the Panel Text Editor

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.8.3.2 Inserting a Hyperlink Use this procedure to insert a hyperlink into the text of a script panel.

Prerequisites

Insert a panel onto the canvas.

Steps

1. In the tool palette, click the **Toggle Select Mode** tool.
2. On the canvas, select a panel.
3. From the menu, choose **Tools > Panel Text Editor**.

   The panel text editor appears.
4. Select or type the text that represents the hyperlink and then click **Insert Link** in the toolbar.

   The selected text is blue and underlined.
5. With the cursor in the text or with the text selected, click **Modify Properties** in the toolbar.

The Hyperlink dialog box appears.

6. In the hyperlink window, do one of the following:
   - If you want to define the hyperlink using a URL (web address), then type the URL for the hyperlink. Include the protocol (e.g., **`http://`**) in the URL field.
   - If you want to define the hyperlink using a command, then select Command as Hyperlink and **Edit Command**. (See Defining Commands.)
7. Click **OK** to exit the Hyperlink dialog box.
8. If you want to save the text to the panel, then choose **File > Save** from the menu.
9. If you want to save the text to an HTML file, then choose **File > Export** from the menu.

> **Note:** Exporting the text to an HTML file does not save the text to the panel.

See Also

Entering Panel Text

Inserting an Embedded Value

Inserting an Image

Exporting Panel Text to an HTML File

Importing an HTML File into the Panel Text Editor

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.8.3.3 Inserting an Embedded Value Use this procedure to embed a value into the text of a script panel.

Prerequisites

Insert a panel onto the canvas.

Steps

1. In the tool palette, click the **Toggle Select Mode** tool.
2. On the canvas, select a panel.
3. From the menu, choose **Tools > Panel Text Editor**.

   The panel text editor appears.
4. Type the text placeholder for the embedded value and then click **Insert Embedded Value** in the toolbar.

   The background of the selected text is yellow.
5. With the cursor in the text or with the text selected, click **Modify Properties** in the toolbar.

   The Command dialog box appears.
6. Define a command that returns a value to be displayed in the script panel at runtime. (See Defining Commands.)

   In the panel text editor, the text selected to represent the embedded value now displays the command name.
7. If you want to save the text to the panel, then choose **File > Save** from the menu.
8. If you want to save the text to an HTML file, then choose **File > Export** from the menu.

Note: Exporting the text to an HTML file does not save the text to the panel.

See Also

Entering Panel Text

Inserting a Hyperlink

Inserting an Image

Exporting Panel Text to an HTML File

Importing an HTML File into the Panel Text Editor

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.8.3.4 Inserting an Image Use this procedure to insert an image into the text of a script panel.

Prerequisites

Insert a panel onto the canvas.

Steps

1. In the tool palette, click the **Toggle Select Mode** tool.
2. On the canvas, select a panel.
3. From the menu, choose **Tools > Panel Text Editor**.

   The panel text editor appears.
4. Position the insertion point where you want to insert the picture.
5. In the toolbar, click **Insert Image**.

   The Open dialog box appears.
6. Locate the file that contains the image you want to insert.
7. Click the file and then click **Open**.

   The image appears at the insertion point.
8. If you want to save the text to the panel, then choose **File > Save** from the menu.
9. If you want to save the text to an HTML file, then choose **File > Export** from the menu.

> **Note:** Exporting the text to an HTML file does not save the text to the panel.

See Also

Entering Panel Text

Inserting a Hyperlink

Inserting an Embedded Value

Exporting Panel Text to an HTML File

Importing an HTML File into the Panel Text Editor

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.8.3.5 Exporting Panel Text to an HTML File Use this procedure to save panel text as an HTML file.

> **Note:** Exporting the text to an HTML file does not save the text to the panel.

Prerequisites

Insert a panel onto the canvas.

Steps

1. In the tool palette, click the **Toggle Select Mode** tool.
2. On the canvas, select a panel.
3. From the menu, choose **Tools > Panel Text Editor**.

   The panel text editor appears.
4. If you want to save the text as an HTML file using its current name and location, then choose **File > Export**.
5. If you want to save the text as an HTML file using a different name and location, then choose **File > Export As**.

If you choose the Export As command, or if the script has never been exported, then the Save dialog box appears.

6. Specify the location and filename.
7. Click **Save**.

See Also

Entering Panel Text

Inserting a Hyperlink

Inserting an Embedded Value

Inserting an Image

Importing an HTML File into the Panel Text Editor

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.8.3.6 Importing an HTML File into the Panel Text Editor Use this procedure to import HTML into the panel text editor.

> **Note:** Importing an HTML file into the panel text editor overwrites any text in the panel text editor.

Prerequisites

Insert a panel onto the canvas.

Steps

1. In the tool palette, click the **Toggle Select Mode** tool.
2. On the canvas, select a panel.
3. From the menu, choose **Tools > Panel Text Editor**.

   The panel text editor appears.

4. Choose **File > Import**.

   The Open dialog box appears.

5. Locate the file that you want to import.

6. Click the file and then click **Open**.

   The HTML appears in the panel text editor.

See Also

Entering Panel Text

Inserting a Hyperlink

Inserting an Embedded Value

Inserting an Image

Exporting Panel Text to an HTML File

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.9 Defining Groups

You can perform the following tasks:

Inserting a Group

Importing a Saved Script as a Group

Defining the Group Name

See Also

Defining Group Pre- and Post-Actions

Getting Started with the Script Author

Obtaining Script Requirements Before Creating a Script

Defining Global Script Attributes

Working with Script Files
Working with the Tool Palette
Viewing Objects on the Canvas
Working With Objects on the Canvas
Defining Panels
Defining Blocks
Defining Branches
Controlling Script Flow
Defining Panel Answer Controls
Working With Answers
Defining Actions
Defining Commands
Deploying the Script
Using the Script Author
Troubleshooting the Script Author
Using the Scripting Engine
Taking a Survey
Using the Survey Admin Console

3.1.9.1 Inserting a Group

Use this procedure to insert a group onto the canvas. For more information on groups, see Defining Groups.

> **Caution:** Inserting a group creates a subgraph that must adhere to the minimum requirements for any graph. Ensure you define a Termination node in the group subgraph and include branching from the Start node to the Termination node, including branching to any other objects as appropriate.

Prerequisites

None

Steps

1. In the tool palette, click the **Group Insertion Mode** tool.

   When the pointer is over the canvas, the pointer is a pointing finger

2. On the canvas, click where you want to insert the group.

   The **Go down into child graph** button on the canvas toolbar is made active. If the Popup on Blob Creation option is selected, then the Properties dialog box for the object appears.

See Also

Setting Properties Dialog Box to Pop Up at Object Creation

Inserting a Panel

Inserting a Block

Inserting a Termination Node

Drilling Up or Down to a Related Script

Importing a Saved Script as a Group

Defining the Group Name

Defining Group Pre- and Post-Actions

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.9.2 Importing a Saved Script as a Group

Use this procedure to import a saved script workflow into the current script. The script workflow is imported as a group.

Prerequisites

None

Steps

1. Choose **File > Import**.

   The Open dialog box appears.

2. Locate the script that you want to import.

3. Click the file and then click **OK**.

   The script appears on the canvas as a group.

See Also

Inserting a Group

Defining the Group Name

Defining Group Pre- and Post-Actions

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.9.3 Defining the Group Name

Use this procedure to define the name of a group. The group name is how the group is identified in Script Author.

Prerequisites

Insert a group onto the canvas.

Steps

1. In the tool palette, click the **Toggle Select Mode** tool.

2. On the canvas, double-click a group.

   The Properties dialog box appears.

3. In the Group tree, select **Properties**.

4. In the Name field, type the name of the group.

5. Click **Apply** to apply your work or click **OK** to save your work and exit the Properties dialog box.

See Also

Defining Group Pre- and Post-Actions

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.10 Defining Blocks

Using blocks, you can integrate data sources into your script. With blocks you can query, update, or insert information in database tables, as well as associate commands. The block gives a clear visual indicator to the script developer that data source integration is occurring at that point in the script.

You can perform the following tasks:

Inserting a Block

Defining the Block Name

Defining a Database Query Block

Defining a Database Insert Block

Defining a Database Update Block

Defining Database Connections for a Block

Defining Database Tables for a Block

Defining Join Conditions for a Block

Defining Data Constraints for an Insert or Update Block

Defining Query Constraints for a Query Block

Defining Query Columns for a Query Block

Defining Panels within a Block

See Also

Defining Block Pre- and Post-Actions

Defining Panel Answer Controls

Defining Actions

Defining Commands

Getting Started with the Script Author

Obtaining Script Requirements Before Creating a Script

Defining Global Script Attributes

Working with Script Files

Working with the Tool Palette

Viewing Objects on the Canvas

Working With Objects on the Canvas

Defining Panels

Defining Groups

Defining Branches

Controlling Script Flow

Defining Panel Answer Controls

Working With Answers

Defining Actions

Defining Commands

Deploying the Script

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.10.1 Inserting a Block

Use this procedure to insert a block onto the canvas. For more information on blocks, see Defining Blocks.

> **Caution:** Inserting a block creates a subgraph that must adhere to the minimum requirements for any graph. Ensure you define a Termination node in the block subgraph and include branching from the Start node to the Termination node, including branching to any other objects as appropriate.

Prerequisites

None

Steps

1. In the tool palette, click the **Block Insertion Mode** tool.

   When the pointer is over the canvas, the pointer is a pointing finger

2. On the canvas, click where you want to insert the block.

   The **Go down into child graph** button on the canvas toolbar is made active. If the Popup on Blob Creation option is selected, then the Properties dialog box for the object appears.

See Also

Defining the Block Name

Defining Block Pre- and Post-Actions

Defining a Database Query Block

Defining a Database Insert Block

Defining a Database Update Block

Defining Database Connections for a Block

Defining Database Tables for a Block

Defining Join Conditions for a Block

Defining Data Constraints for an Insert or Update Block

Defining Query Constraints for a Query Block

Defining Query Columns for a Query Block

Defining Panels within a Block

3.1.10.2 Defining the Block Name

Use this procedure to define the name of a block. The block name is how the block is identified in Script Author.

Prerequisites

Insert a block onto the canvas.

Steps

1. In the tool palette, click the **Toggle Select Mode** tool.
2. On the canvas, double-click a block.

   The Properties dialog box appears.
3. In the Block tree, select **Properties**.
4. In the Name field, type the name of the block.
5. Click **Apply** to apply your work or click **OK** to save your work and exit the Properties dialog box.

See Also

Defining Block Pre- and Post-Actions

Defining a Database Query Block

Defining a Database Insert Block

Defining a Database Update Block

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.10.3 Defining a Database Query Block

Use this procedure to define an Oracle Scripting business object that searches data in a database and retrieves the records that match your search criteria.

Prerequisites

Insert a block onto the canvas. In addition, you need the following information:

- The names of the database tables that contain the information that you want to search or retrieve.
- The names of the table columns that contain the search criteria and that contain the information that you want to retrieve.
- The names of the primary keys and, if any, the foreign keys of the tables.

Steps

1. In the tool palette, click the **Toggle Select Mode** tool.
2. On the canvas, double-click a block.

   The Properties dialog box appears.
3. In the Block tree, select **Type**.
4. In the Type pane, select **Query Block**.
5. In the Block tree, select **Object dictionary**.
6. In the Connection/Table tab, do the following:
   a. Define how Oracle Scripting connects to the database at runtime. (See Defining Database Connections for a Block.)
   b. List the database tables that contain the information that you want to search or retrieve. (See Defining Database Tables for a Block.)
7. If you want to query data from more than one database table, then, in the Join Condition tab, indicate the relationship between related tables. (See Defining Join Conditions for a Block.)
8. In the Query Constraints tab,
   a. If you want to improve network performance during the query, then type the number of rows to be returned at a time during retrieval of the query result in the Prefetch Value field.

> **Note:** The row prefetch value is a feature of Oracle JDBC drivers. Standard JDBC drivers return the query result one row at a time.

b. Optionally, in the Primary Table field, type the name of the primary table.

c. If the rows retrieved by the query are to be locked for a subsequent update, then select **Query for update**.

d. Define your search criteria. (See Defining Query Constraints for a Query Block.)

e. List the information that you want to retrieve. (See Defining Query Columns for a Query Block.)

9. Click **Apply** to apply your work or click **OK** to save your work and exit the Properties dialog box.

See Also

Defining the Block Name

Defining Block Pre- and Post-Actions

Defining a Database Insert Block

Defining a Database Update Block

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.10.4 Defining a Database Insert Block

Use this procedure to define an Oracle Scripting business object that inserts a record into a database.

Prerequisites

Insert a block onto the canvas. In addition, you need the names of the database tables into which you want to insert a record.

Steps

1. In the tool palette, click the **Toggle Select Mode** tool.
2. On the canvas, double-click a block.

   The Properties dialog box appears.
3. In the Block tree, select **Type**.
4. In the Type pane, select **Insert Block**.
5. In the Block tree, select **Object dictionary**.
6. In the Connection/Table tab, do the following:
   a. Define how Oracle Scripting connects to the database at runtime. (See Defining Database Connections for a Block.)
   b. List the database tables into which you want to insert information. (See Defining Database Tables for a Block.)
7. If you want to insert data into more than one database table, then, in the Join Condition tab, indicate the relationship between related tables. (See Defining Join Conditions for a Block.)

   Note: If there is only one table in the block or if the tables in the block are not related, then you must still indicate the primary keys for each table.
8. If you want to insert a panel answer into a database table, then do the following:
   a. Drill down into the block
   b. Insert a panel in the block.
   c. Insert a Termination node onto the canvas.
   a. Draw branches between the objects to indicate the flow of the script.
   b. Define an answer. (See Defining Panel Answer Controls.)
   c. In the Answers pane, select the answer and then click **Data Dictionary**.

      The Data Dictionary dialog appears. The default data dictionary name is *answername*DataDict.

d. In the Table field, type the name of the table into which the answer is inserted.

e. In the Column field, type the name of the column into which the answer is inserted.

f. Click **OK** to exit the Data Dictionary dialog box.

Note: Answers must be listed according to the default order of the columns in the table.

---

g. Click **Apply** to apply your work or click **OK** to save your work and exit the Properties dialog box.

9. If you want to insert a panel answer into an additional table or to insert a value that is not represented by a panel answer into a table, then do the following:

a. In the Data Constraints tab of the insert block, click **Add**.

The Data Constraint dialog appears.

b. In the Command area, click Edit.

The Command dialog box appears.

c. Define a command that obtains or derives the value to be inserted. (See Defining Commands.)

d. In the Name field, type the name of the data constraint.

Note: After the command is executed, the value is stored on the Oracle Scripting blackboard under the name of the data constraint.

---

e. In the Table field, type the name of the table into which the value is inserted.

f. In the Column field, type the name of the column into which the value is inserted.

g. Click **OK** to exit the Data Constraints dialog box.

h. If you want to add another data constraint, then repeat steps a through g.

10. Click **Apply** to apply your work or click **OK** to save your work and exit the Properties dialog box.

See Also

Defining the Block Name

Defining Block Pre- and Post-Actions

Defining a Database Query Block

Defining a Database Update Block

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.10.5 Defining a Database Update Block

Use this procedure to define an Oracle Scripting business object that updates a record in a database.

Prerequisites

Insert a block onto the canvas. In addition, you need the following information:

- The names of the database tables that you want to update.

> **Note:** An update block must be preceded by a query block that retrieves at least one valid row.

Steps

1. In the tool palette, click the **Toggle Select Mode** tool.
2. On the canvas, double-click a block.

   The Properties dialog box appears.
3. In the Block tree, select **Type**.
4. In the Type pane, select **Update Block**.
5. In the Block tree, select **Object dictionary**.
6. In the Connection/Table tab, do the following:

a. Define how Oracle Scripting connects to the database at runtime. (See Defining Database Connections for a Block.)

b. List the database tables that contain the information that you want to update. (See Defining Database Tables for a Block.)

7. If you want to update data in more than one database table, then, in the Join Condition tab, indicate the relationship between related tables. (See Defining Join Conditions for a Block.)

Note: If there is only one table in the block or if the tables in the block are not related, then you must still indicate the primary keys for each table.

---

8. If you want to use a panel answer to update a database table, then do the following:

a. Drill down into the block

b. Insert a panel in the block.

c. Insert a Termination node onto the canvas.

d. Draw branches between the objects to indicate the flow of the script.

e. Define an answer. (See Defining Panel Answer Controls.)

f. In the Answers pane, select the answer and then click **Data Dictionary**.

The Data Dictionary dialog appears. The default data dictionary name is *answername*DataDict.

g. In the Table field, type the name of the table to be updated.

h. In the Column field, type the name of the column to be updated.

i. Click **OK** to exit the Data Dictionary dialog box.

Note: Answers must be listed according to the default order of the columns in the table.

---

j. Click **Apply** to apply your work or click **OK** to save your work and exit the Properties dialog box.

9. If you want to update an additional table with a panel answer or to update a table with a value that is not represented by a panel answer, then do the following:

   a. In the Data Constraints tab of the update block, click **Add**.

      The Data Constraint dialog appears.

   b. In the Command area, click Edit.

      The Command dialog box appears.

   c. Define a command that obtains or derives the value used to update the table. (See Defining Commands.)

   d. In the Name field, type the name of the data constraint.

      Note: After the command is executed, the value is stored on the Oracle Scripting blackboard under the name of the data constraint.

   e. In the Table field, type the name of the table to be updated.

   f. In the Column field, type the name of the column to be updated.

   g. Click **OK** to exit the Data Constraints dialog box.

   h. If you want to add another data constraint, then repeat steps a through g.

10. Click **Apply** to apply your work or click **OK** to save your work and exit the Properties dialog box.

See Also

Defining the Block Name

Defining Block Pre- and Post-Actions

Defining a Database Query Block

Defining a Database Insert Block

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.10.6 Defining Database Connections for a Block

Use this procedure to define how Oracle Scripting connects to the database at runtime prior to the execution of the block.

Prerequisites

Insert a block onto the canvas.

Steps

1. In the tool palette, click the **Toggle Select Mode** tool.
2. On the canvas, double-click a block.

   The Properties dialog box appears.
3. In the Block tree, select **Object dictionary**.
4. If you want to define new connection parameters, then, in the Connection/Tables tab, do the following:
   a. If necessary, clear the **Reuse an existing connection** box.
   b. Type the connection parameters. (See Database Connection Guidelines.)

   Note: When you save your work in the Properties dialog box, the connection parameters are available in the Existing Connections list for this script and can be reused in another block or when deploying the script.
5. If you want to reuse an existing connection, then, in the Connections/Tables tab, do the following:
   a. Select the **Reuse an existing connection** box.
   b. From the Existing Connections list, select a connection.
6. Click **Apply** to apply your work or click **OK** to save your work and exit the Properties dialog box.

See Also

Defining Database Tables for a Block

Defining Join Conditions for a Block

Defining Data Constraints for an Insert or Update Block

Defining Query Constraints for a Query Block

Defining Query Columns for a Query Block

Defining Panels within a Block

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.10.7 Defining Database Tables for a Block

Use this procedure to list the tables that you want to query, update, or insert.

Prerequisites

Insert a block onto the canvas. In addition, you need the names of the database tables that contain the information that you want to query, update, or insert.

Steps

1. In the tool palette, click the **Toggle Select Mode** tool.
2. On the canvas, double-click a block.

   The Properties dialog box appears.
3. In the Block tree, select **Object dictionary**.
4. In the Connections/Table tab in the Tables area, click **Add table**.

   The Table dialog box appears.
5. Type the name of the table and then click **OK** to exit the Table dialog box.
6. If you want to add another table, then repeat steps 4 through 5.

> **Note:** List master tables first and then tables related by foreign key.

7. Click **Apply** to apply your work or click **OK** to save your work and exit the Properties dialog box.

See Also

Defining Database Connections for a Block

Defining Join Conditions for a Block

Defining Data Constraints for an Insert or Update Block

Defining Query Constraints for a Query Block

Defining Query Columns for a Query Block

Defining Panels within a Block

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.10.8 Defining Join Conditions for a Block

Use this procedure to indicate the links between related tables. Tables are joined according to common values existing in corresponding columns.

> **Note:** If there is only one table in the block or if the tables in the block are not related, then you must still indicate the primary keys for each table.

The primary key is a column or set of columns that uniquely identifies each row of a table. The foreign key is a column or set of columns that references a primary or unique key in the same or a different table.

Prerequisites

Insert a block onto the canvas. In addition, you need the names of the primary keys and foreign keys, if any, of the tables that you want to query, update, or insert.

Steps

1. In the tool palette, click the **Toggle Select Mode** tool.
2. On the canvas, double-click a block.

   The Properties dialog box appears.
3. In the Block tree, select **Object dictionary**.
4. In the Join Condition tab, click **Add**.

   The Join Condition Dialog dialog box appears.
5. In the Master PK Table tab, do the following:
   a. In the Master Table field, type the name of the table.
   b. Click **Add Master PK**.

      The Column Entry dialog box appears.

      Note: Prefix the column name with the table name (for example, *table1.column1*).

      ---

   c. Type the name of the column that is the primary key of the master table and then click **OK** to exit the Column Entry dialog box.
   d. If you want to add another primary key, then repeat steps b through c.
6. In the Detail PK Table tab, do the following:
   a. In the Detail Table field, type the name of the related table.
   b. Click **Add Detail PK**.

   The Column Entry dialog box appears.

   Note: Prefix the column name with the table name (for example, *table1.column1*).

   ---

   c. Type the name of the column that is the primary key of the related table and then click **OK** to exit the Column Entry dialog box.
   d. If you want to add another primary key, then repeat steps b through c.
7. In the Detail FK Table tab, do the following:

a. Click **Add Detail FK**.

The Column Entry dialog box appears.

b. Type the name of the column that links the tables and then click **OK** to exit the Column Entry dialog box.

c. If you want to add another column name, then repeat steps a through b.

8. Click **OK** to exit the Join Condition Dialog dialog box.
9. Click **Apply** to apply your work or click **OK** to save your work and exit the Properties dialog box.

See Also

Defining Database Connections for a Block

Defining Database Tables for a Block

Defining Data Constraints for an Insert or Update Block

Defining Query Constraints for a Query Block

Defining Query Columns for a Query Block

Defining Panels within a Block

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.10.9 Defining Data Constraints for an Insert or Update Block

Use this procedure to list the columns to be inserted or updated.

Prerequisites

Insert a block onto the canvas. In addition, you need the names of the database tables that contain the information that you want to update, or insert.

Steps

1. In the tool palette, click the **Toggle Select Mode** tool.

2. On the canvas, double-click an insert or update block.

   The Properties dialog box appears.

3. In the Block tree, select **Object dictionary**.
4. In the Data Constraints tab, click **Add**.

   The Data Constraints dialog box appears.

5. Define the data constraints.
6. Click **OK** to exit the Data Constraints dialog box.
7. Click **Apply** to apply your work or click **OK** to save your work and exit the Properties dialog box.

See Also

Defining Database Connections for a Block

Defining Database Tables for a Block

Defining Join Conditions for a Block

Defining Query Constraints for a Query Block

Defining Query Columns for a Query Block

Defining Panels within a Block

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.10.10 Defining Query Constraints for a Query Block

Use this procedure to define your search criteria for the database query.

Prerequisites

Insert a block onto the canvas. In addition, you need the names of the tables and table columns that contain the information that you want to query.

Steps

1. In the tool palette, click the **Toggle Select Mode** tool.
2. On the canvas, double-click a query block.

   The Properties dialog box appears.
3. In the Block tree, select **Object Dictionary**.
4. In the Query Data tab, in the Query Constraints tab, click Add Constraint.

   The Query Constraint Input dialog box appears.
5. Type the query constraint (for example, *table1.column1 operator value*).

   Note: Conditions based on user input require a panel in the block. The *value* is an answer name in the panel. See Defining Panels within a Block.
6. Click **OK** to exit the Query Constraint Input dialog box.
7. If you want to add another query constraint, then repeat steps 4 through 6.

   Note: Insert logical operators (such as, `NOT`, `AND`, and `OR`) at the end of the query constraint. If no logical operator is used, then `AND` is assumed. Use parentheses at the beginning or end of a query constraint to override the rules of precedence.
8. Click **Apply** to apply your work or click **OK** to save your work and exit the Properties dialog box.

See Also

Defining Database Connections for a Block

Defining Database Tables for a Block

Defining Join Conditions for a Block

Defining Data Constraints for an Insert or Update Block

Defining Query Columns for a Query Block

Defining Panels within a Block

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.10.11 Defining Query Columns for a Query Block

Use this procedure to list the information that you want to retrieve from the database.

Prerequisites

Insert a block onto the canvas. In addition, you need the names of the tables and table columns that contain the information that you want to query.

Steps

1. In the tool palette, click the **Toggle Select Mode** tool.
2. On the canvas, double-click a query block.

   The Properties dialog box appears.
3. In the Block tree, select **Object Dictionary**.
4. In the Query Data tab, in the Query Columns tab, click Add Columns.

   The Enter a Key/Value dialog box appears.
5. In the Name field, type the column name.

   Note: Prefix the column name with the table name (for example, ***table1.column1***).
6. If you want to improve network performance during the query, then select the type of data in the column from the Data Type list.
7. Click **OK** to exit the Enter a Key/Value dialog box.
8. If you want to add another column, then repeat steps 4 through 7.
9. Click **Apply** to apply your work or click **OK** to save your work and exit the Properties dialog box.

See Also

Defining Database Connections for a Block

Defining Database Tables for a Block

Defining Join Conditions for a Block

Defining Data Constraints for an Insert or Update Block

Defining Query Constraints for a Query Block

Defining Panels Within a Block

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.10.12 Defining Panels Within a Block

Use this procedure to add a panel to a block. A block serves as a method to add a database function such as select (query), insert or update statement, or to add a command such as an API. It also serves as a container, creating a subgraph that may contain panels, groups or other blocks.

> **Note:** A block creates a subgraph on the canvas. This subgraph must follow all the syntactical rules of any graph in a script. At minimum, it must contain a Termination node and default branching between the Start node and the Termination node.

Prerequisites

Insert a block onto the canvas.

Steps

1. In the tool palette, click the **Toggle Select Mode** tool.
2. On the canvas, click a block and then click the **Go down into child graph** button on the toolbar.

   The graph for the block appears.

3. Define the graph, adding panels (or other objects, including.

See Also

Defining Database Connections for a Block

Defining Database Tables for a Block

Defining Join Conditions for a Block

Defining Data Constraints for an Insert or Update Block

Defining Query Constraints for a Query Block

Defining Query Columns for a Query Block

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.11 Defining Branches

Branches directly control the flow of a script in runtime. A branch must always begin from an existing Script Author object on the canvas. All branches but the Indeterminate branch must also end at another scripting object on the canvas.

You can perform the following tasks:

Setting Properties Dialog Box to Pop Up at Branch Creation

Defining a Default Branch

Defining a Distinct Branch

Defining a Conditional Branch

Defining an Indeterminate Branch

Defining the Branch Name

Reordering Branches

Working with Branches on the Canvas

See Also

Getting Started with the Script Author

Obtaining Script Requirements Before Creating a Script

Defining Global Script Attributes

Working with Script Files

Working with the Tool Palette

Viewing Objects on the Canvas

Working With Objects on the Canvas

Defining Panels

Defining Groups

Defining Blocks

Controlling Script Flow

Defining Panel Answer Controls

Working With Answers

Defining Actions

Defining Commands

Deploying the Script

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.11.1 Setting Properties Dialog Box to Pop Up at Branch Creation

Use this procedure to set up Script Author to automatically display the Properties dialog box when you insert the branch onto the canvas.

Prerequisites

None

Steps

- Choose **View** > **Popup on Branch Creation**.

See Also

Defining a Default Branch

Defining a Distinct Branch

Defining a Conditional Branch

Defining an Indeterminate Branch

Defining the Branch Name

Reordering Branches

Working with Branches on the Canvas

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.11.2 Defining a Default Branch

Use this procedure to navigate the script to a default destination. This type of branch is also used when there is only one destination.

Prerequisites

Insert a panel onto the canvas.

Steps

1. In the tool palette, click the **Default Branch Mode** tool.

   When the pointer is over the canvas, the pointer is a cross hair (+).

2. On the canvas, drag the pointer from the source object to the destination object.

   When you release the pointer over the destination object, the branch arrow appears. The branch is red. This indicates that the branch is currently selected.

See Also

Setting Properties Dialog Box to Pop Up at Branch Creation

Defining a Distinct Branch

Defining a Conditional Branch

Defining an Indeterminate Branch

Defining the Branch Name

Reordering Branches

Working with Branches on the Canvas

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.11.3 Defining a Distinct Branch

Use this procedure to navigate the script to an object based on an answer selected in the preceding panel.

> **Caution:** Inserting a distinct branch requires a value to be defined for that branch, as described below. If omitted, the script will not be syntactically correct.

Prerequisites

- Insert a panel onto the canvas.
- Insert a distinct branch.
- Define an answer control for the script panel.

Steps

1. In the tool palette, click the **Toggle Select Mode** tool.
2. On the canvas, double-click a distinct branch.

   The Properties dialog box appears.

3. In the Distinct tree, select **Value**.

   In the Value pane, the Current Unused Answers list contains the display values for the default answer defined in the source panel that have not been assigned to a distinct branch.

4. In the Value pane, do the following:

   a. In the Current Unused Answers list, click the answer that triggers the branch.

   b. Click the double right-arrow button to move the answer to the Selected Answer list.

5. Click **Apply** to apply your work or click **OK** to save your work and exit the Properties dialog box.

See Also

Setting Properties Dialog Box to Pop Up at Branch Creation

Defining a Default Branch

Defining a Conditional Branch

Defining an Indeterminate Branch

Defining the Branch Name

Reordering Branches

Working with Branches on the Canvas

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.11.4 Defining a Conditional Branch

Use this procedure to navigate the script to an object based on the evaluation of a conditional expression as true.

Caution: Inserting a distinct branch requires a value to be defined for that branch, as described below. If omitted, the script will not be syntactically correct.

Note: The custom code referenced by the command defined for the conditional branch is not verified to exist in order to check the syntax of the script. Nonetheless, the referenced custom code must be available in runtime in order for the script to execute. Custom code must be available to the base Java classes that provide script runtime functionality. Thus, any custom code must be appropriately deployed to the server and any configuration files adjusted accordingly so the custom code can be referenced. For more information, contact your systems administrator or refer to the *Oracle Scripting Implementation Guide*.

Prerequisites

- Insert a panel onto the canvas.
- Insert a conditional branch.

Steps

1. In the tool palette, click the **Toggle Select Mode** tool.
2. On the canvas, double-click a conditional branch.

   The Properties dialog box appears.
3. In the Conditional tree, select **Condition**.
4. In the Condition pane, click **Edit**.

   The Command dialog box appears.
5. Define the condition as a command.

   The command must return true or false.
6. Click **OK** to exit the Command dialog box.
7. Click **Apply** to apply your work or click **OK** to save your work and exit the Properties dialog box.

See Also

Setting Properties Dialog Box to Pop Up at Branch Creation

Defining a Default Branch

Defining a Distinct Branch

Defining an Indeterminate Branch

Defining the Branch Name

Reordering Branches

Working with Branches on the Canvas

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.11.5 Defining an Indeterminate Branch

Use this procedure to navigate the script based on the evaluation of an expression that returns the name of the destination object.

Caution: Inserting an indeterminate branch requires an expression to be defined for that branch, as described below. If omitted, the script will not be syntactically correct.

Note: The custom code referenced by the command defined for the indeterminate branch is not verified to exist in order to check the syntax of the script. Nonetheless, the referenced custom code must be available in runtime in order for the script to execute. Custom code must be available to the base Java classes that provide script runtime functionality. Thus, any custom code must be appropriately deployed to the server and any configuration files adjusted accordingly so the custom code can be referenced. For more information, contact your systems administrator or refer to the *Oracle Scripting Implementation Guide*.

Prerequisites

Insert an indeterminate branch.

Steps

1. In the tool palette, click the **Toggle Select Mode** tool.
2. On the canvas, double-click an indeterminate branch.

   The Properties dialog box appears.
3. In the Indeterminate tree, select **Expression**.
4. In the Expression pane, click **Edit**.

   The Command dialog box appears.
5. Define the indeterminate expression as a command.

   The command must return the name of:

   a. an "sibling" object (a panel, block or group on the same graph as the source object)

   b. the Shortcut name associated with a group on any hierarchical level of the script
6. Click **OK** to exit the Command dialog box.
7. Click **Apply** to apply your work or click **OK** to save your work and exit the Properties dialog box.

See Also

Setting Properties Dialog Box to Pop Up at Branch Creation

Defining a Default Branch

Defining a Distinct Branch

Defining a Conditional Branch

Defining the Branch Name

Reordering Branches

Working with Branches on the Canvas

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.11.6 Defining the Branch Name

Use this procedure to define the name of the branch. Except for default branches, the branch name appears with the branch in the script canvas.

Prerequisites

Insert a branch onto the canvas.

Steps

1. In the tool palette, click the **Toggle Select Mode** tool.
2. On the canvas, double-click a branch.

   The Properties dialog box appears.
3. In the Panel tree, select **Properties**.
4. In the Name field, type the name of the branch.
5. Click **Apply** to apply your work or click **OK** to save your work and exit the Properties dialog box.

See Also

Setting Properties Dialog Box to Pop Up at Branch Creation

Defining a Default Branch

Defining a Distinct Branch

Defining a Conditional Branch

Defining an Indeterminate Branch

Reordering Branches

Working with Branches on the Canvas

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.11.7 Reordering Branches

Use this procedure to place branches in the order in which they are to be evaluated.

Prerequisites

- Insert a panel onto the canvas.
- Insert branches onto the canvas.

Steps

1. In the tool palette, click the **Toggle Select Mode** tool.
2. On the canvas, double-click a panel.

   The Properties dialog box appears.
3. In the Panel tree, select **Branches**.
4. In the Branches pane, select a branch.
5. If you want the branch to be evaluated at runtime earlier than any other branch in the list, then click **Move Up**.
6. If you want the branch to be evaluated at runtime later than any other branch in the list, then click **Move Down**.
7. Click **Apply** to apply your work or click **OK** to save your work and exit the Properties dialog box.

See Also

Setting Properties Dialog Box to Pop Up at Branch Creation

Defining a Default Branch

Defining a Distinct Branch

Defining a Conditional Branch

Defining an Indeterminate Branch

Defining the Branch Name

Working with Branches on the Canvas

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.11.8 Working with Branches Visually on the Canvas

Once a branch has been defined on the canvas, it can be modified visually to accommodate the objects placed on the canvas and create a clear picture of the intended flow of the script. Working with branches, you can perform the following tasks:

- Inserting a Straight Branch
- Inserting a Branch with Corners
- Adding a Vertex to a Branch

Adding a Corner to an Existing Branch

Moving a Corner of a Branch

Deleting a Corner from a Branch

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.11.8.2 Inserting a Branch with Corners Use this procedure to insert a branch with corners.

Prerequisites

To insert a branch, you must have at least two object on the canvas.

Steps

1. In the tool palette, click a branch tool.

   When the pointer is over the canvas, the pointer is a cross hair (+).

2. On the canvas, drag the pointer from the source object to the desired location of the first corner and release the pointer.

3. Move the pointer to the desired location of the next corner and click. The line is drawn automatically.

4. Click the destination object.

   When you click destination object, the branch arrow appears. The branch is red. This indicates that the branch is currently selected.

See Also

Setting Properties Dialog Box to Pop Up at Branch Creation

Inserting a Straight Branch

Adding a Corner to an Existing Branch

Moving a Corner of a Branch

Deleting a Corner from a Branch

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.11.8.3 Adding a Corner to an Existing Branch Use this procedure to add a corner to an existing branch.

Prerequisites

None

Steps

1. In the tool palette, click the **Toggle Select Mode** tool.
2. On the canvas, select a branch.

   When the branch is selected, the branch is red.
3. Drag the branch to the desired location and then release the pointer.

   When you release the pointer, a corner is created on the branch.

See Also

Inserting a Straight Branch

Inserting a Branch with Corners

Moving a Corner of a Branch

Deleting a Corner from a Branch

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.11.8.4 Moving a Corner of a Branch Use this procedure to move a corner of a branch.

Prerequisites

None

Steps

1. In the tool palette, click the **Toggle Select Mode** tool.
2. On the canvas, select a branch.
3. Select a branch corner.

   When the pointer is over the branch corner, the pointer is a move icon. When the branch corner is selected, the branch is no longer red.
4. Drag the branch to the desired location and then release the pointer.

See Also

Inserting a Straight Branch

Inserting a Branch with Corners

Adding a Corner to an Existing Branch

Deleting a Corner from a Branch

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.11.8.5 Deleting a Corner from a Branch Use this procedure to delete a corner from a branch.

Prerequisites

None

Steps

1. In the tool palette, click the **Toggle Select Mode** tool.
2. On the canvas, select a branch.
3. Select a branch corner.

When the pointer is over the branch corner, the pointer is a move icon. When the branch corner is selected, the branch is no longer red.

4. Choose **Edit** > **Delete**.

See Also

Inserting a Straight Branch

Inserting a Branch with Corners

Adding a Corner to an Existing Branch

Moving a Corner of a Branch

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.12 Controlling Script Flow

Script flow is controlled by several aspects. The first and most obvious method is defining branches. However, script flow is also controlled by other aspects, such as the following, each with its own section:

Defining Panel Answer Controls

Defining Actions

Defining Blocks

Defining Commands

See Also

Getting Started with the Script Author

Obtaining Script Requirements Before Creating a Script

Defining Global Script Attributes

Working with Script Files

Working with the Tool Palette

Viewing Objects on the Canvas

Working With Objects on the Canvas

Defining Panels

Defining Groups

Defining Blocks

Defining Branches

Defining Panel Answer Controls

Working With Answers

Deploying the Script

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.13 Defining Panel Answer Controls

Answer controls are the user interface controls that display in a script at runtime. Every panel requires at least one answer definition or node to be defined. Each node may contain a single user interface type and may have text associated with the answer control. The end user (interaction center agent or survey respondent) must interact with an answer control for each panel, whether the panel contains a single node or multiple nodes. If an answer definition includes validation, that panel at runtime requires the end user to interact with that answer control following the parameters dictated by the code behind the validation command.

You can perform the following tasks:

Defining a Text Field Answer Control in a Panel

Defining a Text Area Answer Control in a Panel

Defining a Radio Button Group Answer Control in a Panel

Defining a Check Box Group Answer Control in a Script Panel

Defining a Button Answer Control in a Panel

Defining a Drop Down List Answer Control in a Panel

Defining a Password Answer Control in a Panel

See Also

Getting Started with the Script Author

Obtaining Script Requirements Before Creating a Script

Defining Global Script Attributes

Working with Script Files

Working with the Tool Palette

Viewing Objects on the Canvas

Working With Objects on the Canvas

Defining Panels

Defining Groups

Defining Blocks

Defining Branches

Controlling Script Flow

Working With Answers

Defining Actions

Defining Commands

Deploying the Script

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.13.1 Defining a Text Field Answer Control in a Panel

Use this procedure to define a text field answer control in a script panel. A text field is a text entry field that displays a single row for text entry. The text field is generally intended for use to capture text answers of one line or less in runtime.

Beyond the space provided, the end user can continue to enter characters with no limit. There is no restriction to the content allowed in a text field unless validation is associated with this answer node.

Prerequisites

Insert a panel onto the canvas.

Steps

1. In the tool palette, click the **Toggle Select Mode** tool.
2. On the canvas, double-click a panel.

   The Properties dialog box appears.
3. In the Panel tree, select **Answers**.
4. In the Answers pane, click **Add**.

   The Answer Entry Dialog dialog box appears.
5. If this is the only answer control for this panel or if this answer control is used for distinct branching, then select **Default for Distinct Branching**.

   Note: Every panel is required to have at least one answer defined, and at least one answer definition (or "node") per panel *must* be designated as the default answer for distinct branching. This is true regardless of whether distinct branching is used by the panel.
6. In the Name field, type the name of the answer.

   Note: The answer name must be unique.
7. From the UI Type list, select **Text Field**.
8. In the UI Label field, type the label that appears next to the text field in the script panel at runtime.
9. Click **OK** to exit the Answer Entry Dialog dialog box.

   The answer appears in the Answer pane in the Properties dialog box.
10. If you want to define another text field in the same script panel, then repeat steps 4 through 9.

11. Click **Apply** to apply your work or click **OK** to save your work and exit the Properties dialog box.

See Also

Defining a Text Area Answer Control in a Panel

Defining a Radio Button Group Answer Control in a Panel

Defining a Check Box Group Answer Control in a Script Panel

Defining a Button Answer Control in a Panel

Defining a Drop Down List Answer Control in a Panel

Defining a Password Answer Control in a Panel

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.13.2 Defining a Text Area Answer Control in a Panel

Use this procedure to define a text area answer control in a script panel. A text area is a text entry field that displays multiple rows for text entry. The text area is generally intended for use to capture text answers longer than one line in runtime. Beyond the space provided, the end user can continue to enter characters with no limit. There is no restriction to the content allowed in a text field unless validation is associated with this answer node.

Prerequisites

Insert a panel onto the canvas.

Steps

1. In the tool palette, click the **Toggle Select Mode** tool.
2. On the canvas, double-click a panel.

   The Properties dialog box appears.
3. In the Panel tree, select **Answers**.

4. In the Answers pane, click **Add**.

   The Answer Entry Dialog dialog box appears.

5. If this is the only answer control for this panel or if this answer control is used for distinct branching, then select **Default for Distinct Branching**.

   Note: Every panel is required to have at least one answer defined, and at least one answer definition (or "node") per panel *must* be designated as the default answer for distinct branching. This is true regardless of whether distinct branching is used by the panel.

6. In the Name field, type the name of the answer.

   Note: The answer name must be unique.

7. From the UI Type list, select **Text Area**.
8. In the UI Label field, type the label that appears next to the text area in the script panel at runtime.
9. Click **OK** to exit the Answer Entry Dialog dialog box.

   The answer appears in the Answer pane in the Properties dialog box.

10. If you want to define another text area in the same script panel, then repeat steps 4 through 9.
11. Click **Apply** to apply your work or click **OK** to save your work and exit the Properties dialog box.

See Also

Defining a Text Field Answer Control in a Panel

Defining a Radio Button Group Answer Control in a Panel

Defining a Check Box Group Answer Control in a Script Panel

Defining a Button Answer Control in a Panel

Defining a Drop Down List Answer Control in a Panel

Defining a Password Answer Control in a Panel

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.13.3 Defining a Radio Button Group Answer Control in a Panel

Use this procedure to define one or more radio button groups in a script panel.

Prerequisites

Insert a panel onto the canvas.

Steps

1. In the tool palette, click the **Toggle Select Mode** tool.
2. On the canvas, double-click a panel.

   The Properties dialog box appears.
3. In the Panel tree, select **Answers**.
4. In the Answers pane, click **Add**.

   The Answer Entry Dialog dialog box appears.
5. If this is the only answer control for this panel or if this answer control is used for distinct branching, then select **Default for Distinct Branching**.

   Note: Every panel is required to have at least one answer defined, and at least one answer definition (or "node") per panel *must* be designated as the default answer for distinct branching. This is true regardless of whether distinct branching is used by the panel.
6. In the Name field, type the name of the answer.

   Note: The answer name must be unique.
7. From the UI Type list, choose **Radio Button**.

8. Optionally, in the UI Label field, type the label that appears next to the radio button group in the script panel at runtime.
9. Click **Edit Data Dictionary**.

   The Edit Data Dictionary dialog box appears.
10. In the Lookups tab, select **Specify lookups** and then click **Add**.

    The Lookup Entry dialog box appears.
11. In the Display Value field, type the text that appears next to the radio button in the script panel at runtime.
12. In the Value field, type the value that is stored in the Oracle Scripting database schema when the radio button is selected at runtime.
13. If you want to add another radio button to the group, then repeat steps 10 through 12.
14. Click **OK** to exit the Lookup Entry window.

    The lookup entry values you provided as choices for this answer in runtime appear in the Specify Lookups window.
15. Click **OK** to exit the Edit Data Dictionary dialog box.
16. Click **OK** to exit the Answer Entry dialog box.
17. If you want to define another radio button group in the same script panel, then repeat steps 4 through 15.
18. Click **Apply** to apply your work or click **OK** to save your work and exit the Properties dialog box.

See Also

Defining a Text Field Answer Control in a Panel

Defining a Text Area Answer Control in a Panel

Defining a Check Box Group Answer Control in a Script Panel

Defining a Button Answer Control in a Panel

Defining a Drop Down List Answer Control in a Panel

Defining a Password Answer Control in a Panel

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.13.4 Defining a Check Box Group Answer Control in a Panel

Use this procedure to define a one or more check boxes in a script panel.

Prerequisites

Insert a panel onto the canvas.

Steps

1. In the tool palette, click the **Toggle Select Mode** tool.
2. On the canvas, double-click a panel.

   The Properties dialog box appears.
3. In the Panel tree, select **Answers**.
4. In the Answers pane, click **Add**.

   The Answer Entry Dialog dialog box appears.
5. If this is the only answer control for this panel or if this answer control is used for distinct branching, then select **Default for Distinct Branching**.

   Note: Every panel is required to have at least one answer defined, and at least one answer definition (or "node") per panel *must* be designated as the default answer for distinct branching. This is true regardless of whether distinct branching is used by the panel.
6. In the Name field, type the name of the answer.

   Note: The answer name must be unique.
7. From the UI Type list, choose **Check Box**.
8. In the UI Label field, type the label that appears next to the check box in the script panel at runtime.

9. Click **OK** to exit the Answer Entry Dialog dialog box.

   The answer appears in the Answer pane in the Properties dialog box.

10. If you want to define another check box in the same script panel, then repeat steps 4 through 9.

11. Click **Apply** to apply your work or click **OK** to save your work and exit the Properties dialog box.

See Also

Defining a Text Field Answer Control in a Panel

Defining a Text Area Answer Control in a Panel

Defining a Radio Button Group Answer Control in a Panel

Defining a Button Answer Control in a Panel

Defining a Drop Down List Answer Control in a Panel

Defining a Password Answer Control in a Panel

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.13.5 Defining a Button Answer Control in a Panel

Use this procedure to define a buttons in a script panel.

> **Caution:** The button UI type answer control must only be used in panels that contain *a single node or answer definition.* Use of buttons for more than one node in a single panel *is not supported.*

Prerequisites

- Insert a panel onto the canvas.
- Ensure only one answer definition is required in this panel.

Steps

1. In the tool palette, click the **Toggle Select Mode** tool.
2. On the canvas, double-click a panel.

   The Properties dialog box appears.
3. In the Panel tree, select **Answers**.
4. In the Answers pane, click **Add**.

   The Answer Entry Dialog dialog box appears.
5. If this is the only answer control for this panel or if this answer control is used for distinct branching, then select **Default for Distinct Branching**.

   Note: Every panel is required to have at least one answer defined, and at least one answer definition (or "node") per panel *must* be designated as the default answer for distinct branching. This is true regardless of whether distinct branching is used by the panel.
6. In the Name field, type the name of the answer.

   Note: The answer name must be unique.
7. From the UI Type list, choose **Button**.
8. Click **OK** to exit the Answer Entry Dialog dialog box.

   The answer appears in the Answer pane in the Properties dialog box.
9. In the Answers pane, select the answer and then click **Edit Data Dictionary**.

   The Edit Data Dictionary dialog box appears.
10. In the Lookups tab, select **Specify lookups** and then click **Add**.

    The Lookup Entry dialog box appears.
11. In the Display Value field, type the text that appears on the button in the script panel at runtime.
12. In the Value field, type the value that is stored in the Oracle Scripting database schema when the button is selected at runtime.
13. Click **OK** to exit the Lookup Entry window.

14. Click **OK** to exit the Edit Data Dictionary dialog box.
15. Click **Apply** to apply your work or click **OK** to save your work and exit the Properties dialog box.

See Also

Defining a Text Field Answer Control in a Panel

Defining a Text Area Answer Control in a Panel

Defining a Radio Button Group Answer Control in a Panel

Defining a Check Box Group Answer Control in a Script Panel

Defining a Drop Down List Answer Control in a Panel

Defining a Password Answer Control in a Panel

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.13.6 Defining a Drop Down List Answer Control in a Panel

Use this procedure to define one or more drop down lists in a script panel.

Prerequisites

Insert a panel onto the canvas.

Steps

1. In the tool palette, click the **Toggle Select Mode** tool.
2. On the canvas, double-click a panel.

   The Properties dialog box appears.
3. In the Panel tree, select **Answers**.
4. In the Answers pane, click **Add**.

   The Answer Entry Dialog dialog box appears.

5. If this is the only answer control for this panel or if this answer control is used for distinct branching, then select **Default for Distinct Branching**.

   Note: Every panel is required to have at least one answer defined, and at least one answer definition (or "node") per panel *must* be designated as the default answer for distinct branching. This is true regardless of whether distinct branching is used by the panel.

6. In the Name field, type the name of the answer.

   Note: The answer name must be unique.

7. From the UI Type list, choose **Drop Down**.
8. Optionally, in the UI Label field, type the label that appears next to the drop down list in the script panel at runtime.
9. Click **Edit Data Dictionary**.

   The Edit Data Dictionary dialog box appears.

10. In the Lookups tab, select **Specify lookups** and then click **Add**.

    The Lookup Entry dialog box appears.

11. In the Display Value field, type the text that appears in the drop down list in the script panel at runtime.
12. In the Value field, type the value that is stored in the Oracle Scripting database schema when the drop down is selected at runtime.
13. If you want to add another item to the drop down list, then repeat steps 10 through 13.
14. Click **OK** to exit the Lookup Entry window.

    The lookup entry values you provided as choices for this answer in runtime appear in the Specify Lookups window.

15. Click **OK** to exit the Edit Data Dictionary dialog box.
16. Click **OK** to exit the Answer Entry Dialog dialog box.

    The answer appears in the Answer pane in the Properties dialog box.

17. If you want to define another drop down list in the same script panel, then repeat steps 4 through 15.
18. If you want to define a validation command for this password field, follow the procedure in Adding Validation to an Answer Definition.
19. Click **Apply** to apply your work or click **OK** to save your work and exit the Properties dialog box.

See Also

Defining a Text Field Answer Control in a Panel

Defining a Text Area Answer Control in a Panel

Defining a Radio Button Group Answer Control in a Panel

Defining a Check Box Group Answer Control in a Script Panel

Defining a Button Answer Control in a Panel

Defining a Password Answer Control in a Panel

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.13.7 Defining a Password Answer Control in a Panel

Use this procedure to define a password answer control in a script panel. A password field is a single-row text entry field that will display the input as asterisks at runtime. Any password functionality such as validation against a valid range of values must be developed independently and associated with the password answer control using the Validation command in the data dictionary. This feature is not automatically included in this answer UI type.

Prerequisites

Insert a panel onto the canvas.

Steps

1. In the tool palette, click the **Toggle Select Mode** tool.

2. On the canvas, double-click a panel.

   The Properties dialog box appears.

3. In the Panel tree, select **Answers**.

4. In the Answers pane, click **Add**.

   The Answer Entry Dialog dialog box appears.

5. If this is the only answer control for this panel or if this answer control is used for distinct branching, then select **Default for Distinct Branching**.

   Note: Every panel is required to have at least one answer defined, and at least one answer definition (or "node") per panel *must* be designated as the default answer for distinct branching. This is true regardless of whether distinct branching is used by the panel.

6. In the Name field, type the name of the answer.

   Note: The answer name must be unique.

7. From the UI Type list, select **Password**.

8. In the UI Label field, type the label that appears next to the text field in the script panel at runtime.

9. Click **OK** to exit the Answer Entry Dialog dialog box.

   The answer appears in the Answer pane in the Properties dialog box.

10. If you want to define another password in the same script panel, then repeat steps 4 through 9.

11. If you want to add validation to this password, see Adding Validation to an Answer Definition.

12. Click **Apply** to apply your work or click **OK** to save your work and exit the Properties dialog box.

See Also

Defining a Text Field Answer Control in a Panel

Defining a Text Area Answer Control in a Panel

Defining a Radio Button Group Answer Control in a Panel

Defining a Check Box Group Answer Control in a Script Panel

Defining a Button Answer Control in a Panel

Defining a Drop Down List Answer Control in a Panel

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.14 Working With Answers

After defining an answer for a panel, you can perform the following procedures:

Defining Vertical or Horizontal Answer Control Layout

Adding Validation to an Answer Definition

Defining Data Constraints for an Answer Control

Reordering Answer Options

Reordering Answer Controls

Deleting Answer Options from an Answer Control

Deleting Answer Controls from a Script Panel

Substituting a Java Bean for an Answer

See Also

Getting Started with the Script Author

Obtaining Script Requirements Before Creating a Script

Defining Global Script Attributes

Working with Script Files

Working with the Tool Palette

Viewing Objects on the Canvas

Working With Objects on the Canvas

Defining Panels
Defining Groups
Defining Blocks
Defining Branches
Controlling Script Flow
Defining Panel Answer Controls
Defining Actions
Defining Commands
Deploying the Script
Using the Script Author
Troubleshooting the Script Author
Using the Scripting Engine
Taking a Survey
Using the Survey Admin Console

3.1.14.1 Defining Vertical or Horizontal Answer Control Layout

Use this procedure to choose the layout of the answer controls in the script panel.

Prerequisites

Insert a panel onto the canvas.

Steps

1. In the tool palette, click the **Toggle Select Mode** tool.
2. On the canvas, double-click a panel.

   The Properties dialog box appears.
3. In the Panel tree, select **Properties**.
4. In the Answers pane, do one of the following:
   - If you want to vertically align all answer controls for the script panel, then select **Vertical Answer Layout**.

- If you want to horizontally align all answer controls for the script panel, then select **Horizontal Answer Layout**.

5. Click **Apply** to apply your work or click **OK** to save your work and exit the Properties dialog box.

See Also

Adding Validation to an Answer Definition

Defining Data Constraints for an Answer Control

Reordering Answer Options

Reordering Answer Controls

Deleting Answer Options from an Answer Control

Deleting Answer Controls from a Script Panel

Substituting a Java Bean for an Answer

Packaging Java Bean Code Into a JAR File

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.14.2 Adding Validation to an Answer Definition

Use this procedure to define a validation command to be applied at runtime against an answer provided into the runtime answer control GUI. When the end user provides an answer for this node in a panel, the answer provided will be evaluated based on the parameters dictated by the validation code. In this way, you can enforce the answer to be not null, or in a specified range of numbers, in a particular format, or whatever other conditions controlled by the validation command.

Prerequisites

- Write the code for the validation command.
- Insert a panel onto the canvas.
- Define an answer control for the script panel.

Steps

1. In the tool palette, click the **Toggle Select Mode** tool.
2. On the canvas, double-click a panel.

   The Properties dialog box appears.
3. In the Panel tree, select **Answers**.
4. In the Answers pane, select the answer and then click **Edit Data Dictionary**.

   The Data Dictionary dialog box appears.
5. In the General tab, click **Edit** on the Validation control

   The Command dialog box appears.
6. Define a command for the validation action. (See Defining Commands.)
7. Click **OK** to exit the Data Dictionary dialog box.
8. Click **OK** to exit the Data Dictionary dialog box.
9. Click **Apply** to apply your work or click **OK** to save your work and exit the Properties dialog box.

See Also

Defining Vertical or Horizontal Answer Control Layout

Defining Data Constraints for an Answer Control

Reordering Answer Options

Reordering Answer Controls

Deleting Answer Options from an Answer Control

Deleting Answer Controls from a Script Panel

Substituting a Java Bean for an Answer

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.14.3 Defining Data Constraints for an Answer Control

Use this procedure to define data constraints for an answer control.

Prerequisites

- Identify the data constraints applicable to this answer
- Insert a panel onto the canvas.
- Define an answer control for the script panel.
-

Steps

1. In the tool palette, click the **Toggle Select Mode** tool.
2. On the canvas, double-click a panel.

   The Properties dialog box appears.
3. In the Panel tree, select **Answers**.
4. In the Answers pane, select the answer and then click **Edit Data Dictionary**.

   The Data Dictionary dialog box appears.
5. Define the data constraints for the answer control.
6. Click **OK** to exit the Data Dictionary dialog box.
7. Click **Apply** to apply your work or click **OK** to save your work and exit the Properties dialog box.

See Also

Defining Vertical or Horizontal Answer Control Layout

Adding Validation to an Answer Definition

Reordering Answer Options

Reordering Answer Controls

Deleting Answer Options from an Answer Control

Deleting Answer Controls from a Script Panel

Substituting a Java Bean for an Answer

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.14.4 Reordering Answer Options

Use this procedure to order the options in an answer control that has a list of values (such as a radio button group or check box group).

Prerequisites

- Insert a panel onto the canvas.
- Define an answer control for the script panel.

Steps

1. In the tool palette, click the **Toggle Select Mode** tool.
2. On the canvas, double-click a panel.

   The Properties dialog box appears.
3. In the Panel tree, select **Answers**.
4. In the Answers pane, select the answer and then click **Edit Data Dictionary**.

   The Edit Data Dictionary dialog box appears.
5. In the Lookups tab, select a value.
6. If you want to move the option up, then click **Move Up**.
7. If you want to move the option down, then click **Move Down**.
8. Click **OK** to exit the Edit Data Dictionary dialog box.
9. Click **Apply** to apply your work or click **OK** to save your work and exit the Properties dialog box.

See Also

Defining Vertical or Horizontal Answer Control Layout

Adding Validation to an Answer Definition

Defining Data Constraints for an Answer Control

Reordering Answer Controls

Deleting Answer Options from an Answer Control

Deleting Answer Controls from a Script Panel

Substituting a Java Bean for an Answer

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.14.5 Reordering Answer Controls

Use this procedure to order the answer controls in a script panel.

Prerequisites

- Insert a panel onto the canvas.
- Define an answer control for the script panel.

Steps

1. In the tool palette, click the **Toggle Select Mode** tool.
2. On the canvas, double-click a panel.

   The Properties dialog box appears.
3. In the Panel tree, select **Answers**.
4. In the Answers pane, select an answer.
5. If you want to move the answer control up, then click **Move Up**.
6. If you want to move the answer control down, then click **Move Down**.
7. Click **Apply** to apply your work or click **OK** to save your work and exit the Properties dialog box.

See Also

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.14.6 Deleting Answer Options from an Answer Control

Use this procedure to delete an options from an answer control.

Prerequisites

- Insert a panel onto the canvas.
- Define an answer control for the script panel.

Steps

1. In the tool palette, click the **Toggle Select Mode** tool.
2. On the canvas, double-click a panel.

   The Properties dialog box appears.
3. In the Panel tree, select **Answers**.
4. In the Answers pane, select an answer and then click **Edit Data Dictionary**.

   The Edit Data Dictionary dialog box appears.
5. In the Lookups tab, select a value and then click **Remove**.
6. Click **OK** to exit the Edit Data Dictionary dialog box.
7. Click **Apply** to apply your work or click **OK** to save your work and exit the Properties dialog box.

See Also

Defining Vertical or Horizontal Answer Control Layout

Adding Validation to an Answer Definition

Defining Data Constraints for an Answer Control

Reordering Answer Options

Reordering Answer Controls

Deleting Answer Controls from a Script Panel

Substituting a Java Bean for an Answer

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.14.7 Deleting Answer Controls from a Script Panel

Use this procedure to delete an answer controls from a script panel.

Prerequisites

- Insert a panel onto the canvas.
- Define an answer control for the script panel.

Steps

1. In the tool palette, click the **Toggle Select Mode** tool.
2. On the canvas, double-click a panel.

   The Properties dialog box appears.
3. In the Panel tree, select **Answers**.
4. In the Answers pane, select an answer and then click **Remove**.
5. Click **Apply** to apply your work or click **OK** to save your work and exit the Properties dialog box.

See Also

Defining Vertical or Horizontal Answer Control Layout

Adding Validation to an Answer Definition

Defining Data Constraints for an Answer Control

Reordering Answer Options

Reordering Answer Controls

Deleting Answer Options from an Answer Control

Substituting a Java Bean for an Answer

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.14.8 Substituting a Java Bean for an Answer

Substituting a Java bean for an answer (or panel node) is no longer supported in Oracle Scripting as of release 11.5.6 and subsequent releases, due to the new WYSIWYG editing feature of the Script Author.

See Also

Defining a Button in a Script Panel

Defining a Text Field in a Script Panel

Defining a Multi Line Text Area in a Script Panel

Defining a Drop Down List Answer Control in a Script Panel

Defining a Radio Button Group Answer Control in a Script

Defining a Check Box Group Answer Control in a Script Panel

Defining Vertical or Horizontal Answer Control Layout

Defining Data Constraints for an Answer Control

Reordering Answer Options

Reordering Answer Controls

Deleting Answer Options from an Answer Control

Deleting Answer Controls from a Script Panel

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.15 Defining Actions

You can perform the following tasks:

Defining Script Pre- and Post-Actions

Defining Panel Pre- and Post-Actions

Defining Group Pre- and Post-Actions

Defining Block Pre- and Post-Actions

Defining Branch Actions

See Also

Defining Panel Answer Controls

Getting Started with the Script Author

Obtaining Script Requirements Before Creating a Script

Defining Global Script Attributes

Working with Script Files

Working with the Tool Palette

Viewing Objects on the Canvas

Working With Objects on the Canvas

Defining Panels

Defining Groups

Defining Blocks

Defining Branches

Controlling Script Flow

Defining Panel Answer Controls

Working With Answers

Defining Commands

Deploying the Script

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.15.1 Defining Script Pre- and Post-Actions

Use this procedure to define an action to take place before or after a script.

Prerequisites

Create or open a script.

Steps

1. Choose **File > Script Properties** or double-click on the canvas away from any objects.

   The Properties dialog box appears.

2. In the Script tree, expand **Actions** and then select **Pre Actions** or **Post Actions**.

3. In the Actions pane, click **Add**.

   The Command dialog box appears.

4. Define a command for the action. (See Defining Commands.)

5. Click **Apply** to apply your work or click **OK** to save your work and exit the Properties dialog box.

See Also

Defining Panel Pre- and Post-Actions

Defining Group Pre- and Post-Actions

Defining Block Pre- and Post-Actions

Defining Branch Actions

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.15.2 Defining Panel Pre- and Post-Actions

Use this procedure to define an action to take place before or after a panel.

Prerequisites

Insert a panel onto the canvas.

Steps

1. In the tool palette, click the **Toggle Select Mode** tool.
2. On the canvas, double-click a panel.

   The Properties dialog box appears.
3. In the Panel tree, expand **Actions** and then select **Pre Actions** or **Post Actions**.
4. In the Actions pane, click **Add**.

   The Command dialog box appears.
5. Define a command for the action. (See Defining Commands.)
6. Click **Apply** to apply your work or click **OK** to save your work and exit the Properties dialog box.

See Also

Defining Panels

Inserting a Panel

Defining Script Pre- and Post-Actions

Defining Group Pre- and Post-Actions

Defining Block Pre- and Post-Actions

Defining Branch Actions

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.15.3 Defining Group Pre- and Post-Actions

Use this procedure to define an action to take place before or after a group.

Prerequisites

Insert a group onto the canvas

Steps

1. In the tool palette, click the **Toggle Select Mode** tool.
2. On the canvas, double-click a group.

   The Properties dialog box appears.
3. In the Group tree, expand **Actions** and then select **Pre Actions** or **Post Actions**.
4. In the Actions pane, click **Add**.

   The Command dialog box appears.
5. Define a command for the action. (See Defining Commands.)
6. Click **Apply** to apply your work or click **OK** to save your work and exit the Properties dialog box.

See Also

Defining Groups

Defining Script Pre- and Post-Actions

Defining Panel Pre- and Post-Actions

Defining Block Pre- and Post-Actions

Defining Branch Actions

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

3.1.15.4 Defining Block Pre- and Post- Actions

Use this procedure to define an action to take place before or after a block.

Prerequisites

Insert a block onto the canvas.

Steps

1. In the tool palette, click the **Toggle Select Mode** tool.
2. On the canvas, double-click a block.

   The Properties dialog box appears.
3. In the Block tree, expand **Actions** and then select **Pre Actions** or **Post Actions**.
4. In the Actions pane, click **Add**.

   The Command dialog box appears.
5. Define a command for the action. (See Defining Commands.)
6. Click **Apply** to apply your work or click **OK** to save your work and exit the Properties dialog box.

See Also

Defining Blocks

Defining Script Pre- and Post-Actions

Defining Panel Pre- and Post-Actions

Defining Group Pre- and Post-Actions

Defining Branch Actions

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.15.5 Defining Branch Actions

Use this procedure to define an action to take place upon branching

Prerequisites

Insert branches onto the canvas.

Steps

1. In the tool palette, click the **Toggle Select Mode** tool.
2. On the canvas, double-click a branch.

   The Properties dialog box appears.
3. In the tree, select **Actions**.
4. In the Actions pane, click **Add**.

   The Command dialog box appears.
5. Define a command for the action. (See Defining Commands.)
6. Click **Apply** to apply your work or click **OK** to save your work and exit the Properties dialog box.

See Also

Defining Branches

Defining Script Pre- and Post-Actions

Defining Panel Pre- and Post-Actions

Defining Group Pre- and Post-Actions

Defining Block Pre- and Post-Actions

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

3.1.16 Defining Commands

You can perform the following tasks:

Invoking a Public Method in a Java Class

Invoking a PL/SQL Stored Procedure or Function in a Database

Invoking a PL/SQL Procedure or Function in a Forms Server

Setting a Constant

Retrieving a Panel Answer From the Scripting Blackboard

See Also

Defining Panel Answer Controls

Getting Started with the Script Author

Obtaining Script Requirements Before Creating a Script

Defining Global Script Attributes

Working with Script Files

Working with the Tool Palette

Viewing Objects on the Canvas

Working With Objects on the Canvas

Defining Panels

Defining Groups

Defining Blocks

Defining Branches

Controlling Script Flow

Defining Panel Answer Controls

Working With Answers

Defining Actions

Deploying the Script

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.16.1 Invoking a Public Method in a Java Class

Use this procedure to invoke a public method in a Java class and return the value returned by the invocation, if any. The Java command object takes zero or more parameters and returns zero or one value.

Prerequisites

Write the code that handles the data exchange.

Steps

1. Navigate to a Command dialog box.
2. In the Command Type area, select **Java Command**.
3. In the Command Info area, do the following:
   a. In the Name field, type the name of the command.
   b. In the Command field, type the command string as follows:

      ```
      <fully qualified class name>::<method name>
      ```
4. If you want to list parameters for the command object, then, in the Parameters area, do the following:
   a. Click **Add**.

      The Parameters dialog box appears.
   b. In the Name field, type the name of the parameter.
   c. In the Value field, type the literal string value for the parameter.
   d. If you want to add the value as the return value of an executed command, then select **Add Value as Command** and then click **Edit** in the Value field.

      Note: A command object used as a parameter to another command must return a string value.

      ---
   e. Click **OK** to exit the Parameters dialog box.
   f. If you want to add another parameter, then repeat steps a through e.
5. Click **OK** to exit the Command dialog box.
6. Save your work.

See Also

Invoking a PL/SQL Stored Procedure or Function in a Database

Invoking a PL/SQL Procedure or Function in a Forms Server

Setting a Constant

Retrieving a Panel Answer From the Scripting Blackboard

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.16.2 Invoking a PL/SQL Stored Procedure or Function in a Database

Use this procedure to invoke a PL/SQL stored procedure or function in an Oracle database and return the value returned by the invocation, if any. The PL/SQL command object takes zero or more parameters and returns zero or one value.

Prerequisites

Write the code that handles the data exchange.

Steps

1. Navigate to a Command dialog box.
2. In the Command Type area, select **PL/SQL Command**.
3. In the Command Info area, do the following:
   a. In the Name field, type the name of the command.
   b. In the Command field, type the name of a PL/SQL stored procedure or function in an Oracle database.
4. If you want to list parameters for the command object, then, in the Parameters area, do the following:
   a. Click **Add**.

      The Parameters dialog box appears.
   b. In the Name field, type the name of the parameter.
   c. In the Value field, type the literal string value for the parameter.
   d. If you want to add the value as the return value of an executed command, select **Add Value as Command** and then click **Edit** in the Value field.

Note: A command object used as a parameter to another command must return a string value.

e. In the In/Out list, select the parameter type.

f. Click **OK** to exit the Parameters dialog box.

g. If you want to add another parameter, then repeat steps a through f.

5. If you want to connect to the Oracle database using new connection parameters, then do the following:

   a. Clear the **Reuse an existing connection** box.

   b. Type the connection parameters. (See Database Connection Guidelines.)

6. If you want to reuse existing connection parameters to connect to the Oracle database schema, then do the following:

   a. Select the **Reuse an existing connection** box.

   b. From the Existing Connections list, select a connection.

   c. In the Password field, type the password for the selected database connection.

7. Click **OK** to exit the Command dialog box.

8. Save your work.

See Also

Invoking a Public Method in a Java Class

Invoking a PL/SQL Procedure or Function in a Forms Server

Setting a Constant

Retrieving a Panel Answer From the Scripting Blackboard

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.16.3 Invoking a PL/SQL Procedure or Function in a Forms Server

Use this procedure to invoke a PL/SQL procedure or function defined in an Oracle Forms PL/SQL package on the Oracle Forms server and return the value returned by the invocation, if any. The Forms command object takes zero or more parameters and returns zero or one value.

Prerequisites

Write the code that handles the data exchange.

Steps

1. Navigate to a Command dialog box.
2. In the Command Type area, select **Forms Command**.
3. In the Command Info area, do the following:
   a. In the Name field, type the name of the command.
   b. In the Command field, type the command string.
4. If you want to list the parameters for the command object, then, in the Parameters area, do the following:
   a. Click **Add**.

      The Parameters dialog box appears.
   b. In the Name field, type the name of the parameter.
   c. In the Value field, type the literal string value for the parameter.
   d. If you want to add the value as the return value of an executed command, then select **Add Value as Command** and then click **Edit** in the Value field.

      Note: A command object used as a parameters to another command must return a string value.

      ---

   e. In the In/Out list, select the parameters type.
   f. Click **OK** to exit the Parameters dialog box.
   g. If you want to add another parameter, then repeat steps a through f.
5. If you want to define a return value, then, in the Return Value area, do the following:

a. Click **Edit**.

   The Parameters dialog box appears.

b. In the Name field, type the name of the return value.

c. Click **OK** to exit the Parameters dialog box.

6. Click **OK** to exit the Command dialog box.
7. Save your work.

See Also

Invoking a Public Method in a Java Class

Invoking a PL/SQL Stored Procedure or Function in a Database

Setting a Constant

Retrieving a Panel Answer From the Scripting Blackboard

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.16.4 Setting a Constant

Use this procedure to return a constant value. The Constant command object takes no parameters and returns one value.

Prerequisites

None

Steps

1. Navigate to a Command dialog box.
2. In the Command Type area, select **Constant Command**.
3. In the Command Info area, type the name of the command in the Name field.
4. In the Return Value area, do the following:

   a. Click **Edit**.

The Parameters dialog box appears.

b. In the Value field, type the return value.

c. Click **OK** to exit the Parameters dialog box.

5. Click **OK** to exit the Command dialog box.

6. Save your work.

See Also

Invoking a Public Method in a Java Class

Invoking a PL/SQL Stored Procedure or Function in a Database

Invoking a PL/SQL Procedure or Function in a Forms Server

Retrieving a Panel Answer From the Scripting Blackboard

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.16.5 Retrieving a Panel Answer From the Scripting Blackboard

Use this procedure to return a value from the Oracle Scripting blackboard. The Blackboard command object takes no parameters and returns one value.

Prerequisites

None

Steps

1. Navigate to a Command dialog box.

2. In the Command Type area, select **Blackboard Command**.

3. In the Command Info area, do the following:

   a. In the Name field, type the name of the command.

   b. In the Command field, type the name of the panel answer to be returned.

Note: Panel answers are listed in the Properties dialog for a panel, on the Answers pane.

4. Click **OK** to exit the Command dialog box.
5. Save your work.

See Also

Invoking a Public Method in a Java Class

Invoking a PL/SQL Stored Procedure or Function in a Database

Invoking a PL/SQL Procedure or Function in a Forms Server

Setting a Constant

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.17 Deploying the Script

You can perform the following tasks:

Checking Script Syntax

Deploying a Script to the Database

Connecting to the Database

See Also

Getting Started with the Script Author

Obtaining Script Requirements Before Creating a Script

Defining Global Script Attributes

Working with Script Files

Working with the Tool Palette

Viewing Objects on the Canvas
Working With Objects on the Canvas
Defining Panels
Defining Groups
Defining Blocks
Defining Branches
Controlling Script Flow
Defining Panel Answer Controls
Working With Answers
Defining Actions
Defining Commands
Using the Script Author
Troubleshooting the Script Author
Using the Scripting Engine
Taking a Survey
Using the Survey Admin Console

3.1.17.1 Checking Script Syntax

Use this procedure to check the syntax of a script.

Prerequisites

Create or open a script.

Steps

1. Choose **Tools > Syntax Check** or click the **Check Syntax** icon in the toolbar.

   The **Check Syntax** icon is the fourth icon from the right of the toolbar, located immediately below the menus in the Script Author interface.

   When the syntax check is complete, a message appears in the status bar. If there are errors, then the Syntax tab is displayed.

2. In the Syntax tab, click a node to view the error message (if any).

See Also

Deploying a Script to the Database

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.17.2 Deploying a Script to the Database

Use this procedure to deploy the compiled script to the Oracle Scripting database schema.

Prerequisites

- You must have a syntactically correct script.
- You will need the database hostname, TNS port and SID.
- Open a script to be deployed in the Script Author.

Steps

1. Choose **Tools > Deploy Script** or click the **Deploy Script to DataBase** icon in the toolbar.

   The **Deploy Script to DataBase** icon is the third icon from the right of the toolbar, located immediately below the menus in the Script Author interface.

2. If there are errors, then the Syntax tab is displayed and the Debug pane appears. (See Checking the Script Syntax.) Otherwise, the Deploy Script To dialog box appears.

   Note: Always use a new connection.

3. Type the parameters for connecting to the Oracle Scripting schema.
4. Click **OK** to deploy the script.

See Also

Defining the Script Name

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.1.17.3 Connecting to the Database

When deploying a script, performing a query, insert, or update command, or performing a PL/SQL command, you will need to use the Script Author interface to connect to the database.

Enter database connection information for your environment based on the following steps and guidelines.

Steps

1. In the **Name** field, enter a name for this connection.

   This information is used by the Script Author only. Any characters may be used as part of this name.

2. In the **Host Name** field, enter the name of the database host (including domain).

   Example: `database.vision.com`

3. In the **Port Number** field, enter the TNS port of the database.

4. In the **SID** field, enter the database global name or SID.

5. In the **User Name** field, enter an appropriate database User Name.

   This user must have privileges to access the `IES_DEPLOYED_SCRIPTS` table of the applications database (for example, the `apps` user).

6. In the **Password** field, enter the password for the appropriate user name.

7. Once you have entered all appropriate connection information, click **OK**.

Follow these database connection guidelines.

Database Connection Guidelines

Database connections are made from Scripting using thin JDBC only. When you invoke a connect object, the following JDBC driver is invoked automatically,

specifying the Java class to be used as the driver for connecting to the Oracle database: `oracle.jdbc.driver.OracleDriver`

Connection Name. This value is used by Script Author only and can contain any characters.

Host Name. The `<hostname>` parameter specifies the name of the database server machine. Include the domain when specifying a host.

Port Number. The `<Port Number>` parameter specifies the port number for TNS connections. This value is determined by the configuration of the database.

SID. The `<SID>` parameter specifies the SID of the database or the instance name.

User Name Administrator Oracle8*i* or Oracle9*i* database login

Password Administrator password

See Also

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.2 Troubleshooting the Script Author

This topic includes the following topic groups:

Obtaining Help in the Script Author

Storing Strings in the Script Author

Known Issues Using COMMIT in Custom PL/SQL Procedures

See Also

Using the Script Author

Using the Scripting Engine

Taking a Survey

Using the Survey Admin Console

3.2.1 Obtaining Help in the Script Author

Script Author 11*i* includes online help. However, at the current time, this online help is not accessible from the Help menu of the Script Author. To view online help, perform the steps indicated below.

Prerequisites

- Online help has been updated as of release 11.5.6. You must access online help from a computer loaded with the Script Author tool.
- You must use a Web browser to access online help.

Steps

1. On your local computer or network, navigate to the directory in which the Script Author was installed. As an example, the path to access this directory on your computer may be similar to the following:

   ```
   C:\OraHome1\oracle\apps\ies\author
   ```

   Note: *The path to the ScriptAuthor directory may be different on your computer.* The Script Author may be located in your Oracle Home directory, although this is not a requirement. To locate your Oracle Home directory: (1) Right click on My Computer from your desktop; (2) Click the Environment tab; (3) View the directory path in the `Path` variable field.

2. Navigate to the `docs` subdirectory.
3. From your Web browser, open the Script Author Online Help table of contents, `toc.htm`, from the `docs` subdirectory.
4. In your Web browser, navigate to the appropriate topic as applicable.

See Also

Storing Strings in the Script Author

Known Issues Using COMMIT in Custom PL/SQL Procedures

3.2.2 Storing Strings in the Script Author

A file called `bundle.contents` is used to store strings in the Script Author. When the Script Author is launched, it checks to see if the file exists--if it does, then the

However, the SQL calls that were made within the autonomous transaction *will not be rolled back.*

2. In addition, Oracle Scripting provides the ability to script user from changing answers in earlier panels. The two API calls, `deactivateAllPrevious Panels()` and `deactivateAllPreviousPanels(Stringdeactivation Message)`, will prevent the agent from re-enabling any panel from the first panel to the current panel (when the API is called). In addition, specifying a `deactivationMessage` parameter will cause the Scripting Engine to display the `deactivationMessage` when the agent attempts to re-enable any panel that has been deactivated.

3.3 Using the Scripting Engine

Scripts created in the Script Author are deployed to the applications database. Thereafter, interaction center agents with the appropriate Oracle Applications responsibility can launch a script and run it in the Scripting Engine. Oracle Scripting 11*i* is integrated with three business applications in release 11.5.6: Oracle TeleService, Oracle TeleSales, and Oracle Collections. For script testing purposes, scripts can also be launched in stand-alone mode using the `Script User` or `Scripting Agent` responsibilities. For information on these responsibilities, see Responsibilities That Can Launch the Scripting Engine.

The Scripting Engine GUI is simple to use. An agent can be thoroughly trained in a very short time. The GUI contains the following features:

- The Panel Display Area
- The Progress Area
- The Script Information Area
- The Shortcut Button Bar
- The Disconnect Button
- The Toolbar

The GUI elements listed above are described in detail in the section Understanding the Scripting Engine. As a summary, when the Scripting agent launches a script, a separate Forms window opens with the Scripting Engine GUI. The agent navigates from panel to panel in the Panel Display Area. Script Author objects such as groups, blocks, Start and Termination nodes and the various branch types provide functionality behind the scenes—they do not display.

Each panel requires agent interaction (typically by clicking the answer controls and selecting the panel's **Continue** button). An agent can click on and visit a previously displayed panel by using the Toolbar controls or by clicking a panel label in the Progress Area. Optionally, a script may include information above the Panel Display Area in the Script Information Area, which has nine possible field values, each able to have its own label. These values may be static or dynamic, and may contain alphanumeric information or a timer. Below this (if present) and immediately above the Panel Display Area, a set of Shortcut buttons may optionally be programmed into the script. These will jump a user to a section of the script (based on a group's Shortcut property) or fire a Scripting API (for example, to start or stop a timer in the Script Information Area). If enabled, the Disconnect button will "jump" the user to whatever group contains the shortcut property "WrapUpShortcut."

See Also

Using the Script Author

Troubleshooting the Script Author

Taking a Survey

Using the Survey Admin Console

Understanding the Scripting Engine

Scripting Engine Users

Responsibilities That Can Launch the Scripting Engine

3.4 Taking a Survey

There are a variety of ways to obtain survey respondents. (These methods are discussed in the section Survey Respondent Entry Points.) For list-based survey campaigns, the typical method is to send an e-mail invitation with the Survey deployment-specific URL (typically as a hyperlink). List-based survey campaign deployment URLs also contain unique respondent IDs which can control the amount of times a respondent may take a survey, and enables tracking of respondents against all list members invited to participate.

Non-list-based survey campaigns include a URL (which, depending on the method of delivery, may be a hyperlink) that a user may choose to visit. Users that choose not only to visit the ULR but also to participate in the survey are survey respondents. There is no respondent ID for non-list-based respondents, although

iSupport respondents may be tracked using unique iSupport identification codes or cookies.

Regardless of the entry point, the respondent must use an Oracle Applications 11*i*-compliant Web browser to take the survey. When they navigate to the appropriate URL, they will see the header JSP resource at the top of each page, followed by any panel text and answer controls. Each HTML page that displays equates to a single panel in the Script Author. Just as in the Scripting Engine model, Script Author objects such as groups, blocks, Start and Termination nodes and the various branch types provide functionality behind the scenes—they do not display.

Each HTML page requires agent interaction (typically by clicking the answer controls and selecting the panel's **Continue** button). If a Java stack error results (typically due to environmental factors at the enterprise), the error JSP resource page will display. Upon completion of the last panel in the script, the final JSP resource page will display.

Upon viewing the final JSP survey resource, the survey has been completed and the information is immediately committed to the Scripting schema in the Oracle Applications database. Concurrent processing then makes summarized survey information from the Scripting Schema available to the **Analysis** tab of the Survey Admin console for subsequent reporting and analysis. This is transparent to the user. Upon reaching the final JSP survey resource page, the respondent may exit the browser, surf to any links contained on that HTML page, and so forth. Their information has been successfully captured. For list-based survey campaigns, their response will immediately be available to view from the **Response** tab of the Survey Admin console.

See Also

Survey Respondent Entry Points

What Is Oracle Scripting? > Survey Component

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Using the Survey Admin Console

3.5 Using the Survey Admin Console

Using the Survey Admin console, a survey administrator can define survey campaign information, create cycles and deployments, define JSP resources, deploy surveys, monitor individual responses to an active or formerly active deployment, and generate reports based on summarized respondent information.

The Survey Admin console is a JSP/HTML-based survey campaign administrative GUI. Users with the `Survey Administrator` Oracle Applications responsibility can access this console from the Oracle eBusiness Center HTML login.

Substantial step-by-step procedures for accomplishing particular administration tasks are discussed in the section Administering Oracle Scripting Survey Campaigns. This section discusses applications and theory regarding the use of the Survey Admin console.

Topics in this topic group include:

- Campaigns and the Survey Component of Oracle Scripting
- Survey Campaign, Cycle, Deployment Hierarchy
- Survey Media Channels
- Survey Respondent Entry Points

See Also

Using the Script Author

Troubleshooting the Script Author

Using the Scripting Engine

Taking a Survey

3.5.1 Campaigns and the Survey Component of Oracle Scripting

A **campaign** can be defined as a focused effort to achieve a particular goal from a targeted population over a specific period of time for a particular business purpose. A campaign is a concept often employed in interaction centers.

From a Survey component perspective, the goal of a campaign is to obtain measurable, actionable information. This information allows the organization to analyze its customer/prospect responses to key initiatives, existing level of service, and so forth. The vehicle for obtaining this information is a survey questionnaire

(created in the Script Author, as discussed in the next section). There is a one-to-one correspondence between a survey campaign and a specific script.

The Survey component employs an administrative console using Java Server Pages technology built on the JTF HTML-based technology stack. A survey administrator can log into the Survey Admin console (from the eBusiness Center login) to define and configure survey campaign information. Prior to defining the survey campaign:

- The survey campaign information must be defined by a campaign administrator and other relevant individuals within the interaction center or enterprise;
- The Survey component of Oracle Scripting (and related components) must be appropriately implemented;
- A customized **survey questionnaire** must be developed and deployed.

See Also

- Survey Campaign, Cycle, Deployment Hierarchy
- Survey Media Channels
- Survey Respondent Entry Points

3.5.2 Survey Campaign, Cycle, Deployment Hierarchy

The hierarchy for defining a survey campaign in the Survey Admin console is:

- Survey Campaign
- Cycle
- Deployment

Survey Campaign

The survey campaign, at the top level of the hierarchy, is where parameters affecting the entire survey campaign are administered, such as the specific script to be used as the survey questionnaire, the survey campaign name, a description of the survey, status (open and cancelled), and survey resources. A survey campaign contains one or more **cycles** (described below).

Cycle

Cycle properties include a cycle name, an anonymous flag, a minimum response percentage, and status (open, active, or cancelled). Each cycle contains one or more

deployments (described below). The ability to define multiple cycles in a survey campaign aids in reporting for comparative data analysis for surveys to be conducted over a span of time.

Deployment

A deployment is the lowest member of the survey hierarchy. It is also the most granular. Deployments belong to a particular cycle. Deployment properties include list (if any), media type, the parent survey campaign and cycle, status (incomplete, active, closed), owner (Survey administrator login), deploy date, response end date, and last update date. Essentially, deployments are the construct within the survey campaign that contain key business rules (the "who," the "when," and the "how long") for that portion of a survey campaign. A deployment must be explicitly made active before respondents can participate in a survey campaign.

List-Based Information

The **Create Deployment** page of the Survey Admin console also contains fields in which survey administrators identify information specific to list-based survey campaigns. If you fill out list information on this page, the deployment will be considered list-based; otherwise it will be considered a non-list-based deployment.

Caution: For current releases, if you designate a deployment as list-based, this property has a backwards-cascading effect, causing the entire survey campaign to be a list-based campaign.

Future releases are expected to allow both list-based and non-list-based deployments in a single survey campaign.

The single campaign-level property configured here is the Web Server Uniform Resource Locator (URL). This will auto-populate with the URL of the current Apache Web server session (used to log into the eBusiness Center to access the Survey Admin console). If this value is incorrect or fails to auto-populate, you must enter the appropriate Apache Web server, port, and OA_HTML bin. If using a secondary Apache Web server, you must define this secondary Web server here.

.

Note: When entering data into the **Web Server URL** field, include the name of the Web server, port, and reference to HTML bin (ending with a slash). For example:
`http://www.company.com:8888/OA_HTML/`

The remaining properties are specific to the deployment and include the list name, invitation and reminder template names, the maximum number of responses to the survey allowed per person, the minimum number of responses enforced prior to closing the deployment, and the subject line of the e-mail invitation. For deployments using reminders, the number of reminders and reminder interval (in days) is configured here. Otherwise, leave this information blank.

See Also

- Administering Oracle Scripting Survey Campaigns > Defining Deployments
- Campaigns and the Survey Component of Oracle Scripting
- Survey Media Channels
- Survey Respondent Entry Points

3.5.3 Survey Media Channels

In current releases of Oracle Scripting, there is a single supported media channel for taking surveys: the World-Wide Web (`Web`). In the future, other media channels are intended to be supported. For example, a media channel such as "Call Center" would support the business case in which call center agents would input survey responses provided from respondents over the telephone. In current releases this parameter is hard-coded. When other media channels are available, you will select the appropriate channel at the *deployment* level when administering a survey campaign.

See Also

- Campaigns and the Survey Component of Oracle Scripting
- Survey Campaign, Cycle, Deployment Hierarchy
- Survey Respondent Entry Points

3.5.4 Survey Respondent Entry Points

Surveys range in goals from measuring customer satisfaction to obtaining market research, providing benchmark data, measuring political or public opinion, and so on. Populations targeted for participation in a survey vary according to the business drivers and business purpose for obtaining the data.

Sometimes the survey **sample** (the population from which information is solicited) is clearly defined, either from an existing list or an identifiable transaction session, and other times it is more random. In current releases of the Survey component, there are four supportable entry points for survey respondents, falling into two categories: list-based and non-list-based.

3.5.4.1 List-Based Entry Points

Some survey campaigns require an already-identified target population. Business rules underlying these survey campaigns may require respondents to be customers or prospects who have already been served by or otherwise have had initial contact with the enterprise. Oracle Marketing Online is a tool to manage lists of customers or prospects which can be leveraged for survey campaigns.

The single list-based entry point for iSurvey respondents is a scenario in which Fulfillment and OMO are implemented at the enterprise. Lists created in OMO are employed using Fulfillment APIs to send **invitations** — e-mail messages inviting list members to take part in a survey. Those list members who choose to participate in the survey (**respondents**) click on a URL or hyperlinked invitation image in the e-mail message, and are directed to the appropriate survey questionnaire in a Web browser.

List-based entry points offer survey campaign administrators the ability, at the cycle level, to monitor and track survey responses down to an individual respondent if desired, or to designate the campaign as anonymous.

List-based survey campaigns can continue to leverage Fulfillment and OMO integration using **reminders**. With reminders, follow-up e-mail messages can be sent to list members reminding them of the deadline for participating in the survey.

If a particular cycle was designated as not anonymous (i.e., the Anonymous flag was set to No), reminders will be sent only to those list members who have not yet responded. If the cycle was designated as anonymous, all list members (respondents and the remaining pool of potential respondents) will receive reminder e-mail messages. At this time, reminders are the only aspect affected by the Anonymous flag.

3.5.4.2 Non-List-Based Entry Points

The three non-list-based entry points for iSurvey respondents include Oracle iSupport, "pop-up" surveys from an enterprise Web site, and "walk-through" Web browsing from non-targeted members of the general population.

Oracle iSupport

Oracle iSupport is a self-service web application that enables enterprises to provide a method for their customers to log service or maintenance requests. If a Survey hyperlink is added to the iSupport interface, those utilizing iSupport can provide immediate feedback via iSurvey regarding their iSupport customer experience. As an added benefit, iSurvey can capture and retain the iSupport user ID assigned when the customer logged in, and tie the responses to that iSupport user, enabling some level of tracking between the two applications.

Pop-Up Surveys

Some campaigns may be designed to solicit feedback from individuals visiting an enterprise's Web site by adding "pop-up" surveys—using JavaScript or other technology to pop a separate window inviting the Web site visitor to take a short survey. The visitor has the option to accept or decline the invitation. The choice to accept is the hyperlink to launch the survey, whereas the choice to decline closes the pop-up window and allows the visitor to continue viewing the Web site.

Walk-Through Surveys

The Survey component of Oracle Scripting provides the added ability to obtain survey data from a wide audience. Some campaigns are best served by soliciting feedback from a random sample of the general population. This would best be served by "walk-through" traffic, with a sample population of respondents gathered from banner advertisements on Web sites and by respondents directed to the iSurvey URL from print and broadcast media. For enterprises with substantial Web traffic, walk-through survey data can also be obtained by simply providing a survey link on the enterprise Web site.

As opposed to list-based respondents who are responding to an invitation, or iSupport customers, there is no standard tracking mechanism for respondents whose entry point is a Web site either from a pop-up survey or as a walk-in survey respondent. Nor are lists, invitations or reminders relevant, precluding the need for OMO and Fulfillment as well as reliance upon OES.

See Also

- Understanding the Survey Component > The Survey Admin Console

- Campaigns and the Survey Component of Oracle Scripting
- Survey Campaign, Cycle, Deployment Hierarchy
- Survey Media Channels

4

Administering Oracle Scripting Survey Campaigns

This section provides descriptions of the tasks required to administer Oracle Scripting survey campaigns successfully. Administration of survey campaigns is accomplished using the Survey Admin console, accessible from the eBusiness Center login. Administrative tasks for this product include:

- Navigating the Survey Admin Console Graphic User Interface, including:
  - Navigating Using Tabs and Sub-Tabs
  - Using the Quick Find Feature
  - Using the View By Dropdown Control
  - Setting General Preferences for Logged-In User in the Survey Admin Console
- Administering survey resources, including:
  - Creating survey resources
  - Defining survey resources
  - Uploading survey resources
  - Setting the default survey resource in the Survey Admin console
- Administering survey campaign details, including:
  - Creating survey campaigns
  - Defining cycles
  - Defining deployments
  - Deploying or suspending a survey campaign deployment

    - Modifying Survey Campaign Details
- Generating Lists for List-Based Survey Campaigns
(using Oracle Marketing Online functionality)
- Setting Up Invitations and Reminders for List-Based Survey Campaigns
(using One-to-One Fulfillment functionality), including:
    - Fulfillment Activities Required for List-Based Survey Campaigns
    - Recommended Order of Steps
    - Invitations
    - Reminders
    - Sample Invitation Master Document
    - Sample Query
    - Verifying Status of a Fulfillment Request
- Managing survey campaign responses, including:
    - Navigating the Response Tab
    - Viewing Individual Responses in the Survey Admin Console
    - Manually Sending Reminders to Individual List Members
    - Administering concurrent processing to move respondent data to Survey summary tables
- Reporting and Analyzing Survey Respondent Data

4.1 Navigating the Survey Admin Console Graphic User Interface

Upon logging into the Survey Admin Console graphic user interface (GUI), the **Home** tab results. At this time, the contents of the **Home** tab are blank, and there are no sub-tabs for this category. In the future, it is anticipated that Oracle's Declarative Component Framework technology will be used to publish personalized bins and reports in home pages unique to each enterprise's own environment. At this time, the only function accessible from the home page is the Quick Find Feature.

Topics in this group include:

- Navigating Using Tabs and Sub-Tabs
- Using the Quick Find Feature

- Using the View By Dropdown Control
- Setting General Preferences for Logged-In User in the Survey Admin Console

4.1.1 Navigating Using Tabs and Sub-Tabs

There are six tabs in the Survey Admin console graphic user interface (GUI). Each may have sub-tabs associated with it. In brief, the tabs provide access to the following functions:

Table 4–1 Tabs and Functions in the Survey Admin Console GUI

Tab	Administrative Function
Home	No function at this time.
Survey Campaign	Administering survey campaign details and Administering survey resources
List	Administer lists using OMO functionality.
Fulfillment	Administer invitations and reminders using Fulfillment functionality.
Response	View individual survey responses.
Analysis	Generate three survey reports and one Scripting Engine report.

The following table includes a detailed summary of the tabs and sub-tabs These include:

Table 4–2 Tabs and Sub-Tabs in the Survey Admin Console GUI

Tab	Sub-Tab	Administrative Function
Home	None	No function at this time.
Survey Campaign	Survey Campaign	List existing survey campaigns. Create survey campaigns Modify Survey Campaign Details. Provide access to survey campaign details.

Table 4–2 Tabs and Sub-Tabs in the Survey Admin Console GUI

Tab	Sub-Tab	Administrative Function
Survey Campaign	Cycle	List existing cycles. Define cycles. Modify cycle details. Provide access to cycle details.
Survey Campaign	Deployment	List existing deployments. Define deployments. Provide access to deployment details. Deploy or suspend survey campaign deployments.
Survey Campaign	Survey Resources	List existing survey resources. Define survey resources
List	Audience	Define list details and audience.
List	Segment	Administer market segment.
List	Import	Import lists.
List	Search	Search lists.
Response	Home	None.
Response	Survey Campaign	Lists existing survey campaigns from which you can drill down two levels to view responses
Response	Cycle	Lists existing cycles from which you can drill down one level to view responses. Allows you to refresh the list to pull down most recent data.
Response	Deployment	Lists existing deployments from which you can view individual responses for list- and non-list-based survey campaigns. Viewing Individual Responses in the Survey Admin Console Manually sending reminders to individual list members.

Table 4–2 Tabs and Sub-Tabs in the Survey Admin Console GUI

Tab	Sub-Tab	Administrative Function
Analysis	Response Summary	Generates report based on summary tables (surveys only)
Analysis	List Summary	Generates reports for list-based deployments (surveys only)
Analysis	Question Frequency	Generates reports indicating question frequency against targeted response percent (surveys only)
Analysis	Panel Footprint	Generates reports for panels accessed (for scripts run in the Scripting Engine only)

Clicking on **List** and **Fulfillment** tabs from the Survey Admin console invokes GUI functions directly from other Oracle applications— Oracle Marketing Online (OMO) for list management functionality, and Oracle One-to-One Fulfillment (Fulfillment) for creation, management, deployment and delivery of invitations and reminders. These two tabs are only required for list-based survey campaigns.

See Also

- Using the Quick Find Feature
- Using the View By Dropdown Control
- Setting General Preferences for Logged-In User in the Survey Admin Console

4.1.2 Using the Quick Find Feature

Regardless of your location on the Survey Admin console, entering valid values for the name of a survey campaign, cycle, or deployment (based on the choice selected in the dropdown) and clicking Go results in navigation directly to the details page of that survey campaign, cycle, or deployment.

Thus, for example, if you wanted to view deployment details of a deployment named `testDeploy`, you would follow the steps below.

Prerequisites

None

Steps

1. Select `Deployment` from the **Quick Find** dropdown.
2. In the text field, enter `testDeploy`.

> **Note:** You can also use the % wildcard character in your search. If multiple records match your search criteria, a hyperlinked list will result. Clicking one of the hyperlink entries will result in the detail page for that survey object (survey campaign detail, survey cycle detail or survey deployment detail).

The detail page for that survey object (the survey campaign detail, cycle detail or survey deployment detail page) will result.

See Also

- Navigating Using Tabs and Sub-Tabs
- Using the View By Dropdown Control
- Setting General Preferences for Logged-In User in the Survey Admin Console

4.1.3 Using the View By Dropdown Control

On each list page in the Survey Admin console GUI, there is a **View By** dropdown control. This is context-sensitive to the category being listed.

The View By control determines whether the items listing for the current category (survey campaigns, cycles, deployments, or resources) will display only those items created by your login, or *all* items in that category.

For example, on the **Survey Campaigns** page (**Survey Campaigns** tab > **Survey Campaigns** sub-tab), the default choice of the **View By** dropdown is **My Surveys**. If you select the other choice (**All Surveys**) from the dropdown, the list of survey campaigns that display after the screen refreshes includes all survey campaigns defined for your environment.

This topic describes how to toggle the **View By** dropdown selection on any list page of the Survey Admin console. To keep the steps generic, the value `[category]` will be used. Thus, for example, instead of reading `My Deployments`, the steps below will read `My [category]`.

Note: Some applications, including Fulfillment and Marketing Online, have a **View** dropdown control instead of a **View By** dropdown control. These have the same function.

Prerequisites

None

Steps

1. In any list page of the Survey Admin console, check the status of the **View By** dropdown control.

   If the current value reads "`My [category]`," the resulting list will include only items created by the username you used to log into the Survey Admin console.

   If the current value reads "`All [category]`," the resulting list will include all items of that category.

2. Click on the arrow to the right of the **View By** dropdown control.

   All choices in the list of values for this dropdown control will display.

3. Scroll up or down in the list of values displayed and select the choice *not previously selected* from the dropdown control.

   As you make your selection, the HTML page refreshes, displaying a fresh list with the new criteria you selected.

See Also

- Navigating Using Tabs and Sub-Tabs
- Using the Quick Find Feature
- Using the View By Dropdown Control
- Setting General Preferences for Logged-In User in the Survey Admin Console

4.1.4 Setting General Preferences for Logged-In User in the Survey Admin Console

The following steps describe how to set or change general preferences for the logged-in user in the Survey Admin console. The following preferences can be changed on this page:

Table 4–3 General Preferences for User-Level Survey Profile Options

Profile Category	Profile Option
Responsibility Management	Current Responsibility
	Default Responsibility
General Display	Display Style [style sheet]
	Language
	User Currency
	Date Format
Table Display	Number of Rows Per Page
	Number of Blank Rows Per Table

Prerequisites

The following assumptions apply:

- The Survey component of Oracle Scripting is fully implemented in your environment.
- Resources have already been defined in the Survey Admin console.
- Your User ID has `Survey Administrator` responsibility and you have access to the eBusiness Center login for your environment.

Steps

1. Using an Oracle Applications 11*i*-compliant Web browser, access the appropriate URL for your environment to log into JTF tech stack HTML-based applications.
2. From the eBusiness Center applications login screen, enter in the provided fields a **User ID** and **Password** for a user with the `Survey Administrator` responsibility, and click **Go**.

   The Survey Admin console results.
3. At the top of the Survey Admin console, click the **Profile** icon or hyperlink.
4. The **General Preferences** page will display. Two navigational sub-tabs appear on the left, the active **General Preferences** hyperlink and an **iSurvey** hyperlink.

5. If General Preferences are not already visible, click the **General Preferences** hyperlink.
6. The **General Preferences** page will display.

   The fields listed in the table above describe all general preferences accessible from this page.
7. Change general preference profiles as desired.

   These will apply only to the logged-in user with the `Survey Administrator` responsibility.
8. When you are satisfied that you have made all appropriate changes, click **Update** from the bottom of the **General Preferences** page.
9. Continue your work or log out, as applicable.

See Also

- Navigating Using Tabs and Sub-Tabs
- Using the Quick Find Feature
- Setting the Default Survey Resource in the Survey Admin Console

4.2 Administering Survey Resources

This topic group provides task-based procedures for administering survey resources. Survey resources are JSP-format files that display (in certain conditions) during execution of a survey in a Web browser. Three resources—a Header Page, Error Page, and Final Page—may display when a survey questionnaire is taken over the Web. From an administration perspective, after physically creating or modifying a file to be used as a survey resource, each must be defined in the Survey Admin console, and the corresponding physical file must be uploaded to `$OA_HTML`. Survey resources must be in **`.jsp`** file format, and can either be constructed of pure HTML or can be a Java Server Page.

Once defined, Survey resources can be used any number of times. Default resources can be set per user by setting Profile options in the Survey Admin console, if desired. When you define a survey resource from the **Create Resource** page, the values you assign are stored in the `IES_SVY_RESOURCES` table of the applications database. Once you create a survey campaign and identify resources to be associated with that survey campaign, these resources and the campaigns with which they are associated are populated in the `IES_SVY_SVYRESOURCES` table of the applications database.

Topics include:

- Creating Survey Resources
- Defining Survey Resources
- Uploading Survey Resources
- Setting the Default Survey Resource in the Survey Admin Console

4.2.1 Creating Survey Resources

Survey resources must be JSP files, or HTML files saved with a `.jsp` file extension. You must have the requisite knowledge and resources to create or modify HTML and JSP files to use these appropriately with the Survey component of Oracle Scripting.

No provisions are made to create HTML or JSP files within the context of the Oracle Scripting product. However, as described in Test Resources below, four test resource JSP files ship with Oracle Applications, seeded in the appropriate directory on the server (`$OA_HTML`). In addition to using these to test your implementation, you may use these resources as building blocks, modifying them as appropriate.

Note: JSP files *must* be tested on an existing Web server to view appropriately.

Sample Code for Test Header Resource

Following is the source code for the sample test header. ***This sample is provided for informational purposes only.*** Oracle Corporation is not responsible for the correct implementation of JSP in your environment, and does not provide support for customized survey resources. Consult appropriate HTML and JSP experts for more information.

```
<!-- $Header: iessvytestheader.jsp 115.1 2001/03/19
12:06:28 pkm ship       $ -->

<table align=center border=0>
     </tbody>
      <tr>
         <td> </td>
         <td class=pageTitletcolspan=4 noWrap>Test
Header Section</td>
         <td> </td>
```

```
        </tr>
      </tbody>
    </table>
```

See Also

- Defining Survey Resources
- Uploading Survey Resources
- Setting the Default Survey Resource in the Survey Admin Console

4.2.2 Defining Survey Resources

Survey resources must be defined in the Survey Admin console and must map to existing HTML or JSP documents residing on the server (as described in Uploading Survey Resources).

> **Note:** The JSP files are *not* required to be physically on the server *at the time of definition* in the Survey Admin console. However, these files *must* be in their appropriate location on the server *when a survey is executed* in HTML, or a `file not found` exception will result.

Test Resources

Four test resources, identified in the table below, ship with Oracle Applications beginning with IES Minipack G, and available with any Rapid Install from 11.5.5 and later. These resources are physically located in `$OA_HTML` on the applications server.

Table 4–4 List of Test Survey Resources Shipping with Oracle Applications

Header Page	`iessvytestheader.jsp`
Error Page	`iessvytesterror.jsp`
Final Page	`iessvytestthanku.jsp`
Hosted survey error page	`iessvymenubasedtesterror.jsp`

Use Test Survey Resources for Implementation Testing

Utilizing these test survey resources provides a method to verify that survey resources are appropriately displayed upon execution of the survey campaign in an HTML GUI. At the same time, you are provided with a layer of abstraction in regard to the need to ensure the product is performing as expected without needing to test the creation of tailored JSP or HTML pages. For this purpose, it is recommended that you first test an implementation of the Survey component of Oracle Scripting with the first three test survey resources listed above, and then create and test customized survey resources.

Steps

1. Using an Oracle Applications 11*i*-compliant Web browser, access the appropriate URL for your environment to log into JTF tech stack HTML-based applications.
2. From the eBusiness Center applications login screen, enter in the provided fields a **User ID** and **Password** for a user with the `Survey Administrator` responsibility, and click **Go**.

   The Survey Admin console results. Upon clicking **Go**, the **Home** tab of the Survey Admin console is active. The content displayed on this page (if any) is environmentally dependent.
3. From the set of tabs at the top of the page, click the **Survey Campaign** tab.

   The **Survey Campaigns** page will result. Sub-tabs will appear immediately below the **Survey Campaign** tab, labeled **Survey Campaign, Cycle, Deployment,** and **Survey Resources**.
4. Click the **Resources** sub-tab

   Upon clicking **Resources**, the **Resources** page results.

   Based on the default choice of the **View By** dropdown for your environment (either **My Resources** or **All Resources**), either all existing survey resources are displayed, or only the set of resources created by your current User ID (if any exist). You can change this view at any time by selecting the remaining choice from the **View By** dropdown.
5. Click **Create** from the **Resources** page to begin defining a survey resource. Three resources must be identified: Header Page, Error Page, and Final Page.
6. Begin defining information for the Header Page. In the **File Name** field, type the name of the file that you have uploaded. This must be a JSP file (e.g., `<filename>.jsp`).

For more information on this file, see Creating Survey Resources above.

7. In the **Logical Name** field, type a name to serve as a pointer to the file.

   This name will be displayed in the Survey Admin console. To avoid confusion, you may choose to name it the same as the JSP resource itself, or a name that follows your enterprise requirements or naming conventions.

8. Optionally, you may type a description in the **Description** text area.

9. Select a language from the **Language** dropdown.

   This field reads and displays available languages from the Oracle RDBMS `FND_LANGUAGES` table. Select the appropriate language for your environment, if it is not already displayed. For example, for English in the United States, select **American English.**

10. When satisfied with the values for the Header Page resource, click **Create**.

    The Resources page will result, including the resource you just defined.

11. Repeat procedural steps 5 through 10 above for the Final Page and Error Page survey resources, respectively.

About the Hosted Survey Error Page

Three of the four test resources are for list- or non-list-based survey campaigns. The fourth resource, `iessvymenubasedtesterror.jsp`, is a sample test error page for a hosted survey scenario such as Oracle iSupport.

To test a hosted scenario, you can use any JSP header page (or HTML page saved in JSP file format). Error and final pages must also be in JSP file format. The error page must adhere to the hosted template. There are two ways to handle the final page:

1. Define as the final survey resource page the JSP page which displays the list of URLs to take the survey. In this way, when the survey is completed, the respondent will be returned to the starting page.

2. Define another JSP page as the final page survey resource, but include in that page a JSP forward command to point the user back to the page that displays the list of URLs to take the survey.

For more information, refer to the Oracle iSupport discussion in the section Non-List-Based Entry Points of the Using Oracle Scripting section of this document.

Next Steps

You now have all appropriate survey resources defined in the Survey Admin console for your survey campaign. Return to the steps to administer your survey campaign. See also Setting the Default Survey Resource in the Survey Admin Console to specify survey resources for a specific Survey Admin user.

See Also

- Creating Survey Resources
- Uploading Survey Resources
- Setting the Default Survey Resource in the Survey Admin Console

4.2.3 Uploading Survey Resources

Once you have JSP files created to use as survey resources, you must upload them to make them accessible to the Web server.

> **Caution:** These files *must* be physically stored in the `$OA_HTML` directory on the applications server. Test resources are already seeded in this directory.

If you employ graphics in custom JSP resources, the appropriate physical image files (typically in `.gif` or `.jpg` format) must be uploaded and stored to a server directory anywhere on the Web server. Since images supporting Oracle Applications are stored in the `$OA_MEDIA` directory on the server, this location is suggested. Correspondingly, the custom code for the resource must reference the image in its path, for example: `<img src="/OA_MEDIA/<image_name>.gif" ...>`. Consult with your systems administrator to determine if there are different requirements for your environment.

Prerequisites

To upload physical survey resource files, you need access to the `$OA_HTML` directory on the applications server and may need access to `$OA_MEDIA`. This level of access is typically granted to the system administrator.

Steps

1. On your local computer (or over a network), locate the `<filename>.JSP` files created as resources.

2. Connect to the applications server through an appropriate method such as TELNET.

   If you do not have permissions to access the appropriate environment, consult your system administrator.

3. Navigate to the `$OA_HTML` directory.

   `$OA_HTML` is a variable to point to a physical directory. JSP resources for the Survey component of Oracle Scripting must physically exist in this directory path in order to execute a survey in a Web browser.

4. Using an appropriate protocol such as FTP, copy or `put` the files to the `$OA_HTML` directory.

   You may choose to specify ASCII format for file transfer.

5. If your resource includes reference to graphics, navigate to the appropriate directory (for example, `$OA_MEDIA`).

   `$OA_MEDIA` is a variable to point to a physical directory in which media items for Oracle Applications are stored. The physical directory path may differ based on your specific environment.

6. Using an appropriate protocol such as FTP, copy or `put` the graphic files to the appropriate directory (for example, `$OA_MEDIA`).

7. The survey resources (and any supporting image files) will now be available to Survey administrators through the Survey Admin console.

Next Steps

You now have all appropriate survey resources created and defined in the Survey Admin console for your survey campaign, and the physical resources uploaded appropriately. Return to the steps to administer your survey campaign.

See Also

- Creating Survey Resources
- Defining Survey Resources
- Setting the Default Survey Resource in the Survey Admin Console

4.2.4 Setting the Default Survey Resource in the Survey Admin Console

The following steps describe how to set or change the default survey resource in the Survey Admin console.

Prerequisites

The following assumptions apply:

- The Survey component of Oracle Scripting is fully implemented in your environment.
- Resources have already been defined in the Survey Admin console.
- Your User ID has `Survey Administrator` responsibility and you have access to the eBusiness Center login for your environment.

Steps

1. Using an Oracle Applications 11*i*-compliant Web browser, access the appropriate URL for your environment to log into JTF tech stack HTML-based applications.
2. From the eBusiness Center applications login screen, enter in the provided fields a **User ID** and **Password** for a user with the `Survey Administrator` responsibility, and click **Go**.

   The Survey Admin console results.
3. At the top of the Survey Admin console, click the **Profile** icon or hyperlink.
4. The **General Preferences** page will display. Two navigational sub-tabs appear on the left, the active **General Preferences** hyperlink and an **iSurvey** hyperlink.
5. Click the **iSurvey** hyperlink.
6. The **iSurvey User Profile Option** page will display.

   The User ID for your current login appears, hard-coded, as the first item in the Survey Customization HTML table. The remaining customizable items for this user are the survey resources, with a database selection field to the right of each.
7. Enter your search criteria for the header survey resource in the **Header Page** field and click **Go**.

   Your criteria should match the logical name of HTML or JSP survey resources already defined in the Resources page of the **Survey Campaign** tab in the Survey Admin console. The physical HTML or JSP files should also be uploaded to the `$OA_HTML` directory.

   You may further refine your search in the Find Resources page by using the wildcard `%` sign.
8. Click the hyperlink for the appropriate resource.

The **iSurvey User Profile Option** page will display, with the resource populating the database selection field.

9. Repeat the selection of the appropriate survey resources for this user by repeating steps 7 and 8 above, once each for the Error Page and Final Page survey resources.
10. When you are satisfied that you have made all appropriate changes, click **Update** from the bottom of the **iSurvey User Profile Option** page.
11. Continue your work or log out, as applicable.

See Also

- Creating Survey Resources
- Defining Survey Resources
- Uploading Survey Resources

4.2.5 Non-List-Based URL Syntax and Example

The syntax for a non-list-based survey campaign deployment URL is:

```
http://enterprise.tld:<Web server port/OA_HTML/
iessvymain.jsp?dID=<deployment ID>
```

A sample URL for a non-list-based deployment might appear as follows:

```
http://vision.com:8888/OA_HTML/
iessvymain.jsp?dID=1111
```

dID: The Deployment ID is specific to a particular deployment. A survey campaign must have one deployment and may have more, as desired. The dID is stored in the `SURVEY_DEPLOYMENT_ID` field within the table `IES_SVY_DEPLYMENTS_ALL`. For information on how to determine the dID, see Obtaining the Deployment ID from the Survey Admin Console in this document.

See Also

- Deploying a Survey Campaign Deployment
- Understanding Status Conditions
- Status Conditions for Survey Campaigns, Cycles, and Deployments
- Deploying or Suspending a Survey Campaign Deployment
- Modifying Survey Campaign Details

4.3 Administering Survey Campaign Details

This topic group provides task-based procedures for administering survey campaign details. Perform these steps in the Survey Admin console to create a survey campaign and define its children components (the cycle and the deployment). After creating a survey campaign, cycle and deployment, the deployment must be deployed (made active) in order to execute the campaign.

Topics include:

- Creating survey campaigns
- Defining cycles
- Defining deployments
- Deploying or suspending a survey campaign deployment

4.3.1 Creating Survey Campaigns

Survey campaigns are created in the Survey Admin console by a survey administrator based on predefined survey campaign requirements.

> **Note:** Once saved, most properties of a survey campaign cannot be changed. For this reason, proper planning (including advance knowledge of all values to be entered into the GUI when performing campaign administration steps) is critical.

Prerequisites

In order to create a survey campaign in the Survey Admin console:

- An appropriate script serving as the survey questionnaire must be deployed to the applications database from the Script Author.
- Survey Resources must have been defined. While the corresponding physical JSP files for defined survey resources *need not exist on the server in order to administer* the survey campaign, they *must* be uploaded in order to *execute* the survey questionnaire in HTML.

Steps

1. Using an Oracle Applications 11*i*-compliant Web browser, access the appropriate URL for your environment to log into JTF tech stack HTML-based applications.

For the purposes of this document we will refer to this URL as the HTML Login URL and this login screen as the Oracle eBusiness Center login screen. This is also sometimes known as the Apache or JTF login screen.

2. From the eBusiness Center applications login screen, enter in the provided fields a **User ID** and **Password** for a user with the `Survey Administrator` responsibility, and click **Go**.

   The **User ID** in JTF applications is the same as a **User Name** in Oracle forms-based applications.

   The Survey Admin console results. Upon clicking **Go**, the **Home** tab of the Survey Admin console is active. The content displayed on this page (if any) is environmentally dependent.

3. From the set of tabs at the top of the page, click the **Survey Campaign** tab.

   The **Survey Campaigns** page results, with a list of survey campaigns (if any) that match the **View By** dropdown criteria. Sub-tabs will appear immediately below the **Survey Campaign** tab, labeled **Survey Campaign, Cycle, Deployment,** and **Survey Resources**.

   > **Note:** If *default survey resources* have already been established for this environment, and are to be used for this survey campaign, proceed with this process. If default survey resources have *not* been established, *or* if you need to use survey resources *other than the established defaults*, refer to Administering Survey Resources to create, define, and upload resources, and then return to this procedure.

   Based on the default choice of the **View By** dropdown for your environment (either **My Survey Campaigns** or **All Survey Campaigns**), either all existing survey campaigns (if any) are displayed, or only the set of survey campaigns created by your current User ID (if any). You can change this view at any time by selecting the remaining choice from the **View By** dropdown.

4. Click **Create** from the **Survey Campaigns** page to begin defining a survey campaign.

   The **Create a Survey** page results.

5. From the **Select a Script** field, enter specific search criteria in order to select the single Script Author script that will serve as the survey questionnaire for this survey campaign.

This criteria must correspond with the global **Script Nam**e of the survey questionnaire script, from a Script Author perspective. The script you select must already be deployed to the applications database.

> **Tip:** You can enter partial criteria, including the database wildcard character, "%". This will result in a list of scripts that meet the specified criteria. Alternatively, you can enter only the database wildcard character, "%," to view a list of all scripts.

6. When you are satisfied with the search criteria, click **Go**.

   The script or scripts in the database which match the specified search criteria will display in a list, each serving as a hyperlink. Optionally, you may refine your search further to narrow the returned script name values in a smaller list, as required.

7. Click the appropriate hyperlinked script name.

   The **Create Survey** page will again display, with the name of the selected script populating the **Select a Script** field. This field will also be referred to as the **Script Name** field.

8. In the **Survey Campaign Name** field, enter the name of the Survey campaign. This name should be specified by the campaign administrator in advance, and should follow your enterprise requirements or project naming conventions. This name will be displayed in the Survey Admin console to refer to this survey campaign.

   > **Caution:** When administering survey campaign information, most information *cannot be modified once it is defined*. If incorrect information is entered, a new survey campaign must be created to obtain the expected results.

9. Optionally, you may type a description in the **Description** text area.

10. If using the default survey resources, select the Default From Profile Option checkbox. If *not* using default survey resources:

    c. Click in the **Header Page** field.

    d. Enter the name of this survey resource type and click **Go**.

Tip: You can enter partial criteria, including the database wildcard character, "%". This will result in a list of survey resources that meet the specified criteria. Alternatively, you can enter only the database wildcard character, "%," to view a list of all survey resources. Refine your search in the **Select a Resource** field, if desired.

The **Resources** page will result, with a list of survey resources matching the specified search criteria, each serving as a hyperlink.

e. Click the appropriate hyperlinked survey resource name.

The **Create Survey** page will again display, with the name of the selected survey resource populating the **Header Page** field.

f. Repeat steps **10b** and **10c** for the Final Page and Error Page survey resources.

11. Note that the status of this survey campaign is **Open**.

The status will remain open until a child cycle is defined. At that point, you can change the status to **Cancelled** or leave the status open. A cycle cannot be set to **Active** status until a child deployment is created and set to **Active** (meaning the deployment is "deployed").

Next Steps

Before executing this survey campaign, you must define at least one cycle as a child object of the survey campaign, at least one deployment as a child object of the cycle, and make it active by deploying it, to execute this survey campaign.

See Also

- Defining cycles
- Defining deployments
- Deploying or suspending a survey campaign deployment

4.3.2 Defining Cycles

A cycle is a child object of the survey campaign. Each survey campaign must have at least one cycle defined (and may have many). Cycles include a cycle name, an anonymous flag, a minimum response percent field, and a status. The anonymous

flag is used for list-based survey campaigns to determine whether to send invitations or reminders to *all* list members (anonymous = yes) or just to send invitations or reminders to list members who have *not yet responded* (anonymous = no). When defining a cycle, you may also specify a minimum response percentage. This value is used in reporting, and includes all deployments defined under the cycle.

Cycles are defined from the **Survey Campaign** tab of the Survey Admin console for any open or active survey campaign and may be defined from two different points in the GUI. You may define a cycle from the **Create Survey Campaign** page immediately after defining a survey campaign. Alternatively, you may define a cycle for any open or active survey campaign from the **Cycle** sub-tab (by clicking **Create**). As with all information entered into the Survey Admin console for administration purposes, cycle data should be provided to the survey administrator from predefined survey campaign requirements.

Survey campaigns are created in the Survey Admin console by a survey administrator (login with the `Survey Administrator` responsibility).

Prerequisites

A survey campaign must be created before a cycle is defined. The following steps describe how to create a cycle from anywhere in the Survey Admin console. *You can also define cycle information on the* **Create Survey Campaign** *page.* If continuing from the procedure creating survey campaigns:

- Begin with step 8 below.
- You will not need to associate a parent survey campaign.
- The **Cycle Name** field will be labeled **Cycle**.

Steps

1. Using an Oracle Applications 11*i*-compliant Web browser, access the appropriate URL for your environment to log into JTF tech stack HTML-based applications.
2. From the eBusiness Center applications login screen, enter in the provided fields a **User ID** and **Password** for a user with the `Survey Administrator` responsibility, and click **Go**.

   The Survey Admin console results.
3. From the set of tabs at the top of the page, click the **Survey Campaign** tab.

The **Survey Campaigns** page will result. Sub-tabs will appear immediately below the **Survey Campaign** tab, labeled **Survey Campaign, Cycle, Deployment,** and **Survey Resources**.

4. Click the **Cycle** sub-tab.

   The **Cycles** page results, with a list of cycles (if any) that match the \ dropdown criteria.

   Based on the default choice of the **View By** dropdown for your environment (either **My Cycles** or **All Cycles**), either all existing cycles (if any) are displayed, or only the set of cycles created by your current User ID (if any). You can change this view at any time by selecting the remaining choice from the **View By** dropdown.

5. Click **Create** from the **Cycles** page to begin defining a cycle as a child of a survey campaign.

   The **Create Cycle** page results.

6. From the **Select a Survey Campaign** field, enter specific search criteria to identify the parent survey campaign and click **Go**.

   A list of survey campaigns matching the criteria you entered appears, each as a hyperlink.

7. Click the desired survey campaign.

   The **Select a Survey Campaign** field will populate with your selection.

8. In the **Cycle Name** field, enter the desired cycle name. Follow any requirements provided to you by the campaign administrator, and follow any existing naming conventions for your enterprise or project.

9. In the **Anonymous** dropdown, select `Yes` or `No` based on survey campaign requirements.

10. Note that the status remains `Open`.

11. In the **Minimum Response Percent** field, enter a target response percentage per survey campaign requirements. This must be an integer between 1 and 100.

12. Click **Create** to save this cycle definition.

Next Steps

You must define at least one deployment as a child object of a cycle, and make it active by deploying it, to execute this survey campaign.

See Also

- Defining deployments
- Deploying or suspending a survey campaign deployment

4.3.3 Defining Deployments

Deployments are children objects to a survey cycle. You can have multiple deployments per cycle, just as you can have multiple cycles per survey campaign. You must have at least one deployment in order to execute a survey campaign. A deployment associates a media type with the survey (at this time, "`Web`" is the only media type), defines the deployment start and response end date, identifies the appropriate Apache Web server upon which the survey will execute, and for list-based survey campaign deployments identifies invitation and reminder information. By setting deployment parameters you can also control how many times a list-based respondent may take the survey and identify a target percentage of responses to guage progress against goals when generating reports. After defining a deployment, its status is `Open` until you deploy it, making it `Active`.

Prerequisites

A survey campaign and cycle must be created before defining a deployment. The following steps describe how to define a deployment from anywhere in the Survey Admin console. *You can also access the Create Deployment page by clicking the* **Create** *button from the* **Cycle Details** *page*. If doing this, skip to step 6 below.

Steps

1. Using an Oracle Applications 11*i*-compliant Web browser, access the appropriate URL for your environment to log into JTF tech stack HTML-based applications.
2. From the eBusiness Center applications login screen, enter in the provided fields a **User ID** and **Password** for a user with the `Survey Administrator` responsibility, and click **Go**.

   The Survey Admin console results.
3. From the set of tabs at the top of the page, click the **Survey Campaign** tab.

   The **Survey Campaigns** page will result. Sub-tabs will appear immediately below the **Survey Campaign** tab, labeled **Survey Campaign, Cycle, Deployment,** and **Survey Resources**.
4. Click the **Deployment** sub-tab.

The **Deployments** page results, with a list of deployments (if any) that match the **View By** dropdown criteria.

Based on the default choice of the **View By** dropdown for your environment (either **My Deployments** or **All Deployments**), either all existing deployments (if any) are displayed, or only the set of deployments created by your current User ID (if any). You can change this view at any time by selecting the remaining choice from the **View By** dropdown.

5. Click **Create** from the **Deployments** page to begin defining a deployment.

   The **Create Deployments** page will result. You must associate your new deployment definition with a parent survey campaign and cycle.

6. In the **Survey Campaign** field, enter specific search criteria in order to select the appropriate parent survey campaign, and click **Go**.

   Tip: You can enter the exact survey campaign name, or you can enter partial criteria, including the database wildcard character, "%". This will result in a list of survey campaigns that meet the specified criteria. Alternatively, you can enter only the database wildcard character, "%," to view a list of *all* survey campaigns.

   The survey campaigns which match the specified search criteria will display in a list, each serving as a hyperlink. Optionally, you may refine your search further to narrow the returned survey campaign name values in a smaller list, as required.

7. Click the appropriate hyperlinked survey campaign name.

   The **Create Deployment** page will again display, with the name of the selected survey campaign populating the **Survey Campaign** field.

8. In the **Cycle** field, enter specific search criteria in order to select the appropriate parent cycle, and click **Go**.

9. Optionally, you can refine your search for this field in the same manner as the previous. Click the appropriate hyperlinked cycle name.

   The **Create Deployment** page will again display, with the name of the selected cycle populating the **Cycle** field.

10. In the **Deployment Name** field, enter the name of this deployment you are defining.

This name should be specified by the campaign administrator in advance, and should follow your enterprise requirements or project naming conventions.

11. Note that the **Media** and **Status** fields are hard-coded.

    For the Survey component of Oracle Scripting release 11.5.6, the only valid media type is `Web`.

    The Status field will later offer different status choices, as appropriate.

12. Click the calendar control widget in the **Deploy Date** field. Navigate to the appropriate date to start your deployment.

    This date should be specified by the campaign administrator in advance, and provided to you as part of the prerequisite information discussed in the Planning section of this document.

13. Optionally, in the **Deployment Time** field (located to the right of the **Deployment Date** field), enter the time for the deployment to begin, using the specified format.

    Tip: If you do not specify a time in the **Deployment Time** field, `12:00 AM` will be established by default.

14. Click the calendar control widget in the **Response End Date** field. Navigate to the appropriate date to end your deployment.

    This date should be specified by the campaign administrator in advance.

15. Optionally, in the **Response End Time** field (located to the right of the **Response End Date** field), enter the time for the deployment to end, using the specified format.

    If you do not specify a time, `12:00 AM` will be established by default.

16. In the **Web Server URL** field, the URL of the current Apache Web server session (used to log into the eBusiness Center to access the Survey Admin console) should populate by default. If it does not, or if you need to point this to a second Apache Web server (based on requirements established by the system administrator, Web administrator, and campaign administrator), enter the name of the Web server, port, and reference to HTML bin, ending with a slash.

    Example: **`http://vision.com:8888/OA_HTML/`**

17. You have now completed all fields required for *non-list-based* survey campaign deployments. For non-list-based deployments, skip to step **29**. If defining a

list-based campaign, continue to fill out list and reminder information as applicable, beginning with the next procedural step.

Note: The next set of steps is only for list-based survey campaigns. For non-list-based survey campaigns, skip to step **29**.

---

18. In the **List Name** field, enter specific search criteria in order to select the appropriate list, and click **Go**.

    The list name should be specified by the campaign administrator in advance, and should follow your enterprise requirements or project naming conventions. Lists must already exist in order to be selected.

    All lists which match the specified search criteria will display in a list in the **List Name** page, each serving as a hyperlink. Optionally, you may refine your search further to return a smaller found set, as required.

19. Click the appropriate hyperlinked list name.

    The **Create Deployment** page will again display, with the name of the selected list populating the **List Name** field.

20. In the **Invitation Template Name** field, enter specific search criteria in order to select the appropriate invitation template, and click **Go**.

    The invitation template associates a specific invitation (Fulfillment Master Document) with any relevant query to populate merge fields in the resulting populated e-mail invitation message.

    All invitation templates which match the specified search criteria will display in a list in the **Template Name** page, each serving as a hyperlink. Optionally, you may further refine your search to return a smaller found set, as required.

21. Click the appropriate hyperlinked invitation template name.

    The **Create Deployment** page will again display, with the name of the selected template populating the **Invitation Template Name** field.

22. Optionally, if your survey campaign requires reminders, enter specific search criteria in order to select the appropriate reminder template, and click **Go**.

    All reminder templates which match the specified search criteria will display in a list in the **Template Name** page, each serving as a hyperlink. Optionally, you may further refine your search to return a smaller found set, as required.

23. Click the appropriate hyperlinked reminder template name.

The **Create Deployment** page will again display, with the name of the selected template populating the **Reminder Template Name** field.

24. If the survey requirements provided to you by the campaign administrator require, change the default value of 1 in the **Maximum Responses Per Person** field.

    This removes the requirement to limit the amount of times a recipient ID can be used to successfully invoke the deployment URL. The number of times specified in this field determines how many times the unique combination of dID and rID can participate online in a survey questionnaire.

25. If the survey requirements provided to you by the campaign administrator

- The **Status** dropdown field is enabled. The value `Open` is selected, and you can view and select the other value (`Cancelled`).
- **Deploy Time** and **Response End Time** unlabeled fields have populated with a value of `12:00 AM`.
- **Maximum Responses Per Person** field has automatically populated with a value of `1`. Note that for non-list-based campaigns, this is not enforceable, as there is no unique identification method possible. List-based survey campaigns enforce this rule by checking both the **Deployment ID** (dID) and the **Respondent ID** (rID) values.
- At the bottom of the **Deployment Detail** page, there are now three buttons: **Update**, **Restore**, and **Deploy**.

30. To make this deployment active, click **Deploy** at the bottom of the **Deployment Detail** page.

    Tip: If you revise any values, click the **Update** button. **Restore** clears all changes in all fields.

    The **Deployment Detail** page will redraw. The Status field now displays a value of **Active**.

See Also

- Deploying or suspending a survey campaign deployment
- Modifying Survey Campaign Details

4.3.4 Deploying or Suspending a Survey Campaign Deployment

Topics include:

- Deploying a Survey Campaign Deployment
- Suspending a Survey Campaign Deployment
- Understanding Status Conditions
- Status Conditions for Survey Campaigns, Cycles, and Deployments

4.3.4.1 Deploying a Survey Campaign Deployment

Prerequisites

In order to deploy or make active a deployment, the deployment must be defined, a cycle as its parent object must be defined, and a survey campaign must have been created.

Steps

1. Using an Oracle Applications 11*i*-compliant Web browser, access the appropriate URL for your environment to log into JTF tech stack HTML-based applications.
2. From the eBusiness Center applications login screen, enter in the provided fields a **User ID** and **Password** for a user with the `Survey Administrator` responsibility, and click **Go**.

   The Survey Admin console results.
3. From the set of tabs at the top of the page, click the **Survey Campaign** tab.

   The **Survey Campaigns** page will result. Sub-tabs will appear immediately below the **Survey Campaign** tab, labeled **Survey Campaign, Cycle, Deployment,** and **Survey Resources**.
4. Click the **Deployment** sub-tab

   Upon clicking **Deployment**, the **Deployments** page results.

   Based on the default choice of the **View By** dropdown for your environment (either **My Deployments** or **All Deployments**), either all existing deployments are displayed, or only the set of deployments created by your current User ID (if any exist). You can change this view at any time by selecting the remaining choice from the **View By** dropdown.
5. Click the hyperlink for the deployment you wish to deploy.

   The **Deployment Details** page will result. The value displayed in the **Status** dropdown of a deployment which has already been defined but not yet made active is `Open`.
6. Click **Deploy** at the bottom of the **Deployment Details** page.

   The **Deployment Details** page will redraw, updating the value in the **Status** dropdown to `Active`. Your deployment is now active or deployed.

7. From the displayed in the **Status** dropdown, select **Create** from the **Resources** page to begin defining a survey resource. Three resources must be identified: Header Page, Error Page, and Final Page.

See Also

- Suspending a Survey Campaign Deployment
- Understanding Status Conditions
- Status Conditions for Survey Campaigns, Cycles, and Deployments
- How to Deploy or Suspend a Survey Campaign

4.3.4.2 Suspending a Survey Campaign Deployment

Prior to making a survey campaign active, you can change the status at the survey campaign and cycle level. Once you make a deployment active (thereby cascading upwards and making its parent objects, the survey campaign and cycle, active), you can only change status at the deployment level.

Once a deployment is active, *and* its Deploy Date is in the past, *and* its Response End Date is in the future, the Survey URL for that deployment is available. For list-based campaigns, this involves using Fulfillment to send out invitations (and reminders, if desired).

Once active, you can change deployment status from the **Deployments** page by selecting `Incomplete` (suspending activity after the campaign has been made active) or `Closed` (indicating that the deployment has run its course based on the parameters established for the survey campaign). This makes the Survey URL unavailable. An individual attempting to respond to a survey deployment with a status of `Incomplete` or `Closed` will see an error message display in the Web browser and will not be able to participate in the survey.

Caution: You cannot change the status of a deployment once you close it or make it incomplete; you must define a new deployment belonging to that survey campaign and cycle.

Prerequisites

- A survey campaign, cycle, and deployment must have been created.
- The deployment must have been deployed or set to `Active` status.

Steps

1. Using an Oracle Applications 11*i*-compliant Web browser, access the appropriate URL for your environment to log into JTF tech stack HTML-based applications.
2. From the eBusiness Center applications login screen, enter in the provided fields a **User ID** and **Password** for a user with the `Survey Administrator` responsibility, and click **Go**.

   The Survey Admin console results.
3. From the set of tabs at the top of the page, click the **Survey Campaign** tab.

   The **Survey Campaigns** page will result. Sub-tabs will appear immediately below the **Survey Campaign** tab, labeled **Survey Campaign, Cycle, Deployment,** and **Survey Resources**.
4. Click the **Deployment** sub-tab

   Upon clicking **Deployment**, the **Deployments** page results.

   Based on the default choice of the **View By** dropdown for your environment (either **My Deployments** or **All Deployments**), either all existing deployments are displayed, or only the set of deployments created by your current User ID (if any exist). You can change this view at any time by selecting the remaining choice from the **View By** dropdown.
5. Click the hyperlink for the deployment you wish to suspend.

   The **Deployment Details** page will result. The value displayed in the **Status** dropdown of a deployed deployment is `Active`.
6. From the **Status** dropdown, select `Incomplete` or `Closed`.
7. Click **Update** at the bottom of the **Deployment Details** page to save this new status.

   The **Deployment Details** page will result. The value formerly displayed as a value of Active in the **Status** dropdown has been replaced with a hard-coded value of `Incomplete` or `Closed` (as appropriate).

Caution: The deployment will not be suspended unless you click **Update** as described above.

See Also

- Deploying a Survey Campaign Deployment

- Understanding Status Conditions
- Status Conditions for Survey Campaigns, Cycles, and Deployments
- How to Deploy or Suspend a Survey Campaign

4.3.4.3 Understanding Status Conditions

At the three different hierarchies of survey campaign administration—creating the survey campaign, defining a cycle, and defining a deployment— there are different status conditions. This condition indicates to a survey campaign administrator the status of the survey campaign, cycle and deployment.

For example, a survey campaign cannot be executed when its status is `Open`. A cycle and deployment must be associated with a survey campaign, and the deployment must be explicitly deployed (its status set to `Active`), before the survey campaign can be executed. As another example, while survey campaigns cannot be deleted from the Survey Admin console, you can change the status of an open survey campaign to `Cancelled`. A list of all status levels for the various hierarchies is included in the section Status Conditions for Survey Campaigns, Cycles, and Deployments.

Survey Campaign and Cycle Status

Once created, and prior to defining a deployment, the status of a survey campaign (and cycle) is `Open`. You can view and change the status from the **Survey Campaigns** page. From the **Status** dropdown control, you can select `Cancelled` to invalidate this definition. Once the survey campaign has been completed, the other possible status is **Closed**.

Deployment Status

Once a deployment is created, its status is `Open`. This does not change until a deployment is deployed, although once created it can also immediately be changed to a status of `Cancelled`. This would have the effect of invalidating this deployment definition.

Once you deploy a deployment, its status becomes `Active`. This is true whether the **Deploy Date** is in the past or the future.

For non-list-based survey campaign deployments for which the **Deploy Date** is in the past, this means that as soon as you deploy (change the status to `Active`), you can take the survey in a Web browser using the appropriate URL (including dID for the Active deployment). For more information on the URL, see Non-List-Based URL Syntax and Example below.

You can make a non-list-based survey campaign deployment `Active` if its **Deploy Date** is in the future, but the URL will not become active until the **Deploy Date** and **Deploy Time** are reached.

Additional Requirements for List-Based Deployments

For **list-based** survey campaigns, deploying a survey campaign deployment will still make the deployment **Active**. However, you must wait until the Fulfillment Engine completes its activity (completes the Fulfillment request and succeeds in sending the invitation Master Documents to the outgoing mail server specified in Fulfillment) before you will have respondents.

Survey administrators with Fulfillment administrator responsibility will be able to view the status and history of Fulfillment requests from the Survey Admin console. A Request Status of **Submitted** indicates that the list was successfully sent to the Fulfillment engine. An Outcome Code of **SUCCESS** indicates that Fulfillment Server was successful in sending the invitation (or reminder) Master Document. For any other outcome code (**Partially Successful** or **Failure**), you will need to log into the Fulfillment Admin Console to resolve.

In this scenario, list members must receive and respond to an e-mailed invitation before you can expect activity for this deployment.

See Also

- Deploying a Survey Campaign Deployment
- Suspending a Survey Campaign Deployment
- Status Conditions for Survey Campaigns, Cycles, and Deployments
- How to Deploy or Suspend a Survey Campaign

4.3.4.4 Status Conditions for Survey Campaigns, Cycles, and Deployments

Table 4–5 Status Conditions for Survey Campaigns, Cycles, and Deployments

Survey Hierarchy	Status Condition	Explanation
Survey Campaign	Open	Survey campaign and cycle have been defined, but not deployed.
Survey Campaign	Active	Survey campaign, cycle, and deployment have been defined, and at least one deployment has been set to `Active`.

Table 4–5 Status Conditions for Survey Campaigns, Cycles, and Deployments

Survey Hierarchy	Status Condition	Explanation
Survey Campaign	Open	Survey campaign has no cycles defined or no active cycles.
Survey Campaign	Active	One or more cycles havae been defined and at least one cycle is active.
Survey Campaign	Closed	The requirements for a survey campaign have been completed. All cycles are closed and cancelled cycles (if any) are disregarded.
Survey Campaign	Inactive	Survey campaign has been disabled by survey administrator at the deployment level.
Cycle	Open	Cycle has no deployments defined or no active deployments.
Cycle	Active	One or more deployments have been defined and at least one deployment is active.
Cycle	Closed	The requirements for a survey campaign have been completed or disabled by a survey administrator at the cycle level.
Cycle	Cancelled	A cycle has been disabled by a survey administrator at the cycle level.
Deployment	Open	A deployment has been defined, but not yet been made active.
Deployment	Cancelled	A deployment that was previously open can be cancelled at the deployment level *before it is made active* and will then be disregarded.
Deployment	Incomplete	A deployment that was previously made active has been disabled by a survey administrator at the deployment level prior to completion. No responses will be collected.
Deployment	Active	One or more deployments have been defined and at least one made active. If the Deploy Date is in the past and the Response End Date in the future, this deployment may be receiving responses.
Deployment	Closed	The requirements for a survey campaign have been completed or disabled at the deployment level by a survey administrator.

Note that a survey campaign and cycle can be created in the same HTML page. As such, they share a status. Status can be changed from the **Survey Campaigns** page, and that same status is displayed on the **Cycles** page.

See Also

- Deploying a Survey Campaign Deployment
- Suspending a Survey Campaign Deployment
- Understanding Status Conditions

4.3.5 Modifying Survey Campaign Details

Once a survey campaign is created, and before a deployment is active, you can change certain properties of the survey campaign. These include:

Survey Campaign Detail:

- Description
- Header Page
- Final Page
- Error Page

Cycle Detail:

- Anonymous flag
- Minimum Response Percent

Once a survey campaign is deployed, you can no longer change any aspects other than the Status.

See Also

- Suspending a Survey Campaign Deployment
- Understanding Status Conditions
- Status Conditions for Survey Campaigns, Cycles, and Deployments

4.4 Generating Lists for List-Based Survey Campaigns

List generation is a function of Oracle Marketing Online, which is required for list-based survey campaigns. For information regarding the planning aspects of list generation as pertaining to the Survey component of Oracle Scripting, see

Requirements for Generating Lists above. For detailed administration steps, see the *Oracle Marketing Online Concepts and Procedures Manual*, specifically the **View a List, Create a List,** and **Import a List** sections under the topic **Using the Audience Tab**.

See Also:

- Navigating the Survey Admin Console Graphic User Interface
- Administering Survey Campaign Details
- Administering Survey Resources
- Setting Up Invitations and Reminders for List-Based Survey Campaigns
- Managing Survey Campaign Responses
- Reporting and Analyzing Survey Respondent Data

4.5 Setting Up Invitations and Reminders for List-Based Survey Campaigns

For list-based survey campaign operations, list members are sent an *invitation* to participate in a survey. This invitation is an HTML-formatted e-mail message sent from the Fulfillment Engine (a component of Oracle One-to-One Fulfillment) based on a list designated from Oracle Marketing Online (OMO). If desired, a second e-mail message can be sent as a *reminder*.

Invitations and reminders are Fulfillment Master Documents. From a Survey perspective, they are only required for *list-based* survey campaigns. Administration of Master Documents is typically a function of Fulfillment administrators. This topic group includes a discussion of the theory and suggested Fulfillment administration steps, most of which may be performed from the Survey Admin console. For more information on any of the following topics, consult with the Fulfillment administrator for your environment. For more information, refer to Fulfillment documentation and resources.

Topics include:

- Fulfillment Activities Required for List-Based Survey Campaigns
- Recommended Order of Steps
- Invitations
- Reminders
- Sample Invitation Master Document

- Sample Query
- Verifying Status of a Fulfillment Request

4.5.1 Fulfillment Activities Required for List-Based Survey Campaigns

For ease of administration, survey administrators have access to a **Fulfillment** tab in the Survey Admin console. This tab gives access to the Fulfillment steps required for the Survey component of Oracle Scripting. Occasionally, other steps may need to be performed from a Fulfillment perspective. Since the requirement for Fulfillment to be appropriately implemented and functional is prerequisite, a Fulfillment administrator will already exist in the enterprise who has access to the Fulfillment Admin Console to perform additional Fulfillment steps.

When you click the **Fulfillment** tab from the Survey Admin console, the following sub-tabs are accessible:

- Master Document
- Template
- Collaterals (not used for survey campaign administration)
- Query
- DataSource
- Status
- History

There is no method to create an HTML document to serve as a Master Document from Fulfillment. You must use a text or HTML editor to create invitations and reminders following the prescribed format. Once the document is created, however, you can achieve the following from the **Fulfillment** tab in the Survey Admin console:

- The uploading of invitation and reminder Master Documents created in HTML
- The creation of queries to populate merge fields in Master Documents
- The association of a query with a specific Master Document
- Creation of a Fulfillment Template (a logical construct which associates a Master Document, its merge fields, queries and relevant tables)
- The monitoring of Fulfillment Server status and history

See Also

- Recommended Order of Steps
- Invitations
- Reminders
- Sample Invitation Master Document
- Sample Query
- Verifying Status of a Fulfillment Request

4.5.2 Recommended Order of Steps

The recommended order of steps for administering invitations and reminders using Fulfillment functionality is as follows:

1. Identify merge fields to be populated in the Master Document, and identify locations of tables or views in the Oracle Applications database schema from which the merge fields will be populated.
2. Create query, including one merge field for each table field queried.
3. Create Master Document in HTML using text editor or HTML tool.
4. Upload Master Document
5. Associate Master Document with query.
6. Define Template.
7. Associate Template with Master Document.

For More Information

For more information, refer to all appropriate Oracle One-to-One Fulfillment documents.

See Also

- Fulfillment Activities Required for List-Based Survey Campaigns
- Invitations
- Reminders
- Sample Invitation Master Document
- Sample Query

- Verifying Status of a Fulfillment Request

4.5.3 Invitations

Typically, an invitation is an HTML message inviting list members to take a survey. The invitation is referred to in Fulfillment as a Master Document. Using Fulfillment APIs, invitations can be customized by populating data from RDBMS tables.

If personalized, the HTML document contains placeholders called **merge fields**, in which information from a table query is placed by the Fulfillment server upon execution. These might typically include a list member's title, first name, and last name. While such personalizations are not required, a Survey URL *is* required. In list-based scenarios, a unique URL is generated for each list member (using dID and rID as parameters in the URL string). In this way, if a survey campaign has a business rule to allow list members to only take a survey one time, this rule can be enforced.

See Also

- Fulfillment Activities Required for List-Based Survey Campaigns
- Recommended Order of Steps
- Reminders
- Sample Invitation Master Document
- Sample Query
- Verifying Status of a Fulfillment Request

4.5.4 Reminders

If required by a survey campaign, reminders can also be sent to list members. The reminder, also an e-mail message, is another Fulfillment Master Document. It can even be the very same e-mail message initially sent to list members. Typically, however, reminders might be worded slightly differently, as they often emphasize the response end date of the specific deployment for that survey campaign. Nonetheless, the reminder must be an HTML document.

Reminders and the Anonymous Flag

If a deployment has reminders associated with it and the parent cycle is designated as `Anonymous` in the Survey Admin console, reminders will be sent to *all* list

members. If the cycle is *not* anonymous, reminders will be sent only to list members that have *not yet responded* to the survey invitation.

See Also

- Fulfillment Activities Required for List-Based Survey Campaigns
- Recommended Order of Steps
- Invitations
- Sample Invitation Master Document
- Sample Query
- Verifying Status of a Fulfillment Request

4.5.5 Sample Invitation Master Document

The following is a sample of what an invitation Master Document might say, including three merge field. Merge fields are signified by double brackets « ».

Sample Invitation Text

Following is the sample text of the e-mail message:

> Hello «FIRST_NAME» «LAST_NAME»,
>
> Please take a moment to participate in this customer satisfaction survey, using your Web browser. Click «SURVEY_URL_HYPERLINK» to begin.
>
> Sincerely,
>
> Vision Customer Service

Sample Invitation Code

Following is the HTML code of the above e-mail message. Note the merge fields, which use the ASCII references `&laquo;` (for «) and `&raquo;` (for »).

```
<!doctype html public "-//w3c//dtd html 4.0
transitional//en">

<HTML>

<HEAD>
```

```
<META HTTP-EQUIV='Content-Type" CONTENT="text/html;
charset=iso-8859-1">

<META NAME="Author" CONTENT="Anony Mouse">

<META NAME="GENERATOR" CONTENT="Mozilla/4.76 [en]
(WinNT; U) [Netscape]">

<TITLE>hrefdurl</TITLE>

</HEAD>

<BODY>

Hello &laquo;FIRST_NAME&raquo; &laquo;LAST_NAME&raquo;,

<p>Please take a moment to participate in this customer
satisfaction survey, using your Web browser. Click
&laquo;SURVEY_URL_HYPERLINK&raquo; to begin.

<P>Sincerely,</P>

<BR>Vision Customer Service

</BODY>

</HTML>
```

See Also

- Fulfillment Activities Required for List-Based Survey Campaigns
- Recommended Order of Steps
- Invitations
- Reminders
- Sample Query
- Verifying Status of a Fulfillment Request

4.5.6 Sample Query

The following is a sample query that would populate the «FIRST_NAME», «LAST_NAME», and «SURVEY_URL_HYPERLINK» merge fields in the above invitation sample:

```
select

  ale.first_name  FIRST_NAME
```

```
 ,ale.last_name  LAST_NAME
 ,'<a
href='||ihp.SURVEY_URL||'?dID='||asda.survey_deployment_
id||'&rID='||isle.respondent_id||'> here </a>' SURVEY_
URL
 from
  IES_SVY_LIST_ENTRIES_V isle
 ,ams_list_entries ale
 ,IES_SVY_DEPLYMENTS_V asda
 ,IES_SVY_HOSTING_PROF_V ihp
where
  ale.list_entry_id = isle.list_entry_id
  and isle.survey_deployment_id = asda.survey_
deployment_id
  and ihp.survey_deployment_id = asda.survey_deployment_
id
  and asda.survey_deployment_id = :deployment_id
  and ale.list_entry_id = :party_idHello
```

See Also

- Fulfillment Activities Required for List-Based Survey Campaigns
- Recommended Order of Steps
- Invitations
- Reminders
- Sample Invitation Master Document
- Verifying Status of a Fulfillment Request

4.5.7 Verifying Status of a Fulfillment Request

In the **Request Status** page (accessed by clicking the **Status** sub-tab of the **Fulfillment** tab from the Survey Admin console), a request status of **Submitted**

indicates the Fulfillment Request has been successfully submitted to the Fulfillment Engine. This does not necessarily signify the appropriate end state.

In order to verify successful execution of the Fulfillment Engine in completing distribution of Fulfillment requests, check for an outcome code of SUCCESS in the **Request History** page (accessed by clicking the **History** sub-tab of the **Fulfillment** tab).

If any status other than Submitted displays in the **Request Status** page, you must contact the Fulfillment admin, who will access the Fulfillment Admin Console to follow up on the Fulfillment status.

See Also

- Fulfillment Activities Required for List-Based Survey Campaigns
- Recommended Order of Steps
- Invitations
- Reminders
- Sample Invitation Master Document
- Sample Query

4.6 Managing Survey Campaign Responses

The **Response** tab in the Survey Admin console provides you with the means to manage responses for ongoing, active deployments. For example, you can monitor an ongoing, active deployment and view each individual's responses, as soon as the respondent completes a survey questionnaire. For list-based survey campaigns, you can also manually send reminders to individual list members. Any reminders sent in such a manner are *in addition to* scheduled, deployment-specific reminders you have set up when defining the deployment. Scheduled reminders will be batch-processed according to the Reminder Interval increment parameter. Nonetheless, individual reminders use the same Fulfillment components (reminder Master Document, associated query, and Fulfillment Template) as the batch-processed reminders.

Topics in this group include:

- Navigating the Response Tab
- Viewing Individual Responses in the Survey Admin Console
- Understanding the GUI Display of Individual Responses

- Manually Sending Reminders to Individual List Members
- Administering concurrent processing to move respondent data to Survey summary tables

4.6.1 Navigating the Response Tab

Upon selecting the **Response** tab in the Survey Admin Console GUI, the **Home** tab results. At this time, the contents of the **Home** tab are blank for this category. The remaining sub-tabs follow the familiar pattern of **Survey Campaign, Cycle,** and **Deployment.**

The **Survey Campaign** sub-tab lists survey campaigns, the **Cycle** sub-tab lists cycles, and the **Deployment** sub-tab lists deployments. Since the function of this tab is to view responses or send reminders, there is no facility to create or define any of these elements (this is located under the **Survey Campaign** main tab only). Since an individual's response is technically a response to a specific *deployment* (that is a child of a cycle, which in turn is a child of the survey campaign), you can click any sub-tab under the **Response** category and drill down, if desired. For fastest results, navigate immediately to the Deployment sub-tab.

See Also

- Viewing Individual Responses in the Survey Admin Console
- Understanding the GUI Display of Individual Responses
- Manually Sending Reminders to Individual List Members
- Administering concurrent processing to move respondent data to Survey summary tables

4.6.2 Viewing Individual Responses in the Survey Admin Console

Prerequisites

In order to view individual survey respondent answers:

- A survey campaign and cycle must have been created.
- A deployment must have been set to Active status.
- At least one individual must already have completed a survey questionnaire online.

- You must have access to the Survey Admin console (you must have access to a user with the `Survey Administrator` responsibility).
- You must know the survey campaign and cycle in order to view an individual responses.
- For list-based scenarios, a list member must be a respondent in order to view his or her responses to the survey.
- For list-based scenarios, you must know the list member/respondent's name to view specific responses. All list members (whether they are currently respondents or not) will be listed on the **Deployment Responses** page in the **Response List** table.

Tip: You can drill down in the GUI from the survey campaign, or go straight to the list of deployments to view responses.

Steps

1. Using an Oracle Applications 11*i*-compliant Web browser, access the appropriate URL for your environment to log into JTF tech stack HTML-based applications.
2. From the eBusiness Center applications login screen, enter in the provided fields a **User ID** and **Password** for a user with the `Survey Administrator` responsibility, and click **Go**.

   The Survey Admin console results.
3. From the set of tabs at the top of the page, click the **Response** tab.

   The **Home** page for the **Response** tab will result. Sub-tabs will appear immediately below the **Response** tab, labeled **Home, Survey Campaign, Cycle,** and **Deployment**.

   Note: At this time, there is no function associated with the Home sub-tab. This page will therefore appear as a blank page.

   You can drill down in the GUI from the survey campaign, or you can immediately navigate to the list of deployments to view responses. The longer path is shown in this set of steps. The shorter path is shown in the process that describes manually sending reminders to individual list members.

4. Click the **Survey Campaign** sub-tab.

   The **Survey Responses** page results, with a list of survey campaigns (if any) that match the **View By** dropdown criteria.

   Tip: Does this page look familiar? It displays survey campaigns just as if you had selected the **Survey Campaign** sub-tab under the **Survey Campaign** tab. However, under the **Response** tab, survey campaigns cannot be *defined*, only viewed. Therefore, no **Create** button or functionality displays.

5. Optionally, adjust the survey campaigns listed on this page as required so you can see the appropriate survey campaign. You can do this by changing the **View By** dropdown choice or selecting the **First**, **Previous**, **Next**, or **Last** navigation hyperlinks at the bottom of the page.

6. In the **Survey Campaign** column (the first column displayed), click on the hypertext link for the appropriate survey campaign.

   The **Survey Campaign Details** page results, displaying the survey campaign name and its survey questionnaire script. Below this, the text contents of the Description field for this survey campaign (defined at the time of the survey campaign definition) display.

   Under the **Survey Cycles** heading, an HTML table will result with a list of all cycles defined for this survey campaign. These fields are described in the table below.

Table 4–6 Survey Cycles Fields on* Survey Campaign Details *Page

Field Name	Field Type	Explanation
Cycle	Text (VARCHAR2) with Hypertext link	Number of reminders already sent to each list member displayed (will read 0 if no reminders associated with this deployment or if none have yet been sent).
Anonymous	Yes or No	If list member is respondent, this is a hypertext link to the specific answers for this respondent. The word "Answer" will appear without a hypertext link if list member has not responded.

Minimum Responses	Numeric	Clicking this button will send a reminder if supported by the information included in this deployment (i.e., if reminders are specified).
Status	Active	This is a list-based feature only. This feature not yet implemented

7. In the **Cycle c**olumn (the first column displayed), click on the hypertext link for the appropriate cycle.

   The **Cycle Details** page results, displaying the survey campaign name and the current cycle for which details are displayed. Below this, the status of the survey campaign and cycle is displayed, as well as other cycle details such as whether or not the cycle is anonymous, and the designated minimum response percentage that is the target of this cycle.

   Under the **Deployments** heading, an HTML table will result with a list of all deployments defined for this cycle. These fields are described in the table below.

Table 4–7 Deployments Fields on* Cycle Details *Page

Field Name	Field Type	Explanation
Deployment	Text (VARCHAR2) with Hypertext link	Number of reminders already sent to each list member displayed (will read 0 if no reminders associated with this deployment or if none have yet been sent).
List Name	Yes or No	If list member is respondent, this is a hypertext link to the specific answers for this respondent. The word "Answer" will appear without a hypertext link if list member has not responded.
Media	Hard-Coded Text (11.5.6)	This field contains a hard-coded value of `Web` only for 11.5.6, the sole media type supported by the Survey component o71

Table 4–7 Deployments Fields on* Cycle Details *Page

Field Name	Field Type	Explanation
Total Sent	Numeric	For non-list-based survey campaigns this value has no meaning and will be zero. For list-based, this value contains the amount of invitations sent by the Fulfillment server.
Total Received	Numeric	This indicates the number of responses received. *Click* **Refresh** *to ensure this value reflects the most recent number of responses received.*
Errors	Numeric	This indicates errors encountered with existing responses.
Percent Complete	Numeric	This shows the percentage completed based on the amount of list members for list-based survey campaigns.
Refresh	Button	Must be clicked to show most recent data.

8. In the **Deployment** column (the first column displayed), click on the hypertext link for the appropriate deployment for which you wish to view responses.

The **Deployment Responses** page results, displaying the hierarchy of survey campaign, cycle, and deployment designated. Below this, the list used (if list-based) is displayed, as well as the designated media type for this deployment (`Web` is currently the only supported media type).

Under the **Response List** heading, an HTML table will result.

Note: As described below, the fields displayed in the Response List table on the Deployment Responses page are different for list- and non-list-based survey campaigns. Nonetheless, clicking on the **Answer** hyperlink will allow you to view individual responses, whether the survey campaign in question is list-based or non-list-based.

List-Based Response List Fields

Table 4–8 List-Based Response Fields* on Deployment Response *Page

Field Name	Field Type	Explanation
Name	Text	Name of specific list member. Names appear here regardless of whether this list member has responded to a survey invitation.
Invitation Date	Date Field (DD_MMM_YYYY)	Date list member was sent invitation from Fulfillment server.
Response Received Date	Date Field (DD_MMM_YYYY)	Date respondent completed survey
Reminders Sent	Alphanumeric selection	Number of reminders already sent to each list member displayed (will read 0 if no reminders associated with this deployment or if none have yet been sent).
Answer	Text "Answer"	If list member is respondent, this is a hypertext link to the specific answers for this respondent. The word "Answer" will appear without a hypertext link if list member has not responded.
Send Reminder	Button	Clicking this button will send a reminder if supported by the information included in this deployment (i.e., if reminders are specified).
Error	Checkbox	This is a list-based feature only. This feature not yet implemented

Non-List-Based Response List Fields

Table 4–9 Non-List-Based Response Fields **on Deployment Response** *Page*

Field Name	Field Type	Explanation
Sequence	Alphanumeric selection	Survey response in order of survey questionnaires completed online
Response Received Date	Date Field (DD_MMM_YYYY)	Date respondent completed survey
Answer	Hypertext	This is a hypertext link to the specific answers for this respondent
Error	Null	Not relevant to non-list-based deployments

9. Click the **Answer** hyperlink for a particular respondent.

> **Note:** If no hyperlink appears in the **Answer** column under the **Response List** heading for a table entry (row), this signifies a list-based scenario in which the list member had not responded. In this case, no responses are available to view.

10. A new page will appear, in which all responses to a survey questionnaire by this particular respondent will be listed.

See Also

- Navigating the Response Tab
- Understanding the GUI Display of Individual Responses
- Manually Sending Reminders to Individual List Members
- Administering concurrent processing to move respondent data to Survey summary tables

4.6.3 Understanding the GUI Display of Individual Responses

Topics include:

- Relationship to the Script Author
- Survey Responses GUI Description
- How Different UI Types Display

See Also

- Navigating the Response Tab
- Viewing Individual Responses in the Survey Admin Console
- Manually Sending Reminders to Individual List Members
- Administering concurrent processing to move respondent data to Survey summary tables

4.6.3.1 Relationship to the Script Author

Questions for the survey questionnaire are created using **panel** objects in the Script Author. Every Script Author object contains viewable properties such as a name. Every panel must also include, at minimum, one "question" or answer definition. (Each answer definition within a panel is sometimes referred to as a "node.") All answer definitions include:

- **Default for Distinct Branching** checkbox
- **Name** text field
- **UI Type** (this is a dropdown with values of Text Field, Text Area, Radio Button, Check Box, Button, Drop Down, and Password)
- **Label for Reporting** text field

Panel properties displayed in the **Responses** page include the panel name, the answer definition name (the **Name** value in the Answer Entry Dialog of the Script Author), and the answer chosen by the respondent.

See Also

- Survey Responses GUI Description
- How Different UI Types Display

4.6.3.2 Survey Responses GUI Description

You can view the responses from any individual respondent from the **Responses** tab of the Survey Admin console. Navigate to the **Deployment** sub-tab and drill down to the Deployment Responses detail for the particular deployment. While this displays different information for list- and non-list-based survey campaigns, you view response information for both by clicking the **Answer** hyperlink from the **Response List** table on the **Deployment Responses** page.

The resulting HTML page will contain all the responses from that individual's survey questionnaire session in an HTML table with two columns.

The name of each panel visited by the survey respondent appears in the unshaded row in the column on the left. The right column of each unshaded panel name row is empty in all cases. Based on the flow taken by the survey respondent (taking into account any branching included in the script), some panels in the script may not appear on the responses page (only those panels viewed by the respondent will appear).

Below each unshaded row indicating a panel, one or more shaded horizontal rows will display. These shaded rows represent the one or more answer definitions that are associated with that panel in the Script Author. Thus, you can determine the number of answer definitions (or "questions") included in each panel visited by counting the shaded rows under each unshaded row.

Each answer definition associated with that panel appears in a shaded table row below the panel name. The left column contains the answer definition name (question name). This corresponds to the Name value in the Script Author Answer Entry Dialog. The right column contains answers.

For answer definitions with more than one lookup value or choice (whether predefined, or pulled from the database), all answer choices will appear in the column on the right. The answer selected by the respondent displays in a bold, sans serif typeface, whereas the remaining lookup values display in plain text in a serif typeface. For answer definitions with only one value, that value will appear in the shaded row in the right column in a bold, sans serif typeface.

Note: The appearance of individual responses as described above might appear slightly differently based on standard and user-defined Web browser settings and HTML interpretation engines.

See Also

- Relationship to the Script Author
- How Different UI Types Display

4.6.3.3 How Different UI Types Display

Text field or **text area** UI types contain text as entered by the respondent.

If the UI type specified in the Script Author is a **Radio Button** or a **Drop Down,** answers will display as entered into the **Specify Lookups** portion of the **Lookups** tab of the data dictionary (if a lookup value) or as entered into the database (if specified as a **Cursor Lookup** in the **Lookups** tab of the data dictionary).

Checkboxes do not need to be checked unless the control is programmed with a default value in the Script Author. A checkbox that remains unchecked displays a value in the Responses page of **0**, which signifies false. When a respondent marks the checkbox (or such a value is provided by default and not changed by the respondent), this value will display as **1**, which signifies true.

If the UI type specified in the Script Author is a **Button**, this will appear as a button or "push button" in the HTML-rendered page, with the display value populating the button as entered into the **Specify Lookups** portion of the **Lookups** tab of the data dictionary (if a lookup value) or as entered into the database (if specified as a **Cursor Lookup** in the **Lookups** tab of the data dictionary).

Password UI types will display the value entered by the respondent with asterisks.

See Also

- Relationship to the Script Author
- Survey Responses GUI Description

4.6.4 Manually Sending Reminders to Individual List Members

For list-based survey campaigns that employ reminders as well as invitations, you can manually send reminders to individual list members. If you do so, these reminders are in addition to batch-processed reminders defined at the deployment level.

Prerequisites

In order to send reminders, the following prerequisites apply:

- A survey campaign and cycle must have been created.
- A list-based deployment must have been set to Active status.
- You must have access to the Survey Admin console (you must have access to a user with the `Survey Administrator` responsibility).

- You must know the survey campaign, cycle, and deployment in order to reach the **Deployment Responses** page to send reminders.
- You must know the list member/respondent's name to explicitly send a reminder. All list members (whether they are currently respondents or not) will be listed on the **Deployment Responses** page in the **Response List** table.

Tip: You will most likely not need to send reminders to list members who have already responded to a survey invitation. Therefore, only send reminders for list members for whom the Answer text in the Answer column does not contain a hypertext link.

Note: The procedure Viewing Individual Responses in the Survey Admin Console contains one method to reach the **Deployment Responses** page. In contrast, the steps for this topic include a different, shorter path to reach the **Deployment Responses** page.

Steps

1. Using an Oracle Applications 11*i*-compliant Web browser, access the appropriate URL for your environment to log into JTF tech stack HTML-based applications.
2. From the eBusiness Center applications login screen, enter in the provided fields a **User ID** and **Password** for a user with the `Survey Administrator` responsibility, and click **Go**.

   The Survey Admin console results.
3. From the set of tabs at the top of the page, click the **Response** tab.

   The **Home** page for the **Response** tab will result. Sub-tabs will appear immediately below the Response tab, labeled **Home, Survey Campaign, Cycle,** and **Deployment**.
4. Click the Deployment sub-tab.

   The **Deployments** page results, with a list of survey deployments (if any) that match the **View By** dropdown criteria.

5. Optionally, adjust the deployments listed on this page as required so you can see the appropriate survey deployment. You can do this by changing the **View By** dropdown choice or selecting the **First**, **Previous**, **Next**, or **Last** navigation hyperlinks at the bottom of the page.
6. In the **Deployment** column (the first column displayed), click on the hypertext link for the appropriate survey deployment.

   The **Deployment Responses** page results, displaying the hierarchy of survey campaign, cycle, and deployment designated. Below this, the deployment list name is displayed, as well as the `Web` media type.

   Under the **Response List** heading, an HTML table will result. To see all field names, see the table List-Based Response Fields on Deployment Response Page.
7. Click **Reminder** in the Send Reminder column to send a Fulfillment request for immediately issuing a reminder.

   This will utilize the same Fulfillment Master Document, query, and template as the reminders batch processed by deployment. Note that list members with hypertext links in the Answer column have already responded to the survey.

See Also

- Navigating the Response Tab
- Viewing Individual Responses in the Survey Admin Console
- Understanding the GUI Display of Individual Responses
- Administering concurrent processing to move respondent data to Survey summary tables

4.6.5 Administering Concurrent Processing to Export Respondent Data to Survey Summary Tables

The Concurrent Manager is functionality built on Oracle Application Object Library (AOL) classes, which are included in Oracle Applications 11i with an appropriate Rapid Install.

Using Oracle Concurrent Manager, you must schedule or run concurrent programs. From an Oracle Scripting perspective, there are two concurrent programs. They are primarily used with the Survey component. . These generate updated status information (for campaigns created and managed from the Survey Admin console) and reporting data received in the Scripting schema of the Applications databse from executed scripts. The two concurrent programs are described below.

Update Deployment Status

This program checks survey campaign deployments and changes the status to Closed from Active if deployments are past the response end date. Oracle Corporation recommends that this program be scheduled to run every day at a certain time.

Summarize Survey Data

This programs moves certain subsets of data from transaction tables in the IES schema of the applications database to summary tables. As scripts are executed, data is collected over a period of time (for Web-based surveys and for scripts with Footprinting enabled). This program summarizes and indexes the portions of that data required for reporting, moving it to smaller summary tables. These tables have a physical layout structured for efficient processing based not on individual respondents, but on trends for all existing data for each reporting category. The summarized data is used to generate reports from the Survey Admin console using the **Analysis** tab. Oracle Corporation recommends that this program be scheduled to run at periodic intervals.

Prerequisites

- To administer concurrent programs, you must have a user with the `iSurvey User` responsibility.
- Using the **Analysis** tab in the Survey Admin console relies on Java classes that require the Java Development Kit (JDK) 1.2 or higher on the Oracle Applications server. Other Oracle Applications components may use earlier JDK versions such as 1.1.8. *If reporting is required in your enterprise you must upgrade to 1.2 or higher.* For the latest information on certified and supported technologies such as JDK, refer to Oracle*MetaLink*.

Steps

1. Using an Oracle Applications 11*i*-compliant Web browser, access the appropriate URL for your environment to log into Forms-based applications.

   Either the SSWA or Forms login as appropriate.

2. From the applications login screen, enter in the provided fields a **User Name** and **Password** for a user with the `iSurvey User` responsibility, and click **Connect**.

3. From the list of responsibilities, select `iSurvey User`. If necessary, click **OK** to confirm your selection.

The resulting menu will display the various options for concurrent processing

4. Run the two Survey-related concurrent programs: `Update Deployment Status` and `Summarize Survey Data`.

   For specific details on executing or scheduling concurrent programs, please refer to Chapter 5 of the *Oracle System Administrator's Guide*.

5. After executing each concurrent program, log out of Forms-based Oracle Applications if desired by selecting **Exit Oracle Applications** from the **File** menu.

6. Summary data is now available to view in the Survey Admin console by accessing the **Analysis** tab. In order to access this console, you must log into CRM HTML-based applications from the Oracle eBusiness Center login with a user that has the `Survey Administrator` responsibility.

See Also

- Navigating the Response Tab
- Viewing Individual Responses in the Survey Admin Console
- Understanding the GUI Display of Individual Responses
- Manually Sending Reminders to Individual List Members

4.7 Reporting and Analyzing Survey Respondent Data

The ability to report and analyze survey campaign results is critical; enterprises use the analytical information made available through conducting survey campaigns to understand what customers and employees think and how they can improve their sales and service lines and business processes.

While the **Response** tab in the Survey Admin console provides access to how *individuals* responded, data received from survey respondents is *summarized* for reporting purposes and made available as *collective summary data* in several preformatted reports accessible from the **Analysis** tab.

Three of the four existing reports are specific to the Survey component of Oracle Scripting; the remaining report is specific to use of the Scripting Engine component of Oracle Scripting only.

- The **Response Summary** report shows how many responses were received as a percentage of the total amount of responses solicited. For list-based survey campaigns, the "Error" parameter will indicate the amount of invalid list

entries (the Fulfillment Engine was not able to deliver the invitation or reminder based on the e-mail address in the list). By reviewing the Response Summary report, an administrator can determine the success of a survey campaign. For list-based survey campaigns, this would include the number of list members that responded to the survey as compared to the number of invitation sent. In addition, administrators can get a sense of the different response rates for different types of surveys using this report.

- The **List Summary** provides, by list, a view of how many responses were received as a percentage of the total amount of responses solicited, and how this maps to the targeted number of responses established as the "Minimum Response Percent" figure established in the deployment parameters as a deployment goal. By reviewing the List Summary report, an administrator can determine the success of a list-based survey campaign. In addition, administrators can get a sense of the quality of lists. Establishing multiple deployments using the same survey questionnaire and the same target as the "Minimum Response Percent" but with different lists, administrators can compare list penetration and determine more successful and less successful lists.
- The **Question Frequency** is a summary report listing every question in a given survey questionnaire script, the possible lookup values or responses to each answer, and how many responses each choice received. Essentially, this report provides the detail for each question asked in a survey deployment. This report is most useful when you need to drill down to the actual survey results. For instance, in a customer satisfaction survey, what percentage of customers were "satisfied" versus "dissatisfied"? What were the reasons stated for dissatisfaction? These answers show as both quantities and percentages in a tabular format, as well as within a pie graph. Through analysis of the Question Frequency report, an enterprise can understand what initiatives are successful and which need to be improved. Enterprises can determine new initiatives to solve issues identified by consumers based on respondent data received.
- The **Panel Footprint** report indicates, by panel per script, what panels were visited and the duration of time (in milliseconds) spent in each panel. The business objective of running this report is to see how effective a script is at leading an agent through a dialog with a customer (for the interaction center) or determining which questions may have been the most confusing (for survey respondents). Essentially, this report helps in tuning a script for maximum clarity and streamlined flow. It is typically of most use in the interaction center to reduce talk time. Note that if this report is not used, footprinting should be disabled at the script level to conserve system resources.

The reports take the respective parameters listed below.

Table 4–10 Parameters Required for Reports Accessible in Survey Admin Console

Report Type	Parameters Required to Run Report
Response Summary Report	Survey Campaign
List Summary Report	Survey Campaign Start Date End Date
Question Frequency Summary	Survey Campaign
Panel Footprint Report	Script Name Start Date End Date

This section provides steps to generate reports using the Survey Admin console.

Prerequisites

- The Survey component of Oracle Scripting must be fully implemented at an enterprise.
- Survey questionnaire scripts must have been generated, tested and deployed to the enterprise applications database.
- Survey campaign details must have been administered in the Survey Admin console and a deployment deployed.
- Responses to the survey campaign must have been received.

Steps

1. Using an Oracle Applications 11*i*-compliant Web browser, access the appropriate URL for your environment to log into JTF tech stack HTML-based applications.

2. From the eBusiness Center applications login screen, enter in the provided fields a **User ID** and **Password** for a user with the `Survey Administrator` responsibility, and click **Go**.

   The Survey Admin console results.

3. From the set of tabs at the top of the page, click the **Analysis** tab.

   Sub-tabs will appear immediately below the **Analysis** tab, labeled **Response Summary, List Summary, Question Frequency,** and **Panel Footprint.** By default, the **Response Summary** report page displays.

4. Click the sub-tab corresponding to the type of report you wish to generate. (If you wish to generate a **Response Summary** report, skip this step.)

   The selected report page displays.

5. Enter the criteria requested by the selected report type, as described in the Parameters Required for Reports Accessible in Survey Admin Console table above.

6. When you have input all data requested by the report type, click **Run Report** at the bottom of the page.

   The report will generate and display on the page.